

**Jihočeská univerzita v Českých Budějovicích  
Přírodovědecká fakulta**

**Vytvoření CD a MP3 přehrávače za pomoci  
Raspberry Pi**

Bakalářská práce

**Ivan Kratochvíl**

Školitel: Mgr. Jiří Pech, Ph.D.

České Budějovice 2017

**Bibliografické údaje:**

Kratochvíl, I., 2017: Vytvoření CD a MP3 přehrávače za pomoci Raspberry Pi. [The construction of a CD and MP3 player using Raspberry Pi. Bc. Thesis, in Czech.] – 47 p., Faculty of Science, University of South Bohemia, České Budějovice, Czech Republic.

**Anotace:**

Cílem této bakalářské práce je sestrojít funkční CD a MP3 přehrávač za pomoci počítače Raspberry Pi. Přehrávač bude možné ovládat hardwarovými tlačítky, infračerveným dálkovým ovládáním, z webového rozhraní a z mobilní aplikace. Dále práce obsahuje návod ke zprovoznění vytvořeného zařízení, popis zdrojového kódu a postup při jeho testování.

**Klíčová slova:**

Raspberry Pi, hudební přehrávač, CD, MP3, python, javascript

**Annotation:**

The aim of this bachelor thesis is to construct functional CD and MP3 player with a use of the computer Raspberry Pi. The player is going to be controlled with hardware buttons, infrared remote controller, from web interface and from mobile application. The thesis also contains instructions of putting the created device into operation, a description of its source code and the testing process.

**Keywords:**

Raspberry Pi, music player, CD, MP3, python, javascript

## **Prohlášení**

Prohlašuji, že svoji bakalářskou práci jsem vypracoval samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své bakalářské práce, a to v nezkrácené podobě elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce. Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

V Českých Budějovicích dne .....

.....

Ivan Kratochvíl

## **Poděkování**

Rád bych poděkoval Mgr. Jiřímu Pechovi, Ph.D. za vedení mé bakalářské práce, cenné rady a odborný dohled. Také bych chtěl poděkovat svým blízkým za jejich trpělivost při mých studiích.

# Obsah

<b>1</b>	<b>Úvod</b>	<b>1</b>
<b>2</b>	<b>Cíle</b>	<b>2</b>
<b>3</b>	<b>Metodika</b>	<b>3</b>
3.1	Server . . . . .	3
3.2	Webová aplikace . . . . .	4
3.3	Mobilní aplikace . . . . .	4
3.4	Hardware . . . . .	4
<b>4</b>	<b>Teoretická část</b>	<b>5</b>
4.1	CD . . . . .	5
4.2	Raspberry Pi . . . . .	5
<b>5</b>	<b>Výběr hardwaru</b>	<b>7</b>
5.1	DVD mechanika . . . . .	8
5.2	Hardwarové ovládání . . . . .	8
5.2.1	PiFace CAD 2 . . . . .	8
5.3	Zvuková karta . . . . .	8
5.4	Napájení . . . . .	9
<b>6</b>	<b>Software</b>	<b>10</b>
6.1	Operační systém . . . . .	10
6.2	Audio přehrávač . . . . .	10
6.2.1	VLC media player . . . . .	11
6.2.2	mplayer . . . . .	11
6.2.3	mpv . . . . .	11
6.3	Programovací jazyk . . . . .	11
6.4	Další programy . . . . .	12
6.5	Použité knihovny . . . . .	12
6.5.1	pyudev . . . . .	12

6.5.2	python3-pifacecad . . . . .	13
6.5.3	mutagen . . . . .	13
6.5.4	Flask . . . . .	13
6.5.5	Socket.IO . . . . .	13
6.5.6	React a React Native . . . . .	13
6.5.7	Font Awesome . . . . .	14
6.5.8	react-native-music-control . . . . .	14
6.5.9	react-navigation . . . . .	14
<b>7</b>	<b>Praktická část</b>	<b>15</b>
7.1	Instalace . . . . .	15
7.2	Popis kódu . . . . .	17
7.2.1	Server . . . . .	18
7.2.2	Webový klient . . . . .	29
7.2.3	Mobilní klient . . . . .	33
7.3	Testování . . . . .	35
<b>8</b>	<b>Závěr</b>	<b>37</b>
<b>A</b>	<b>Fotografie zařízení</b>	<b>41</b>
<b>B</b>	<b>Screenshoty z webového klienta z tabletu a počítače</b>	<b>42</b>
<b>C</b>	<b>Screenshoty z mobilní aplikace</b>	<b>45</b>

# Kapitola 1

## Úvod

V dnešní době je velice oblíbené připojovat různá multimediální zařízení do počítačových sítí. Cena těchto síťových přístrojů ale bývá relativně vysoká. To samé platí i pro CD přehrávače. Zatímco dříve se výrobci snažili tyto přehrávače dělat přenosné, aby mohl člověk poslouchat hudbu ve sluchátkách kdekoliv, dnes, v době digitální distribuce, nalézají místo spíše na poličkách se zapojenými reproduktory. Člověk by řekl, že hudební CD jsou již archaická, ale vzhledem ke skutečnosti, že kvalita zvuku stále oblíbenějších internetových streamů či MP3 souborů nedosahuje kvality zvuku hudebních CD či vinylových desek, tak stále velké množství lidí upřednostňuje tyto hudební nosiče.

Jedním z nich je i autor práce, který má rád hudební umění a hudba pro něj není jen kulisa. Vlastní nemalé množství starých i nových hudebních CD a v současné době nevlastní zařízení, kde by je mohl přehrát. Zároveň se zajímá o fenomén Internet of Things, který se v moderní společnosti objevuje čím dál více. Tyto skutečnosti autora vedli k výběru tématu této práce.

V této práci je ukázáno, že pomocí jednodeskového počítače Raspberry Pi je možné vytvořit síťový přehrávač za nižší náklady než zařízení na trhu. Takové zařízení může být navíc ovládáno různými způsoby – např. pomocí hardwarových tlačítek, přes webové rozhraní, pomocí mobilního telefonu (smartphonu) nebo třeba pomocí infračerveného dálkového ovládání. Všemi těmito možnostmi se zde zabýváme.

# Kapitola 2

## Cíle

- Sestrojit audio CD a MP3 přehrávač z jednodeskového počítače Raspberry Pi 3 a USB DVD mechaniky.
  - Zjistit možnosti ovládání CD přehrávače – hardwarové i softwarové přes síť.
  - Vybrat vhodný hardware k sestrojení zařízení.
  - Vybrat vhodný software a programovací jazyk k vytvoření aplikace.
  - Napsat samotnou aplikaci.
- Vytvořit technickou a uživatelskou dokumentaci k sestrojenému zařízení včetně návodu k instalaci.
- Otestovat stabilitu sestrojeného hudební přehrávač.
- Sdílet zdrojový kód a dokumentaci ve veřejném repozitáři např. na serveru GitHub.



# Kapitola 3

## Metodika

Samotná aplikace se dělí na tři části – serverovou část, která běží na samotném Raspberry Pi, webovou aplikaci, která se spouští ve webovém prohlížeči uživatele a mobilní aplikaci pro systém Android.

### 3.1 Server

Serverová část aplikace má na starost komunikaci s hardwarem pro ovládání a se softwarem pro přehrávání CD a MP3, spuštění webového serveru a komunikaci s webovým rozhraním a mobilní aplikací, čtení informací o MP3 souborech a sledování CD mechaniky a USB portů pro vložení CD či USB flash disku s MP3 soubory.

Operační systém běžící na Raspberry Pi je Raspbian ve verzi Stretch s jádrem Linuxu ve verzi 4.9.41-v7+. Aplikace použitá k přehrávání se jmenuje mpv. Dalšími použitými programy jsou eject, pro ovládání DVD mechaniky, cd-discid pro zjištění informací o hudebním CD, lirc, pro ovládání infračerveným dálkovým ovládáním, amixer, pro ovládání hlasitosti a socat pro komunikaci skriptu s aplikací mpv.

Samotný skript je napsaný v programovacím jazyce Python ve verzi 3.5.3. Použitými knihovnamy jsou mutagen, pro čtení informací o MP3 souborech, Flask, pro spuštění webového serveru, Flask-SocketIO pro oboustranou komunikaci s webovým rozhraním a mobilní aplikací, Pyudev pro sledování linuxového subsystému udev ohledně informací o vložení CD či USB flash disku a Python3-pifacecad pro komunikaci se shieldem PiFace CAD 2.

## 3.2 Webová aplikace

Webová část má na starost ovládání a zobrazování informací o přehrávači. Kromě ovládacích tlačítek a základních informací, jako je název přehrávané skladby nebo její celkový čas, také obsahuje seznam přehrávaných skladeb a v případě MP3 souborů jejich knihovnu. Je napsána pomocí jazyků HTML, sass a JavaScript ve verzi ECMAScript 6, který byl transpilován (přeložen do jazyka na stejné úrovni abstrakce) do verze ECMAScript 5 pro podporu starších prohlížečů.

Pro tuto část byl použit javascriptový frontendový framework React. Dále byly použity moduly fontawesome a react-fontawesome pro pěkné ikonky a socket.io-client pro oboustrannou komunikaci se serverem.

## 3.3 Mobilní aplikace

Mobilní aplikace má stejné funkce jako webový klient, ale navíc obsahuje správu existujících CD přehrávačů a možnost rychle se k nim připojit. Také je s její pomocí možné CD přehrávač ovládat ze zamčené obrazovky a z notifikace, tudíž i z chytrých hodinek.

Mobilní aplikace je napsána pro operační systém Android ve verzi 4.1 a vyšší pomocí jazyka JavaScript ve verzi ECMAScript 6 s využitím frameworku React Native. Použitými moduly jsou react-navigation pro navigaci mezi pohledy a react-native-music-control pro ovládání na zamčené obrazovce a v notifikaci.

## 3.4 Hardware

Přehrávač je realizovaný pomocí malého jednodeskového počítače Raspberry Pi 3. Ke čtení CD a DVD s MP3 soubory je použita USB DVD mechanika Samsung SE-208GB. Hardwarové ovládání je řešeno pomocí GPIO modulu (shieldu) PiFace CAD 2 a infračerveného dálkového ovládání značky Phillips. Do jednoho z USB portů Raspberry Pi je zapojena externí zvuková karta AXAGON USB HQ audio. K napájení celého systému je použit 3A microUSB zdroj.

# Kapitola 4

## Teoretická část

### 4.1 CD

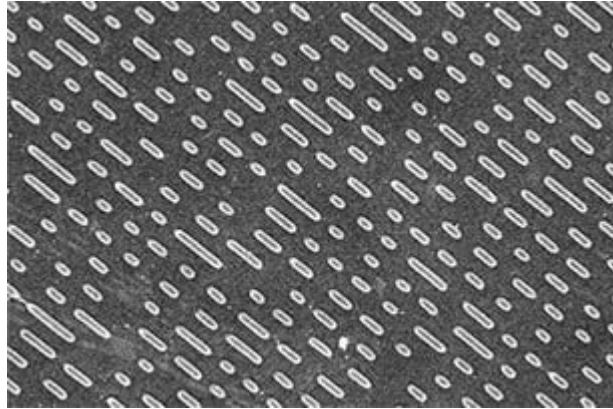
Kompaktní disk (CD) vznikl v sedmdesátých letech 20. století. Po vstupu do digitální éry a představení prvního zařízení pro čtení optických disků začali různé firmy vyvíjet optický disk pro přehrávání hudby. První takový disk byl představen v roce 1977 japonskou firmou Sony. Dva roky poté byl jiný optický disk s průměrem 11,5 cm představen nizozemskou firmou Philips. Spoluprací těchto dvou firem vznikl standardizovaný disk s průměrem 12 cm, který se nazýval compact disc. [1]

Spodní strana disku obsahuje digitální informace ve formě velkého množství miniaturních děr k vidění na obrázku č. 4.1. Tyto informace jsou čteny laserovou diodou, jejíž paprsek je odražen na fotodiodu, která světlo detekuje. Dále je signál převáděn na analogové zvukové informace. Nedochozí tedy k žádnému kontaktu, což zvyšuje životnost CD. [2]

Laserová dioda svítí světlem o vlnové délce 780 nm, což byla nejmenší možná vlnová délka laserových diod existujících v době vytvoření CD. S vlnovou délkou souvisí hustota děr, spodní strana disku obsahuje. Tyto díry jsou 0,6  $\mu\text{m}$  široké, 0,12  $\mu\text{m}$  hluboké a 0,9  $\mu\text{m}$  až 3,3  $\mu\text{m}$  dlouhé. CD plné těchto děr obsahuje 650-700 MB informací, tj. 80 minut (původně 60 minut) zvukového záznamu s bitovou hloubkou 16 bitů. Rychlost otáčení disku v mechanice se u hudebního CD pohybuje od 200 do 500 otáček za minutu. [2]

### 4.2 Raspberry Pi

Raspberry Pi je plnohodnotný počítač velikost kreditní karty, který obsahuje různé porty, přes které se k němu dají připojit periferie. Počítač je dost výkonný na to, aby se na něm dal prohlížet web, přehrávat video či hrát



Obrázek 4.1: spodní strana CD [2]

nenáročné hry. Mimo to ale také obsahuje rozhraní GPIO s 26 nebo 40 piny. Díky tomuto rozhraní lze k Raspberry Pi připojit různé množství hardwaru – senzory, LED diody či celé moduly neboli shieldy. Tyto moduly mohou přidávat další porty, jako například RJ45 či sériový port, mohou být externí zvukovou kartou nebo třeba přidávají displej. [3]

Název počítače vzniklo spojením dvou slov. První z nich bylo zvoleno, protože názvy ovoce mají mezi počítačovými firmami tradici – například značky Apple, Tangerine či Acorn. Druhé slovo vzniklo jako zkomolenina slova Python, což je programovací jazyk, který měl být původně pro Raspberry Pi jediný dostupný. [4, s. 16]

# Kapitola 5

## Výběr hardwaru

Součástí zadání bylo využití jednodeskového počítače Raspberry Pi. K realizaci je použita verze 3, která byla v době zadání práce nejnovější. Všechn hardware využitý k vytvoření CD přehrávače se nachází v tabulce č. 5.1.

Raspberry Pi 3	1 039,00 Kč	<a href="http://rpishop.cz/kategorie/283-raspberry-pi-3-model-b-64-bit.html">http://rpishop.cz/kategorie/283-raspberry-pi-3-model-b-64-bit.html</a>
Karta SanDisk Ultra 16GB microSDHC	229,00 Kč	<a href="http://rpishop.cz/pametove-karty/476-karta-sandisk-16gb-microsdhc-uhs-i-u1-c10-s-sd-adapterem.html">http://rpishop.cz/pametove-karty/476-karta-sandisk-16gb-microsdhc-uhs-i-u1-c10-s-sd-adapterem.html</a>
Samsung SE-208GB	742,00 Kč	<a href="http://rpishop.cz/usb/314-samsung-se-208gb-cerna-externi-dvd-rrw-mechanika.html">http://rpishop.cz/usb/314-samsung-se-208gb-cerna-externi-dvd-rrw-mechanika.html</a>
PiFace CAD 2	999,00 Kč	<a href="http://rpishop.cz/raspberry-pi-prislusenstvi/118-piface-ovladani-displej-v2-pro-model-b.html">http://rpishop.cz/raspberry-pi-prislusenstvi/118-piface-ovladani-displej-v2-pro-model-b.html</a>
AXAGON USB HQ audio	259,00 Kč	<a href="http://rpishop.cz/raspberry-pi-prislusenstvi/114-usb-hq-zvukova-karta-axago-ada-15-mini.html">http://rpishop.cz/raspberry-pi-prislusenstvi/114-usb-hq-zvukova-karta-axago-ada-15-mini.html</a>
3A microUSB napájecí zdroj	279,00 Kč	<a href="http://rpishop.cz/raspberry-pi-prislusenstvi/340-3a-microusb-napajeci-zdroj-cerny.html">http://rpishop.cz/raspberry-pi-prislusenstvi/340-3a-microusb-napajeci-zdroj-cerny.html</a>
<b>Celková cena</b>	<b>3 547,00 Kč</b>	

Tabulka 5.1: použitý hardware, ceny ke dni 30. 10. 2017

## 5.1 DVD mechanika

DVD mechanika pro čtení CD a DVD byla vybírána podle těchto kritérií:

**Velikost mechaniky** Raspberry Pi je malý počítač, bylo by nepraktické použít velkou mechaniku.

**Napájení z USB** K Raspberry Pi existují různé silné zdroje, není tedy třeba mechaniku napájet zvlášť, dokud použijeme dostatečně silný zdroj pro počítač.

**Cena** Cílem je vytvořit relativně levné zařízení.

Všechny DVD mechaniky z nejnižší cenové kategorie (tj. okolo 600–700 Kč) mají podobné vlastnosti, použita byla tedy mechanika Samsung SE-208GB, kterou má v nabídce obchod, kde byl zakoupen i zbytek hardwaru.

## 5.2 Hardwarové ovládání

Jak je zmíněno v teoretické části, k Raspberry Pi GPIO portu se dá připojit různé množství externích modulů. Jeden z typů je takový, který přidává možnosti hardwarového ovládání – obsahuje různé množství tlačítek, joysticků nebo třeba IR čidlo. A právě takovým shieldem je PiFace CAD 2.

### 5.2.1 PiFace CAD 2

PiFace CAD 2 je GPIO modul (shield) pro Raspberry Pi typu A+ a novějších. Zkratka CAD znamená control and display, čili ovládání a displej. Displej je na tomto modulu dvouřádkový a je schopný zobrazit 16 znaků na řádek, což je ideální pro zobrazení informací o přehrávaných skladbách CD přehrávače.

Dále tento modul obsahuje 5 obyčejných tlačítek a 1 třípolohový joystick. Tyto tlačítka se dají použít k ovládání přehrávače. Shield také obsahuje infračervený port pro ovládání libovolným infračerveným dálkovým ovládáním pomocí aplikace LIRC. Pro tuto práci bylo vybráno dálkové ovládání značky Phillips, které dává výrobce ke svým hi-fi věžím. Na použitém dálkovém ovládání ovšem nezáleží, práce je vytvořena tak, aby se dala použít s libovolným ovladačem.

## 5.3 Zvuková karta

K zařízení bylo třeba vybrat nějakou externí zvukovou kartu, protože kvalita zvuku z 3.5mm audio jacku interní zvukové karty není dostačující, především

kvůli jeho citlivosti na rušení – v některých případech dochází ke vzniku šumu.

Externí zvukové karty pro Raspberry Pi existují ve formě GPIO modulu, které dosahují vysoké kvality zvuku, jenže při použití takové karty odpadá možnost využití ovládacího modulu s displejem a tlačítky. Z tohoto důvodu byla vybrána USB karta AXAGON USB HQ audio. Výhodou této karty je také její nízká cena. Stejně jako na dálkovém ovládní ale na zvukové kartě nezáleží, pokud bude uživatel chtít, nemusí kupovat žádnou.

## 5.4 Napájení

Z důvodu využití DVD mechaniky napájené z USB je třeba, kromě správného nastavení Raspbianu, použít dostatečně silný zdroj. Různé internetové zdroje píší, že je třeba použít alespoň 2A zdroj. [5, 6] Pro tuto práci byl vybrán 3A zdroj vytvořený na míru pro obchod RPishop.cz, který není o mnoho dražší než oficiální 2.5A zdroj a navíc dodává jistotu, že výkonu bude vždy dostatek.

# Kapitola 6

## Software

### 6.1 Operační systém

Pro Raspberry Pi existuje několik operačních systémů. Většina z nich jsou distribuce systému Linux, protože při návrhu počítače Raspberry Pi se počítalo s tím, že na něm právě Linux bude fungovat. [4, s. 32] Existuje však i systém od firmy Microsoft Windows 10 IoT Core vytvořený přímo pro Raspberry Pi. Některé tyto systémy jsou plnohodnotnými multimediálními centry, některé obsahují různé balíčky potřebné k chodu Raspberry Pi a další programy vhodné například k výuce či k programování. Takovým systémem je i Raspbian, který je oficiálně doporučovaný a podporovaný Raspberry Pi Foundation. [7] Proto byl právě tento systém vybrán k realizaci této práce.

### 6.2 Audio přehrávač

Důležitou součástí tohoto CD přehrávače je program, který přehraje samotný zvuk z hudebního CD či MP3 souborů. Je důležité, aby se tento program dal ovládat také jiným způsobem než z grafického rozhraní operačního systému – tedy pomocí příkazů v terminálu či pomocí unixového soketu. Tabulka č. 6.1 obsahuje malé porovnání přehrávačů pro potřeby této práce.

Přehrávač	CD	MP3	Programovatelné ovládání
VLC media player	Jen některé	Ano	Unixový soket, příkazy
mplayer	Ano	Ano	Slave mód – pomalý
mpv	Ano	Ano	Unixový soket

Tabulka 6.1: porovnání testovaných přehrávačů



### 6.2.1 VLC media player

Prvním vyzkoušeným přehrávačem byl robustní VLC media player, který umožňuje přehrávat velké množství audio formátů, včetně hudebních CD. [8] Stejně tak tento program obsahuje několik možností, jak ho ovládat z konzole. [9] Zásadní problém je však ten, že VLC někdy nepřehraje některá hudebních CD – většinou se spustí jen první stopa a pak se celý program zasekne a spadne. Při testování se bohužel nepodařilo zjistit, který typ CD takto nelze přehrát, většinou se jednalo o CD starší 20 let, ale nebylo to pravidlem.

### 6.2.2 mplayer

Dalším vyzkoušeným programem byl mplayer, což je především přehrávač videí, ale dokáže přehrávat i zvukové soubory a zvuková CD. Ovládá se čistě z příkazové řádky a obsahuje tzv. slave mód, kde se ovládá příkazy oddělenými symboly nového řádku (`\n`). [10] Bohužel komunikace s tímto módem je pomalá, především při přepínání stop, a ne vždy se povede získat odpověď programu.

### 6.2.3 mpv

Program mpv je fork programu mplayer a tudíž obsahuje většinu jeho funkcí. Mpv se však nevydává za filmový, ale za multimediální přehrávač. Jeho podpora přehrávání audia je tedy lepší, a to včetně dokumentace. Program navíc obsahuje možnost ovládání přes unixový soket, se kterým se dá komunikovat například pomocí programu socat. [11] Tato komunikace na rozdíl od slave módu mplayeru funguje rychle a bezchybně. Právě proto byl vybrán pro realizaci tohoto projektu.

## 6.3 Programovací jazyk

Jak je psáno v teoretické části, druhá polovina názvu počítače Raspberry Pi odkazuje na programovací jazyk Python. Dá se tedy říct, že je to oficiálně doporučený jazyk pro vývoj na tomto počítači. [4, s. 16] Je to vysokoúrovňový jazyk, jehož interpret je implicitně nainstalovaný v operačním systému Raspbian. Pro jazyk Python také oficiálně existuje knihovna pro ovládání modulu PiFace CAD 2, použitého k ovládání a zobrazování informací. Z těchto důvodů byl tento jazyk zvolen pro tvorbu programu, který běží na Raspberry Pi.

Pro psaní webové části není z jazyků moc na výběr, vzhledem k tomu, že dnešní prohlížeče ke skriptování nativně podporují pouze jazyk Javascript.

Z důvodu použití frameworku React je použit javascript ve verzi ECMAScript 6, který je transpilován do verze ECMAScript 5 pro podporu starších prohlížečů, jako je například Internet Explorer 11, [12] který v době psaní práce stále používá 12 % uživatelů. [13]

Vzhledem k využití frameworku React u webové části byl použit k tvorbě mobilní aplikace framework React Native, spolu s jazykem JavaScript.

## 6.4 Další programy

Kromě přehrávače audio souborů je třeba pro správnou funkčnost aplikace použít ještě další programy. Těmi jsou:

**eject** Při spuštění odpojí zařízení ze souborového systému a v případě mechaniky vysune CD. [14]

**cd-discid** Získá informace jako počet stop nebo jejich délky z hudebního CD. [15] Původním plánem bylo použití knihovny pygame, která je v Raspbianu implicitně nainstalovaná, jenže tato knihovna má v sobě chybu, kdy při neúspěšném čtení CD plní velkou rychlostí linuxový syslog.

**socat** Konvertor mezi dvěma datovými toky, kterými může být roura, soubor, socket a další. [16] V práci je použitý ke komunikaci s mpv.

**lirc** Dekóduje nebo posílá infračervené signály a komunikuje s dalšími programy. [17] V práci je využitý pro ovládání infračerveným dálkovým ovládáním za pomoci PiFace CAD 2.

## 6.5 Použité knihovny

### 6.5.1 pyudev

Knihovna pyudev zpřístupňuje API knihovny libudev pro programovací jazyk Python. Libudev je součástí subsystému udev linuxového jádra a má na starost správu zařízení. [18] V této práci je použita ke sledování změn na připojených zařízeních – jednou z takových změn je vložení CD či DVD do mechaniky anebo zapojení USB flash disku.

### 6.5.2 python3-pifacecad

Python3-pifacecad je knihovna, která zpřístupňuje API pro jazyk Python ke komunikaci s modulem PiFace CAD 2, který je používán k hardwarovému ovládání a zobrazování informací o CD přehrávači na displeji.

### 6.5.3 mutagen

Mutagen je knihovna, která má za úkol číst metadata, konkrétně ID3 tagy, ze souborů různých audio formátů, včetně formátu MP3. [19] Zde je použita k přečtení informací o MP3 souborech na CD, DVD nebo USB disku a vytvoření hudební knihovny.

### 6.5.4 Flask

Flask je podle slov tvůrců microframework pro tvoření webových aplikací. Tato knihovna se stará o přijímání a odpovídání na požadavky http protokolu – tedy webové požadavky. [20] V této práci je důležitá pro funkčnost webového rozhraní i mobilní aplikace pro ovládání CD přehrávače.

### 6.5.5 Socket.IO

Knihovna Socket.IO umožňuje jednoduchou obousměrnou komunikaci mezi serverem a webovým prohlížečem. K tomu používá jeden ze dvou protokolů – buď protokol websocket, pokud je podporován, nebo protokol http. [21] Knihovna je primárně určená pro jazyk JavaScript jak na webu, tak na serveru, proto je v práci také použit balík Flask-SocketIO, což je implementace této knihovny pro jazyk Python, konkrétně pro microframework Flask. [22] Socket.IO je důležitá knihovna pro funkčnost webového rozhraní i mobilní aplikace, kde je třeba odesílat data jak z Raspberry Pi ke klientovi (např. informace o přehrávané skladbě), tak zpět (ovládání).

### 6.5.6 React a React Native

React je javascriptová knihovna/framework pro zjednodušení tvorby interaktivních webových uživatelských prostředí. Webové aplikace napsané v Reactu fungují rychleji než běžné, nejavascriptové aplikace, protože odpadá zátěž serveru při každé aktualizaci webové stránky. React funguje na bázi zapouzdřených komponent, takže správně napsaná aplikace by měla být jednoduše znovu použitelná. [23]

React Native je javascriptový framework pro psaní nativních mobilních aplikací pro operační systémy Android a iOS. Kód vypadá jako ve webovém Reactu, ovšem místo HTML a CSS se používají elementy a styly, které vychází z jednotlivých mobilních operačních systémů. Před spuštěním na smartphonu se javascriptový kód kompiluje do nativního kódu operačního systému – tedy Javy u Androidu nebo Objective-C u iOS. [24]

### **6.5.7 Font Awesome**

Balíky fontawesome a react-fontawesome jsou balíky vektorových ikon a CSS stylů. [25] V práci jsou tyto ikony použity ve webovém rozhraní i v mobilní aplikaci.

### **6.5.8 react-native-music-control**

React-native-music-control je knihovna pro React Native, která umožňuje použít mobilní nativní ovládání hudby ze zamknuté obrazovky, z oblasti notifikací nebo třeba z chytrých hodinek. [26] V tomto projektu je využita u mobilní aplikace.

### **6.5.9 react-navigation**

Knihovna react-navigation slouží k vytvoření většího množství pohledů v mobilní aplikaci napsané s pomocí React Native a k navigaci mezi nimi. [27] V této práci je použita v mobilním klientovi pro navigaci mezi úvodním a ovládacím pohledem.

# Kapitola 7

## Praktická část

Tato část práce se zabývá instalací na čisté Raspberry Pi a vysvětlením skriptů, které CD přehrávač řídí – jak serveru, tak klientské části.

### 7.1 Instalace

Prvním krokem po instalaci operačního systému je povolení sběrnice SPI. Ta je potřebná ke správně funkčnosti modulu PiFace CAD 2. To je možné povolit v konfiguračním nástroji raspi-config pod položkou Interfacing options.

```
$ sudo raspi-config
```

Důležitým krokem je zvýšení maximálního limitu elektrického proudu, který dodává USB port, z 0,6A na 1,2A. Bez tohoto navýšení by DVD mechanika napájená z USB nefungovala. Toho se dá dosáhnout přidáním řádku `max_usb_current=1` do souboru `/boot/config.txt`.

```
$ echo 'max_usb_current=1' >> '/boot/config.txt'
```

Dále je třeba nastavit používanou zvukovou kartu. V domovském adresáři uživatele je třeba vytvořit soubor `.asoundrc`, který obsahuje následující řádky, kde číslo za klíčovým slovem `card` je číslo zvukové karty. V případě použití externí USB zvukové karty je toto číslo 1.

```
pcm.!default {
    type hw
    card 1
}
```

```
ctl.!default {
    type hw
    card 1
}
```

Nastavení hlasitosti z webového prostředí či hardwarovými tlačítky po spuštění skriptu neovládá hlasitost systému, ale hlasitost programu mpv, proto je dobré nastavit systémovou hlasitost na 100 %. Toho se docílí následujícím příkazem, kde číslo za operátorem -c je opět číslo zvukové karty.

```
$ amixer -c 1 set Speaker playback 100% unmute
```

Dalším krokem je nainstalovat potřebné balíky pro funkčnost aplikace – jak Linuxové programy, tak knihovny pro Python, které jsou k dispozici v repozitáři.

```
$ sudo apt install mpv eject cd-discid socat lirc
$ sudo pip3 install mutagen flask flask-socketio pyudev
```

Některé knihovny pro Python v době psaní práce nejsou k dispozici v repozitáři systému Raspbian Stretch, ani v repozitáři PyPI jazyka Python, proto je nutné je stáhnout a nainstalovat ručně. Jedná se o balíky python3-lirc, python3-pifacecommon a python3-pifacecad.

```
$ sudo apt install liblircclient-dev cython gcc
$ git clone https://github.com/tompreston/python-lirc.git
$ cd python-lirc
$ make py3 && sudo python3 setup.py install
$ cd ..
$ git clone https://github.com/piface/pifacecommon.git
$ cd pifacecommon
$ sudo python3 setup.py build &&
  sudo python3 setup.py install
$ cd ..
$ git clone https://github.com/piface/pifacecad.git
$ cd pifacecad
$ sudo python3 setup.py build &&
  sudo python3 setup.py install
$ cd ..
```

Pro použití infračerveného dálkového ovládání spolu s modulem PiFace CAD 2 je třeba správně nastavit program lirc, čehož se dosáhne spuštěním

skriptu od tvůrce PiFace CAD 2. V době psaní práce je však tento skript zastaralý, protože program lirc změnil s verzí 0.9.0 syntaxi konfiguračních souborů. Zároveň ale tvůrci programu lirc napsali skript, který starší nastavení převede na nové a v tomto případě funguje dobře. Stažení a spuštění obou těchto skriptů je možné následujícími příkazy.

```
$ wget https://raw.githubusercontent.com/piface/pifacecad/master/bin/
  setup_pifacecad_lirc.sh
$ chmod +x setup_pifacecad_lirc.sh
$ sudo ./setup_pifacecad_lirc.sh
$ wget -O lirc-old2new https://sourceforge.net/p/lirc/git/
  ci/debian/tree/debian/lirc-old2new?format=raw
$ chmod +x lirc-old2new
$ sudo ./lirc-old2new
```

Lirc se ještě musí donastavit pro konkrétní ovladač, předtím ale je nutné Raspberry Pi restartovat. Po restartu je třeba nahrát nastavení ovladače pomocí programu irrecord a přesunout ho do složky s konfiguračními soubory programu lirc. Po nahrání stačí restartovat služby lircd a lircd.socket.

```
$ irrecord remote.lircd.conf
$ sudo mv remote.lircd.conf
  /etc/lirc/lircd.conf.d/remote.lircd.conf
$ sudo rm /etc/lirc/lircd.conf.d/devinput.lircd.conf
$ sudo systemctl restart lircd.socket lircd
```

Předposledním krokem instalace je přidružení tlačítek dálkového ovládání k jednotlivým funkcím CD přehrávače. Toho se docílí vytvoření souboru .lircrc v domovském adresáři. Tento soubor obsahuje několik bloků, kde řetězec za klíčovým slovem prog je název programu, řetězec za klíčovým slovem button je název tlačítka jako z programu irrecord a řetězec za klíčovým slovem config je název funkce CD přehrávače podle vzoru. Vzorový soubor .lircrc je součástí práce a je k dispozici na serveru GitHub [28] a na přiloženém CD.

Posledním krokem je samotné stažení a spuštění CD přehrávače. Na již zmíněném serveru je k dispozici sestavený program připravený ke spuštění. Po stažení stačí program spustit následujícím příkazem.

```
$ python3 main.py
```

## 7.2 Popis kódu

Veškeré zdrojové kódy, včetně stručného návodu pro instalaci a uživatelské dokumentace, jsou k nalezení na serveru GitHub [28] a na přiloženém CD.

### 7.2.1 Server

Jak již bylo řečeno, serverová část běžící na Raspberry Pi je psána v programovací jazyku Python. Samotný kód je rozdělen do šesti souborů.

**main.py** Vstupní bod programu, spouští a nastavuje přehrávač a web server, zprostředkovává komunikaci s webovým rozhraním.

**MediaPlayer.py** Obsahuje třídu MediaPlayer, která obsahuje hlavní logiku pro ovládání programu mpv a získávání informací o disku. Dále obsahuje třídu CD, která reprezentuje DVD mechaniku a disk v ní obsažený.

**MediaPlayerConfig.py** Obsahuje stejnojmennou třídu, která parsuje a reprezentuje nastavení programu

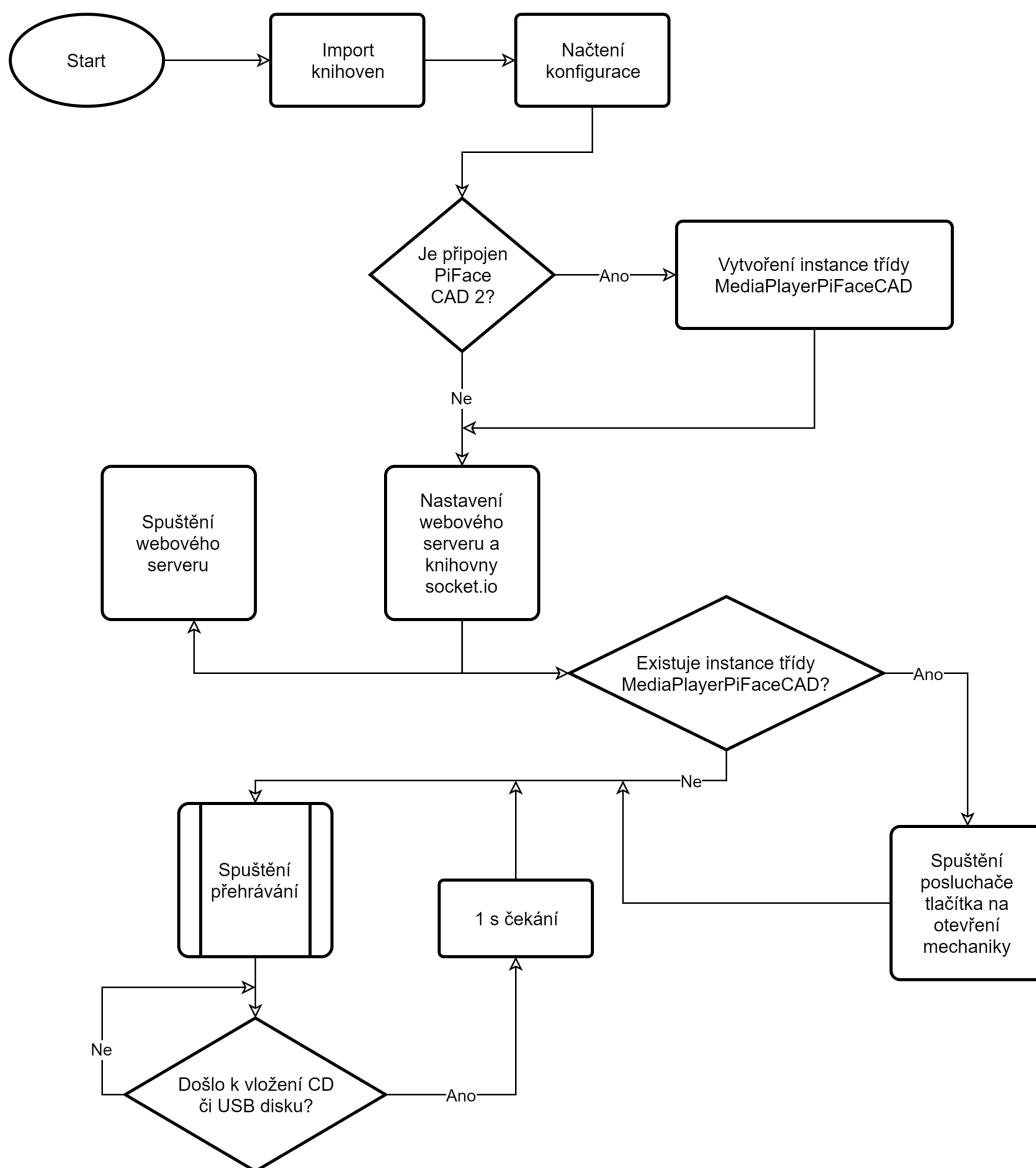
**MediaPlayerInfo.py** Obsahuje třídy MediaPlayerInfo, TrackInfo a CurrentTrackInfo, které jsou spíše datovými strukturami a reprezentují předávané informace o přehrávání. Těmi jsou stav přehrávače, hlasitost, přehrávaná skladba, seznam skladeb a hudební knihovna MP3 souborů.

**MediaLibrary.py** Obsahuje stejnojmennou třídu, která vytváří a reprezentuje hudební knihovnu MP3 souborů.

**MediaPlayerPiFaceCAD.py** Obsahuje třídu, která se stará o komunikaci s modulem PiFace CAD 2.

Životní cyklus programu se dá popsat vývojovým diagramem zobrazeným na obrázku č. 7.1.





Obrázek 7.1: životní cyklus programu

## Načtení konfigurace

Konfigurace aplikace se nachází v souboru `media_player.conf`. V něm je každé nastavení na jednom řádku, oddělené pomocí znaku `=`. Tento soubor je rozparsován do instance třídy `MediaPlayerConfig`, která je dále používána ve zbytku skriptu. Při parsování jsou ignorovány prázdné řádky, řádky začínající znakem `#`, neboli komentáře, a řádky se špatnou syntaxí. Konstruktor této třídy a parsovací algoritmus je k vidění v následujícím bloku.

```

def __init__(self, file_path):
    self._config_dict = {}
    with open(file_path) as file:
        lines = file.read().splitlines()
        for line in lines:
            if line.startswith('#'):
                continue
            ar = line.split('=')
            try:
                self._config_dict[ar[0]] = ar[1]
            except IndexError:
                # syntax error, ignore the line
                pass

```

## Nastavení webového serveru a knihovny socket.io

Microframework Flask je nutné před spuštěním správně nastavit. Ve skriptu je třeba nastavit správnou složku, ve které se nachází šablony, a složku, kde jsou statické soubory (tj. CSS styly, javascriptové soubory), a url cestu k ní. To je třeba kvůli použití frontendového frameworku React, který má vlastní strukturu složek a s Flaskem implicitně nefunguje. Flask defaultně vypisuje do konzole veškeré požadavky. To je zde nežádoucí, proto je logování nastaveno pouze na výpis chyb. Před spuštěním je ještě třeba definovat routy a funkce, které se zavolají při nějaké události knihovny socket.io – tedy při stisknutí různých ovládacích prvků na webu. Webový server je nutné spustit v jiném vlákně, jinak dojde k zastavení skriptu. Ten je ale nutné nechat běžet dál, kvůli přehrávání disků. Zkrácený kód popsany v této části je možné vidět zde:

```

# Web server configuration
app = Flask(__name__, template_folder="web",
            static_folder="web/static",
            static_url_path="/static")
app.debug = False
log = logging.getLogger('werkzeug')
log.setLevel(logging.ERROR)
socket = SocketIO(app, async_mode='threading')
...
@app.route('/')
def index():
    return render_template('index.html')
...

```

```

@socket.on('volumeUp')
def ws_volume_up():
    media_player.volume_up()
...
# Web server thread starting point
def start_web_server():
    if __name__ == '__main__':
        socket.run(app, config['WEB_IP'],
                   port=config['WEB_PORT'])

# Start web server thread
web_server_thread = Thread(target=start_web_server,
                           args=[])
web_server_thread.setDaemon(True)
web_server_thread.start()

```

## Posluchače tlačítek Pi Face CAD2

Knihovna python3-pifacecad, dodávaná k modulu Pi Face CAD 2, obsahuje kromě samotné třídy k ovládání displeje a čtení stavů tlačítek také různé pomocné třídy. Jednou z těchto tříd je `pifacecad.core.SwitchEventListener`. Tato třída umožňuje nastavit sledování stavu tlačítek a volat funkce (callbacky) na jejich změnu (např. stisknutí). Tento posluchač funguje na vlastním vlákně a dá se kdykoliv vypnout.

Chvíli po spuštění skriptu je jeden takový posluchač aktivován. Sleduje tlačítko, které zastavuje přehrávání. Jedná se totiž o jediné tlačítko, které je potřebné ještě před samotným přehráváním zvuku. Po stisku volá funkci `stop()` instance třídy `MediaPlayer`, která, pokud neběží přehrávání, otevře mechaniku.

```

@staticmethod
def create_eject_listener(media_player):
    eject_listener = pifacecad.SwitchEventListener()
    eject_listener.register(4, pifacecad.IODIR_ON,
                           lambda event: media_player.stop())
    eject_listener.activate()
    return eject_listener

if cad is not None:
    eject_listener = MediaPlayerPiFaceCAD.create_eject_listener(
        media_player)

```

## Sledování subsystému udev

Po první spuštění se program pokusí spustit přehrávání. Pokud se to povede, tak se zvuk normálně přehrává, dokud není uživatelem přehrávání zastaveno. Spuštění se ale také povést nemusí – pokud není vložen žádný disk nebo pokud nejsou nalezeny žádné zvukové stopy. Ať už se první přehrávání nepovede, či povede a skončí, tak skript začne sledovat subsystém udev pro změny. Pokud by došlo ke změně typ change (vlození CD či DVD do mechaniky) nebo add (připojení USB flash disku), tak se program znovu pokusí přehrát disk, jinak na takovou změnu čeká. Toto sledování zajišťuje knihovna pyudev. Před vyzkoušením přehrání skript ještě sekundu čeká, než dojde v systému ke všem potřebným úkonům – hlavně k připojení disku do souborového systému. Pokud by nečekal, tak se přehrávání nemusí vždy povést. Z testování vyšlo, že 1 sekunda je dostatečně dlouhá doba.

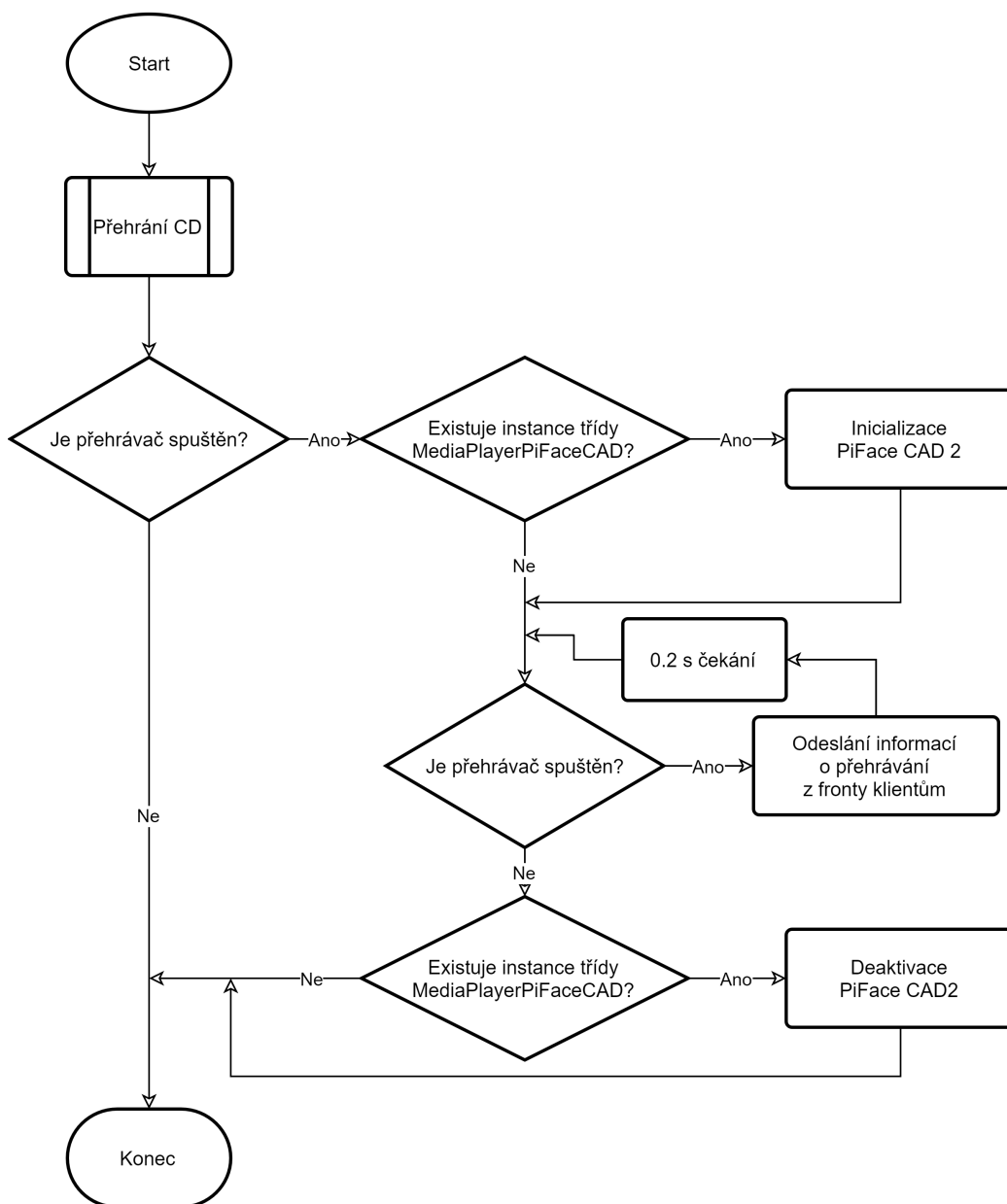
```
udev_context = pyudev.Context()
udev_monitor = pyudev.Monitor.from_netlink(udev_context)
udev_monitor.filter_by(subsystem='block')
for device in iter(udev_monitor.poll, None):
    if device.action == 'change' or device.action == 'add':
        sleep(1)
        play_cd(media_player, cad)
```

Druhou možností, jak zjistit, že bylo vloženo CD, je kontrolování obsahu mechaniky ve while cyklu např. pomocí knihovny pygame. Při tomto způsobu však dochází s každým dotazem na prázdnou mechaniku k chybě, která sice nevyvolá výjimku, ale zapíše se do souboru /var/log/syslog. Pokud by pak program běžel nějakou dobu, syslog svou velikostí zaplní celou paměť Raspberry Pi, což je velmi nepraktické.

## Spuštění přehrávání

Jak je psáno výše, přehrávání se buď může nebo nemusí povést. Pokud se povede, tak skript odesílá informace skrz knihovnu socket.io a čeká, dokud uživatel přehrávání nevypne. Proces přehrávání je popsán vývojovým diagramem na obrázku č. 7.2.

Pokud se povede spuštění přehrávání, dochází k inicializaci modulu PiFace CAD 2 (pokud je připojen). Dále se spustí while cyklus, ve kterém se čtou informace předávané třídou MediaPlayer a předávají klientům. Tyto informace jsou uloženy ve frontě. Na konci každé iterace cyklu je program na 0.2 s uspán, aby nedocházelo ke zbytečnému přetěžování procesoru. Po ukončení cyklu se deaktivuje modul s displejem a ovládáním a dojde k ukončení přehrávání.



Obrázek 7.2: proces přehrávání

### Inicializace PiFace CAD 2

Inicializací je myšleno spuštění a nastavení displeje, vytvoření posluchačů tlačítek a infračerveného signálu a spuštění dvou vláken. V jednom vlákně dochází k výpisu informací na displej a ve druhém se čeká na vypnutí modulu zvenku metodou destroy třídy MediaPlayerPiFaceCAD.

Po zavolání této funkce dojde k deaktivaci displeje a posluchačů, které nelze deaktivovat z vlákna na kterém poslouchají (např. při stisknutí tlačítka na dálkovém ovládní). Toho je docíleno pomocí objektu typu `Threading.Barrier`. Ten dokáže kód pozastavit, dokud určitý počet vláken nezavolá funkci `wait` nebo není resetován. V tomto případě dochází zavoláním metody `destroy` k resetování bariéry, protože při stisku tlačítka nechceme nikdy čekat. Při resetování bariéry však dochází k vyvolání výjimky, kterou je třeba odchytit. Tento proces je k vidění v následujícím bloku kódu.

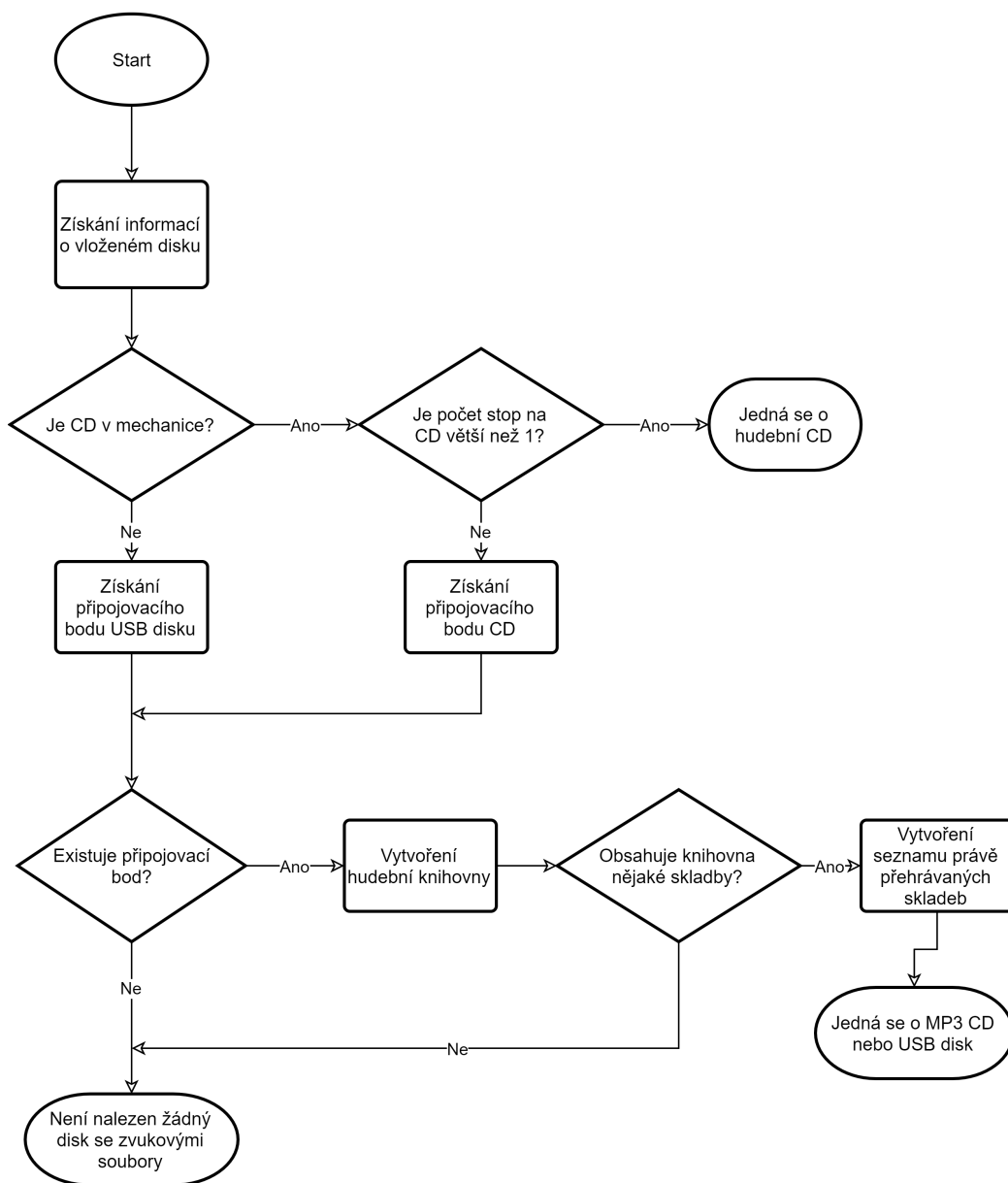
```
self._listeners_barrier = Barrier(2)
...
# v~čekacím vlákně
try:
    self._listeners_barrier.wait()
except BrokenBarrierError:
    pass # expected
self._switch_listener.deactivate()
self._ir_listener.deactivate()
...
def destroy(self):
    self._cad.lcd.clear()
    self._cad.lcd.backlight_off()
    self._listeners_barrier.reset()
```

Při výpisu textu na displej, ke kterému dochází každou vteřinu, se na první řádek napíše interpret a název přehrávání skladby a na druhý čas a číslo stopy a jejich celkový počet. Při pozastaveném přehrávání se na prvním řádku zobrazí řetězec „Paused“ a při změně stavu přehrávače (např. změna stopy) dojde k dočasnému zobrazení této informace. Při vypsání většího počtu znaků na displej může dojít k chybnému zobrazení, proto program vypisuje vždy právě 16 znaků na řádek (ačkoliv autoři tvrdí, že by to nemělo vadit).

## Přehrání CD

Ke spuštění přehrávače dochází zavoláním metody `try_play_cd` třídy `MediaPlayer`. Zde dojde k vytvoření fronty, do které se vkládají informace o přehrávání. Ke vložení informace dojde při změně stavu přehrávače, tj. při změně přehrávané stopy, souboru, změně hlasitosti, přepnutí času přehrávání či při pozastavení a spuštění přehrávání. Následně se zkontroluje mechanika a USB port, zda obsahuje nějaké CD nebo USB disk a typ disku se uloží do jednoho z atributů třídy `MediaPlayer`. Vývojový diagram ke zjištění typu disku je k vidění na obrázku č. 7.3.

Pokud by nedošlo k nalezení žádného disku, tak k přehrání nedojde. Podle typu disku dojde ke spuštění přehrávače – programu mpv. Typ disku může být buď AUDIO\_CD, při klasickém hudebním CD, nebo MP3\_CD, který je používán i při přehrávání z USB disku. Při přehrávání MP3\_CD může být přehráváno buď podle složek obsahujících MP3 soubory nebo podle hudební knihovny interpretů a jejich alb. Druh přehrávání MP3\_CD je uložen a kontrolován při přepínání stop. Pokud dojde k přehrání, do fronty se uloží informace obsahující knihovnu a seznam právě přehrávaných skladeb.



Obrázek 7.3: přehrání CD

### Informace o disku

K získání informací o vloženém disku je používán program cd-discid. Tento program primárně vrací ID souboru potřebné k vyhledávání v CDDb databázích. Toto ID obsahuje kontrolní součet stop, jejich počet a jejich počátky ve timecode framech, což je 1/75 vteřiny. Z těch se dají vypočítat délky



jednotlivých stop.

```
discid = subprocess.getstatusoutput(
    'cd-discid --musicbrainz')
if discid[0] == 0:
    output_split = discid[1].split()
    self._numtracks = int(output_split[0])
    track_offsets = list(map(lambda i: int(i),
        output_split[1:]))
    self._track_lengths = list(
        map(lambda i, offsets=track_offsets:
            int((offsets[i + 1] - offsets[i]) * 1000 / 75),
            range(0, self._numtracks)))
```

### MP3 disk

Pokud program `cd-discid` říká, že CD obsahuje pouze 1 stopu, pravděpodobně se nejedná o hudební CD, ale o CD datové. V tomto případě (a také v případě, že není vložené CD, ale USB disk) je třeba získat připojovací bod (mount point). K tomu je použit program `df`, který vrací informace o všech zařízeních připojených v souborovém systému.

```
df = subprocess.getoutput('df | grep ' +
    self._config['CD_DEVICE']).split()
if len(df) > 0:
    mount_point = ' '.join(df[5:])
```

### Hudební knihovna

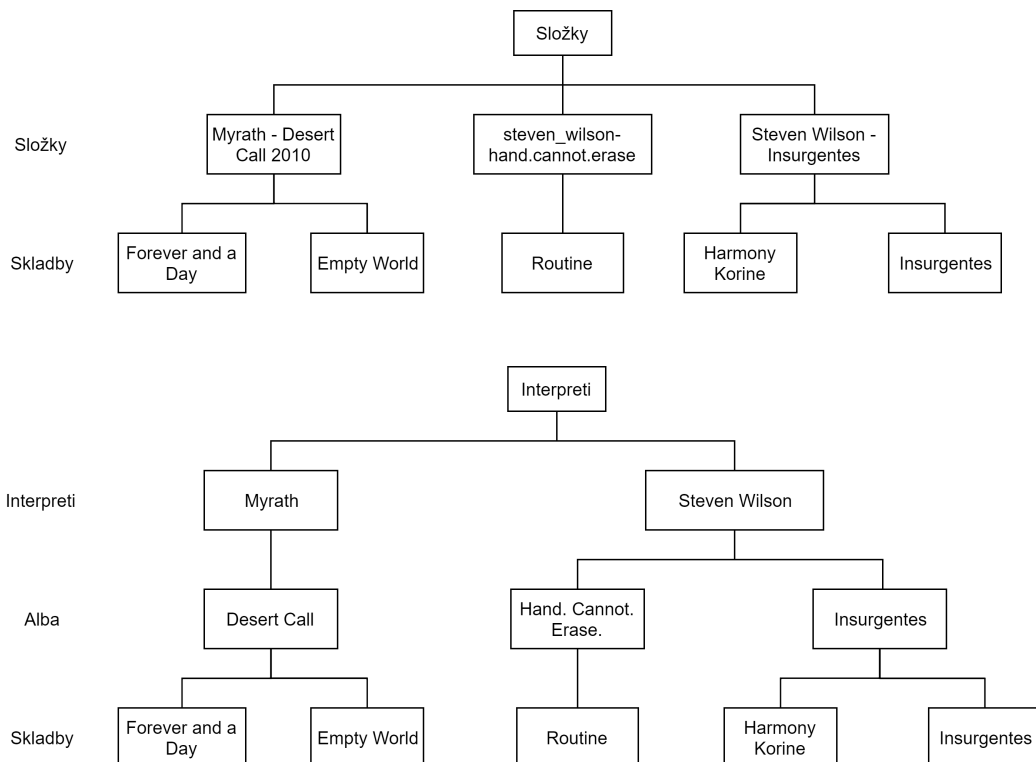
V případě nalezení připojovacího bodu je vytvořena a inicializována instance třídy `MediaLibrary`. Ta reprezentuje hudební knihovnu s MP3 soubory. Při inicializaci prohledá všechny soubory v dané složce (připojovacím bodě) a její podsložkách a najde všechny soubory s příponou `mp3`. Z těchto souborů následně přečte, pomocí knihovny `mutagen`, informace o interpretech a albech a vytvoří 2 stromy skladeb – strom složek obsahující `mp3` soubory a strom interpretů a jejich alb. Rozdíl mezi těmito stromy je vidět na obrázku č. 7.4.

```
for subdir, dirs, files in os.walk(self._root_folder):
    media_folder = MediaLibrary.MediaFolder(subdir)
    for file in files:
        # odebráno odsazení pro přehlednost
        if file.endswith('.mp3'):
```

```

media_file = MediaLibrary.MediaFile(file, subdir)
media_file.init_tags()
if media_folder not in self._media_folders:
    self._media_folders.append(media_folder)
media_folder.add_media_file(media_file)
artist = next((x for x in self._artists if x.name ==
               media_file.artist), None)
if artist is None:
    artist = MediaLibrary.Artist(media_file.artist)
    self._artists.append(artist)
album = next((x for x in artist.albums if x.name ==
              media_file.album), None)
if album is None:
    album = MediaLibrary.Album(media_file.album)
    artist.add_album(album)
album.add_song(media_file)
self._media_file_count += 1

```



Obrázek 7.4: stromy hudební knihovny se vzorovými daty

## Ovládání

Třída `MediaPlayer` obsahuje několik metod, které slouží pro ovládání přehrávače. Tyto metody jsou volány při stisknutí tlačítek na modulu `Pi Face CAD 2`, při přijetí infračerveného signálu z dálkového ovládání a při přijetí příkazu z webového a mobilního rozhraní, pomocí knihovny `socket.io`. V případě metod jako `play_pause` (pozastavení a spuštění přehrávání), `next_track` (další skladba) nebo `volume_down` (snížení hlasitosti) dochází pouze k předání daného příkazu programu `mpv` a poslání informace klientům. Metoda `stop` vypne přehrávač a vysune CD z mechaniky (v případě USB disku ho odpojí ze souborového systému). Metody `next_branch` a `prev_branch` v případě přehrávání MP3 přepínají složky, interprety nebo alba. Pokud je přehráváno hudební CD, tak pouze přepínají stopy.

Webové a mobilní rozhraní obsahují větší možnosti ovládání – umožňují přepnout čas přehrávání jedné stopy na konkrétní čas, k čemuž slouží metoda `seek`. Dále zobrazují celou hudební knihovnu a seznam stop. V nich je možné přepínat skladby, interprety či alba libovolně. K tomu jsou metody `play_track`, které je předáno číslo stopy ze seznamu stop, a `play_file`, které je předán typ přehrávání (složky, interpreti, alba) a index dané větve a skladby z knihovny. Metoda nahraje do seznamu stop skladby, které byly uživatelem vybrány, a to v takovém pořadí, aby byly vždy přehrány všechny skladby v albu či složce. Pokud je tedy v albu pět skladeb a uživatel chce přehrát třetí, postupně se přehrají skladby 3, 4, 5, 1, 2.

```
ordered_files = files[file_index:] + files[0:file_index]
self._current_track_list =
    list(map(lambda media_info:
              TrackInfo(media_info.total_time,
                        media_info.artist,
                        media_info.album,
                        media_info.title),
            ordered_files))
self._run_command('playlist-clear')
self._run_command('loadfile', files[file_index].full_path)
for file in ordered_files[1:]:
    self._run_command('loadfile', file.full_path, 'append')
```

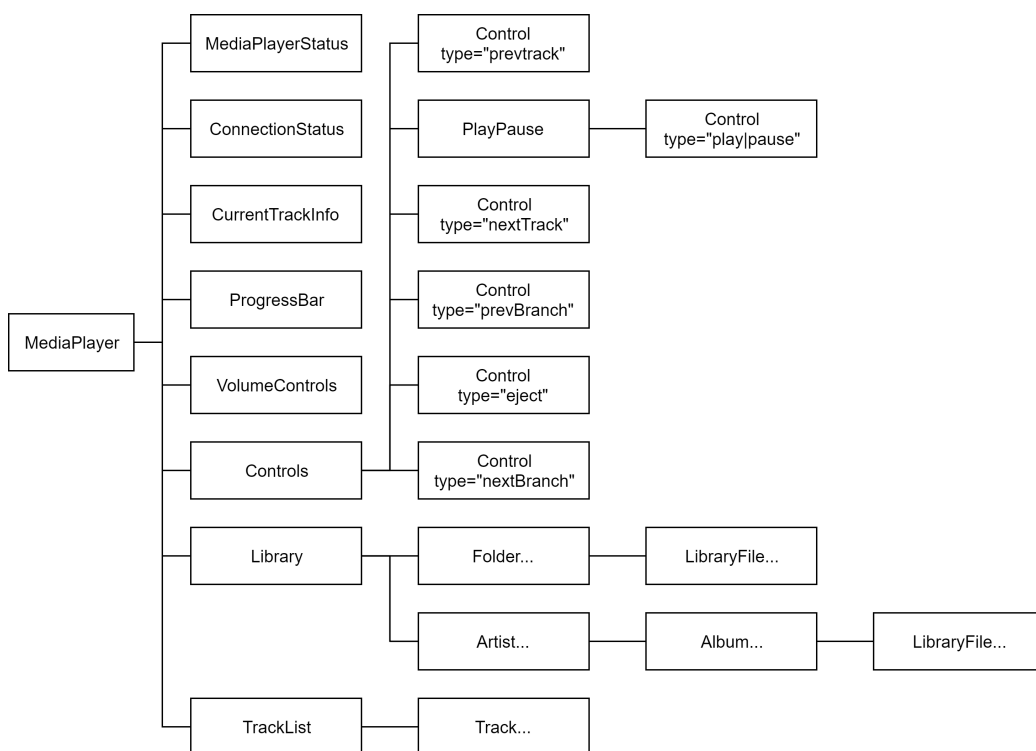
### 7.2.2 Webový klient

Webové ovládací rozhraní přehrávače je vytvořeno za pomoci frameworku `React`. V `Reactu` se vytváří jednotlivé komponenty, které se pak skládají

dohromady pomocí jazyka JSX, čímž vzniká samotná aplikace. JSX je rozšíření JavaScriptu, které používá podobnou syntaxi jako jazyk XML. Proto se dají komponenty aplikace zobrazit jako strom, který je k vidění na obrázku č. 7.5. Samotné komponenty pak obsahují HTML kód.

Webové rozhraní je plně responzivní, dá se tedy zobrazit na smartphonu, tabletu i stolní PC či notebooku. Funguje ve všechno moderních prohlížečích. Screenshots rozhraní na různých zařízeních a jejich popis jsou k dispozici v přílohách.

V základním nastavení běží webový server na portu 51234.



Obrázek 7.5: strom React komponent webového rozhraní

## MediaPlayer

MediaPlayer je hlavní komponenta aplikace. V konstruktoru je vytvořeno připojení pomocí klientské verze knihovny socket.io. Toto připojení je dále předáváno všem dalším komponentám. Každá z nich sama sleduje připojení a při příchodu informací o přehrávání se sama aktualizuje. Komponenta MediaPlayer také přesměrovává předávané příkazy z mobilní aplikace na server. K tomu více v další části práce.

## Status

Status je komponenta, která zobrazuje stav přehrávače. Dědí z ní další dvě komponenty, kterými jsou MediaPlayerStatus a ConnectionStatus. MediaPlayerStatus může nabývat tří hodnot – „Playing media“, když probíhá přehrávání, „Media paused“, když je přehrávání pozastavené a „Waiting for CD“, když server čeká na vložení disku. ConnectionStatus zobrazuje 5 stavů – „Not connected“, když klient není připojen (při spuštění), „Connected“, pokud je webový klient připojen k serveru, „Reconnecting“, pokud dojde k výpadku a klient se pokouší znovu připojit, „Disconnected“, když je klient odpojen a „Connection error“, při chybě při připojování.

## CurrentTrackInfo

Tato komponenta zobrazuje informace o právě přehrávané skladbě – její název, interpreta, album, čas přehrávání a číslo stopy. Tyto údaje jsou touto komponentou předávány také mobilní aplikaci. Aby nedocházelo k odesílání informací ze serveru každou vteřinu, webové rozhraní si samo hlídá čas skladby. Pokud by mělo dojít k situaci, že by současný čas skladby byl vyšší než celkový, tak si klient automaticky zažádá o nové informace.

```
this.interval = setInterval(() => {
  this.setState({
    curTime: this.state.curTime + 100
  });
  if (this.state.curTime > trackInfo.totalTime) {
    clearInterval(this.interval);
    this.props.socket.emit('getCurTrackInfo');
  }
}, 100);
```

## ProgressBar

ProgressBar souvisí s předchozí komponentou, graficky zobrazuje čas skladby. Stejně jako CurrentTrackInfo si sám hlídá čas a v případě většího času, než je čas celkový, dojde k požadavku na server. Zároveň je možné do této komponenty kliknout a přepnout se na danou část stopy.

## Controls

Ovládání přehrávače je dále zprostředkováváno komponentami VolumeControls a Controls. VolumeControls zobrazuje a umožňuje ovládat hlasitost

programu mpv. Controls obsahuje několik dalších komponent typu Control, které po kliknutí odesílají požadavek na server (podle typu). PlayPause je speciálním typem ovládání, protože podle stavu přehrávání mění ikonku a typ. Ovládací prvky jsou z důvodu responzivity na mobilním telefon v jiném pořadí než na tabletu a počítači. Toho je docíleno „přehazováním“ pořadí pomocí CSS atributu margin-left.

## Library a TrackList

Komponenta Library zobrazuje knihovnu MP3 souborů na CD nebo USB disku. Obsahuje několik komponent Folder (složka) a Artist (interpret), které obsahují komponenty Album a LibraryFile (skladba). Po otevření složky se zobrazí skladby, po otevření interpreta se objeví jeho alba a pak až soubory. Při kliknutí na ikonku přehrávání u složky, interpreta nebo alba je jejich „obsah“ přehrán. Při kliknutí na skladbu je přehrána daná skladba a do seznamu stop se vloží buď zbytek alba nebo složky.

TrackList reprezentuje seznam právě přehrávaných stop. Pokud si například uživatel v knihovně vybere, aby bylo přehráno nějaké album, tak se zde objeví pouze obsah tohoto alba. V seznamu je zvýrazněna právě přehrávaná stopa a uživatel zde může stopu kdykoliv změnit.

Výška knihovny a seznamu stop je na počítači vždy taková, aby se vešla na obrazovku, minimálně však 200px (k vidění v následujícím bloku). V případě tabletu a mobilního telefonu se obě komponenty skryjí a objeví se tlačítko na zobrazení. Otevírání a zavírání je docíleno pouze pomocí CSS3 animací, bez použití JavaScriptu.

```
updateHeight() {
  let library = this.refs.library;
  if (window.innerWidth >= 1024) {
    library.style.height = window.innerHeight -
      library.offsetTop + 'px';
  } else {
    let height = window.innerHeight;
    for(let node of
      library.parentNode.childNodes) {
      if(node === library) continue;
      height -= node.offsetHeight;
    }
    library.style.height = height + 'px';
  }
}
```

```

...
componentDidMount() {
  window.addEventListener('resize',
    this.updateHeight.bind(this));
}
...
.library,
.track-list {
  min-height: 200px !important;
}

```

### 7.2.3 Mobilní klient

Po mobilní aplikaci bylo požadováno:

- Stejná funkčnost jako u webového klienta
- Možnost ukládání existujících přehrávačů, aby nebylo nutné pokaždé zadávat IP adresu Raspberry Pi do prohlížeče
- Možnost ovládání skrz notifikaci, a tudíž i z chytrých hodinek a ze zamčené obrazovky

Aplikace je psána v JavaScriptu za pomoci frameworku React Native. Pracuje se s ním podobně jako s webovým Reactem, je totiž také založen na komponentách. Komponenty však neobsahují HTML kód, ale nativní prvky mobilního systému. V této aplikaci jsou vytvořené komponenty pouze dvě, a to sice HomeScreen, což je úvodní obrazovka aplikace a MediaPlayerScreen, což je obrazovka s ovládáním přehrávače.

Vzhledem k vysoké rychlosti webového klienta a skutečnosti, že se jedná o jednu stránku, byla k tvorbě využita nativní komponenta WebView. Ta dokáže v aplikaci zobrazit libovolnou webovou stránku a komunikovat s ní. Tím je docíleno stejné funkčnosti jako u webové aplikace.

Po spuštění aplikace se objeví seznam, do kterého je možné přidat jednotlivé přehrávače, za podmínky, že uživatel zná IP adresu a port, na kterém webový klient běží. Kliknutím na položku v tomto seznamu je možné záznam upravit, odstranit nebo se připojit na daný server. Před připojením se nejdříve aplikace ujistí že se jedná o přehrávač požadavkem na url /getMediaPlayerInfo, který vrací název zařízení z konfigurace serveru. Následně se k přehrávači připojí a zobrazí webové rozhraní skrz WebView. Ukládání a načítání tohoto seznamu je zprostředkováno pomocí třídy AsyncStorage.

```

let ip = this.state.ipList[index];
const uri = 'http://' + ip.ip + ((ip.port !== '') ?
    (':' + ip.port) : '');

const {navigate} = this.props.navigation;
fetch(uri + '/getMediaPlayerInfo').then((response) => {
  if (response.ok) {
    // save last connection
    try {
      AsyncStorage.setItem('@IPList:lastIndex',
                           index.toString());

      this.setState({
        lastIndex: index
      })
    }
    catch (error) {
      alert(error);
    }
    this.closeIPModal();
    navigate('MediaPlayer', {
      uri: uri
    })
  } else {
    showError()
  }
}).catch((error) => showError());

```

Ovládání skrz notifikaci je zařízeno za pomoci knihovny react-native-music-control. Ta obsahuje třídu s několika statickými metodami. Zde jsou použité následující metody.

**enableControl** Povolí jednotlivé ovládací prvky v oznámení, volá se při spuštění aplikace.

**setOnPlaying** Zobrazí notifikaci s předanými informacemi a ovládním, volá se při spuštění přehrávání a změně skladby.

**updatePlayback** Aktualizuje notifikace novými informacemi, volá se při změně stavu přehrávače (pauza).

**resetNowPlaying** Skryje notifikaci, volá se při ukončení přehrávání.



**on** Vytvoří posluchače na stisknutí daného ovládacího prvku, volá se při připojení k serveru.

Při stisknutí tlačítka v oznámení se spustí callback funkce, která komunikuje s webovou stránkou zobrazenou ve WebView. Webová část poslanou zprávu přijme a přepošle na server, kde dojde ke stejnému procesu jako při stisknutí daného tlačítka na webu.

K aplikaci byla vytvořena jednoduchá ikona a následně byla podle návodu na webu React Native zkompileována do apk souboru pro systém Android. [24] Ten je k nalezení na GitHubu [28] a přiloženém CD.

## 7.3 Testování

Testování přehrávače proběhlo od samotného začátku – tedy od čisté instalace Raspbianu. Pak bylo postupováno podle návodu k instalaci, který je popsán v kapitole 7.1 Instalace. Během ní nedošlo k žádnému problému. Všechny potřebné programy a knihovny byly nainstalovány a aplikace lirc byla nastavena pro dálkové ovládání značky Phillips.

Při testování funkčnosti ovládání byla postupně několikrát vyzkoušena všechna tlačítka na všech ovládacích rozhraních – tedy hardwarová tlačítka, dálkové ovládání, webový a mobilní klient, včetně funkčnosti ovládání z oznámení na chytrých hodinkách, které byly se smartphonem propojeny. Jediný zaznamenaný problém byl s dálkovým ovládáním, na které Raspberry Pi občas nereagovalo. Bohužel v tomto chování nebyl nalezen žádný vzor, a proto se chyba nedá cíleně zreprodukovat.

Při testování ovládání byla také sledována synchronizace mezi jednotlivými zobrazeními na displeji a ve webovém a mobilním klientovi. Po změně skladby, hlasitosti či jiného stavu přehrávače nedocházelo téměř k žádné prodlevě zobrazení informací mezi displejem a dalšími klienty. Jediným malým problémem bylo občasné špatné zobrazení času na webu a v mobilu, kdy se při změně skladby již začal počítat čas, ale skladba ještě nebyla načtena. Zhruba po jedné vteřině však došlo k synchronizaci, takže na webu a v mobilu problikl čas 00:01 a vrátil se zpět na 00:00.

Testování stability probíhalo v časovém rozmezí 6 hodin, kdy bylo v přehrávači vyměněno 5 hudebních CD z různých časových období, 2 MP3 CD, 2 USB disky obsahující hudbu a 1 USB disk, který hudbu neobsahoval. Během používání přehrávače a měnění disků nedošlo k žádným závažným chybám a program ani Raspberry Pi nebylo třeba znovu spouštět. Při přehrávání poškrábaného CD však občas došlo k vypnutí přehrávání. V případě špatného MP3 CD se po chvíli přehrávání znovu spustilo, avšak od začátku. U poškrábaného hudebního

CD bylo třeba CD odebrat z mechaniky a znovu vložit. Při přehrávání hudby z USB disku nedošlo k žádným problémům. Při vložení disku bez hudby se přehrávání nespustilo a aplikace dále sledovala stav DVD mechaniky.

# Kapitola 8

## Závěr

Všechny cíle bakalářské práce, tedy sestavení, tvorba dokumentace a testování, byly splněny a popsány. Byl vytvořen hudební CD a MP3 přehrávač za pomoci počítače Raspberry Pi 3. Přehrávač je možné ovládat několika způsoby, jak hardwarově, tak softwarově po síti. Zařízení bylo otestováno a při správném nastavení je výsledek vhodný ke každodennímu použití. Veškeré zdrojové kódy, návod na instalaci a uživatelská dokumentace je k dispozici pod open-source licencí na serveru GitHub [28] a na přiloženém CD.

Budoucí vývoj přehrávače je reálný. Vzhledem k použití DVD mechaniky je možné přidat funkci přehrávání DVD filmů. Dále by šlo přidat možnost přehrávat MP3 soubory z vnitřní paměti Raspberry Pi nebo z mediálního serveru. Také by bylo možné přidat funkci přehrávání ze streamovacích služeb jako je Spotify.

Zařízení je také rozšiřitelné pro použití jiného hardwaru. Počítač Raspberry Pi 3 je dost výkonný k přehrávání BluRay disků, takže připojení BluRay mechaniky k přehrávání hudby (či filmů, po úpravě softwaru) je možné. Program funguje i bez modulu PiFace CAD 2 a proto jde místo něj použít například výkonnější zvukovou kartu pro lepší kvalitu zvuku.

# Literatura

- [1] The history of the CD - The beginning. Research |Philips [online]. [cit. 2017-11-12]. Dostupné z: <https://www.philips.com/a-w/research/technologies/cd/beginning.html>
- [2] The history of the CD - Technology. Research |Philips [online]. [cit. 2017-11-12]. Dostupné z: <https://www.philips.com/a-w/research/technologies/cd/technology.html>
- [3] GPIO moduly. RPishop.cz [online]. [cit. 2017-10-30]. Dostupné z: <http://rpishop.cz/148-gpio-moduly>
- [4] UPTON, Eben a Garoty HALFACREE. Raspberry Pi: uživatelská příručka. 2., aktualizované vydání. Přeložil Jakub GONER. Brno: Computer Press, 2016. ISBN 9788025148198.
- [5] Testing & Setting the USB current limiter on the Raspberry Pi B+ [online]. [cit. 2017-10-28]. Dostupné z: <https://projects.drogon.net/testing-setting-the-usb-current-limiter-on-the-raspberry-pi-b/>
- [6] Is setting 'max\_usb\_current=1' to give more power to USB devices a bad idea? [online]. [cit. 2017-10-28]. Dostupné z: <https://raspberrypi.stackexchange.com/a/27742>
- [7] Raspberry Pi Downloads. Raspberry Pi [online]. [cit. 2017-10-28]. Dostupné z: <https://www.raspberrypi.org/downloads/>
- [8] VLC - Features. VideoLAN [online]. [cit. 2017-10-28]. Dostupné z: <https://www.videolan.org/vlc/features.html>
- [9] Console. VideoLAN Wiki [online]. [cit. 2017-10-28]. Dostupné z: <https://wiki.videolan.org/Console/>
- [10] MPlayer - The Movie Player [online]. [cit. 2017-10-28]. Dostupné z: <http://www.mplayerhq.hu/DOCS/HTML/en/MPlayer.html>

- [11] Mpv.io [online]. [cit. 2017-10-28]. Dostupné z: <https://mpv.io/manual/master/>
- [12] ECMAScript 6 compatibility table [online]. [cit. 2017-10-28]. Dostupné z: <http://kangax.github.io/compat-table/es6/>
- [13] Browser market share [online]. [cit. 2017-10-28]. Dostupné z: <https://www.netmarketshare.com/browser-market-share.aspx?qprid=2&qpcustomd=0>
- [14] Eject [online]. [cit. 2017-10-28]. Dostupné z: <http://eject.sourceforge.net/>
- [15] Taem/cd-discid [online]. [cit. 2017-10-28]. Dostupné z: <https://github.com/taem/cd-discid>
- [16] Socat [online]. [cit. 2017-10-28]. Dostupné z: <http://www.abclinuxu.cz/software/system/site/socat>
- [17] LIRC - Linux Infrared Remote Control [online]. [cit. 2017-10-28]. Dostupné z: <http://www.lirc.org/>
- [18] Pyudev – pure Python libudev binding – pyudev 0.21.0 documentation [online]. [cit. 2017-10-28]. Dostupné z: <https://pyudev.readthedocs.io/en/latest/>
- [19] Overview – mutagen [online]. [cit. 2017-10-28]. Dostupné z: <https://mutagen.readthedocs.io/en/latest/>
- [20] Flask (A Python Microframework) [online]. [cit. 2017-10-28]. Dostupné z: <http://flask.pocoo.org/>
- [21] Socket.IO [online]. [cit. 2017-10-28]. Dostupné z: <https://socket.io/>
- [22] Flask-SocketIO documentation [online]. [cit. 2017-10-28]. Dostupné z: <https://flask-socketio.readthedocs.io/en/latest/>
- [23] React - A JavaScript library for building user interfaces [online]. [cit. 2017-10-28]. Dostupné z: <https://reactjs.org/>
- [24] React Native |A framework for building native apps using React [online]. [cit. 2017-10-28]. Dostupné z: <https://facebook.github.io/react-native/>
- [25] Font Awesome, the iconic font and CSS toolkit [online]. [cit. 2017-10-28]. Dostupné z: <http://fontawesome.io/>

- [26] Tanguyantoine/react-native-music-control [online]. [cit. 2017-10-28].  
Dostupné z:  
<https://github.com/tanguyantoine/react-native-music-control>
- [27] React-community/react-navigation [online]. [cit. 2017-10-28]. Dostupné z: <https://github.com/react-community/react-navigation>
- [28] Ivankrato/raspberry-pi-cdplayer [online]. [cit. 2017-10-28]. Dostupné z: <https://github.com/ivankrato/raspberry-pi-cdplayer>

# Příloha A

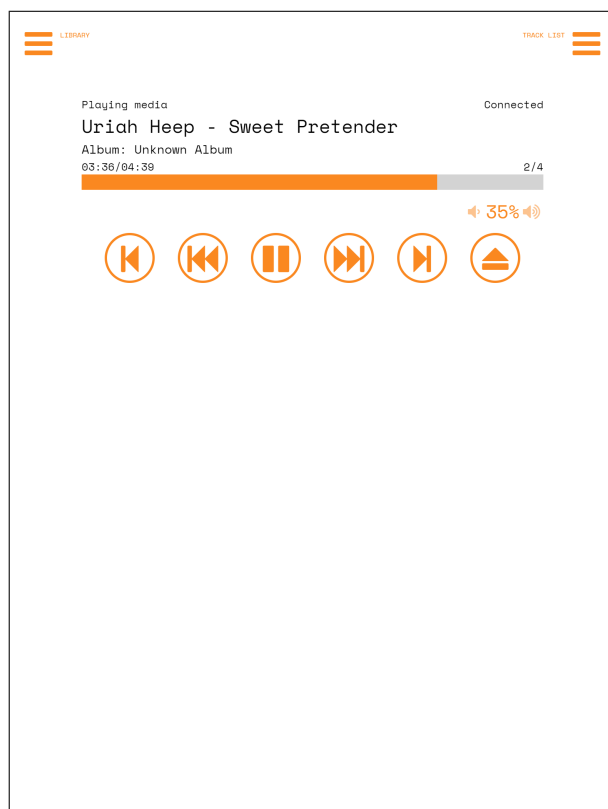
## Fotografie zařízení



Obrázek A.1: fotografie zařízení

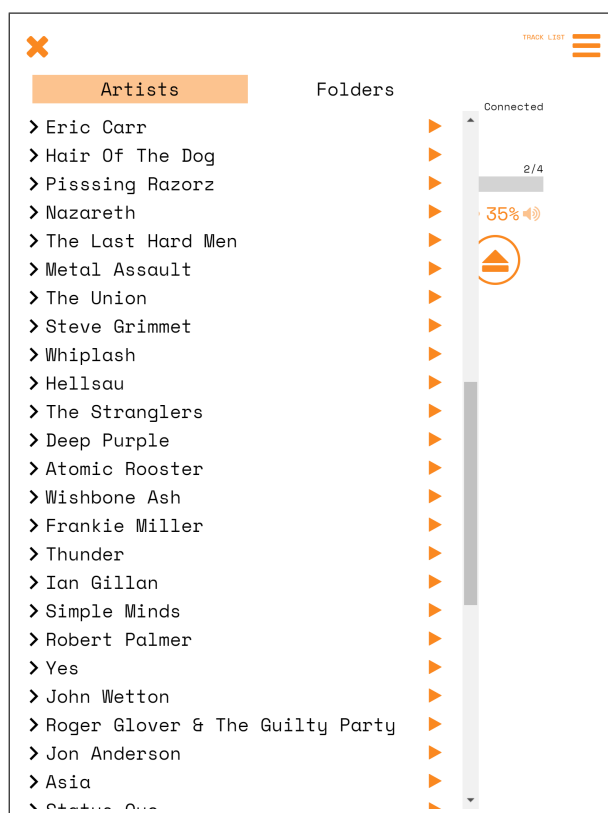
# Příloha B

## Screenshots z webového klienta z tabletu a počítače

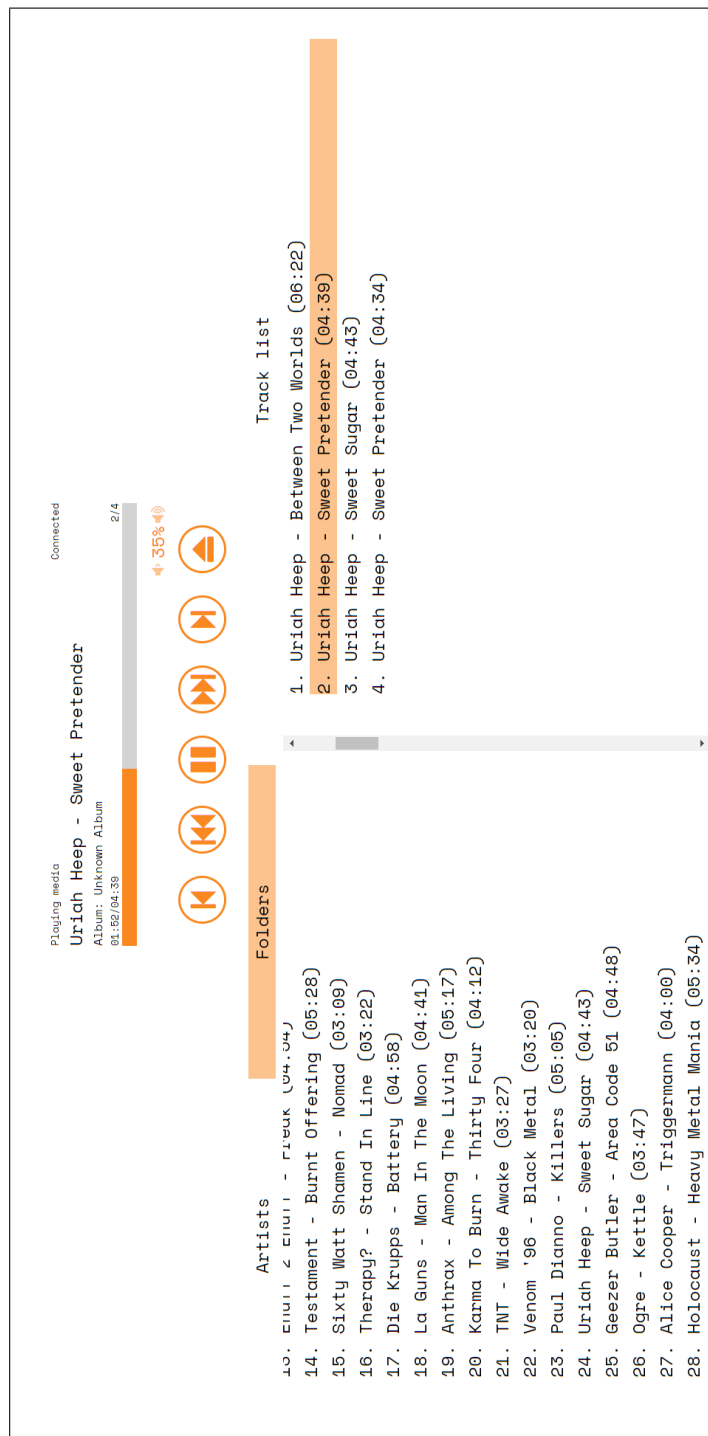


Obrázek B.1: webové rozhraní na tabletu





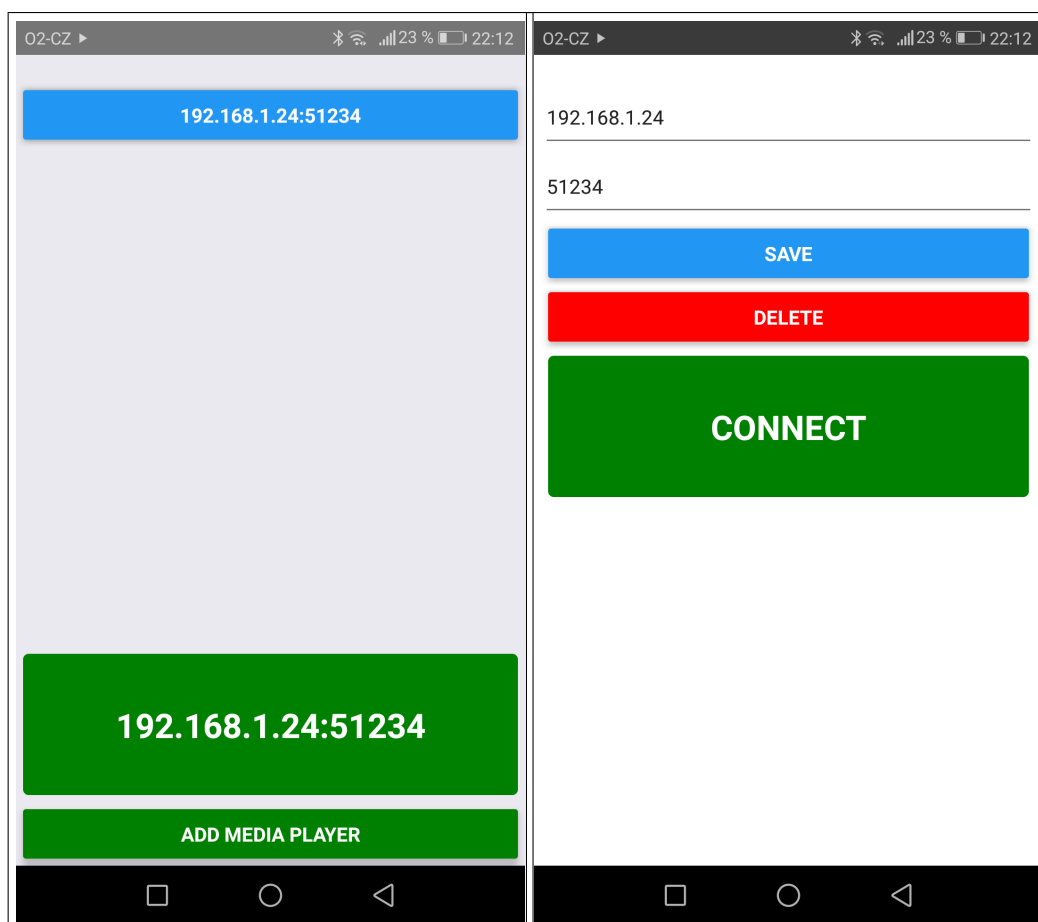
Obrázek B.2: webové rozhraní na tabletu s otevřenou knihovnou



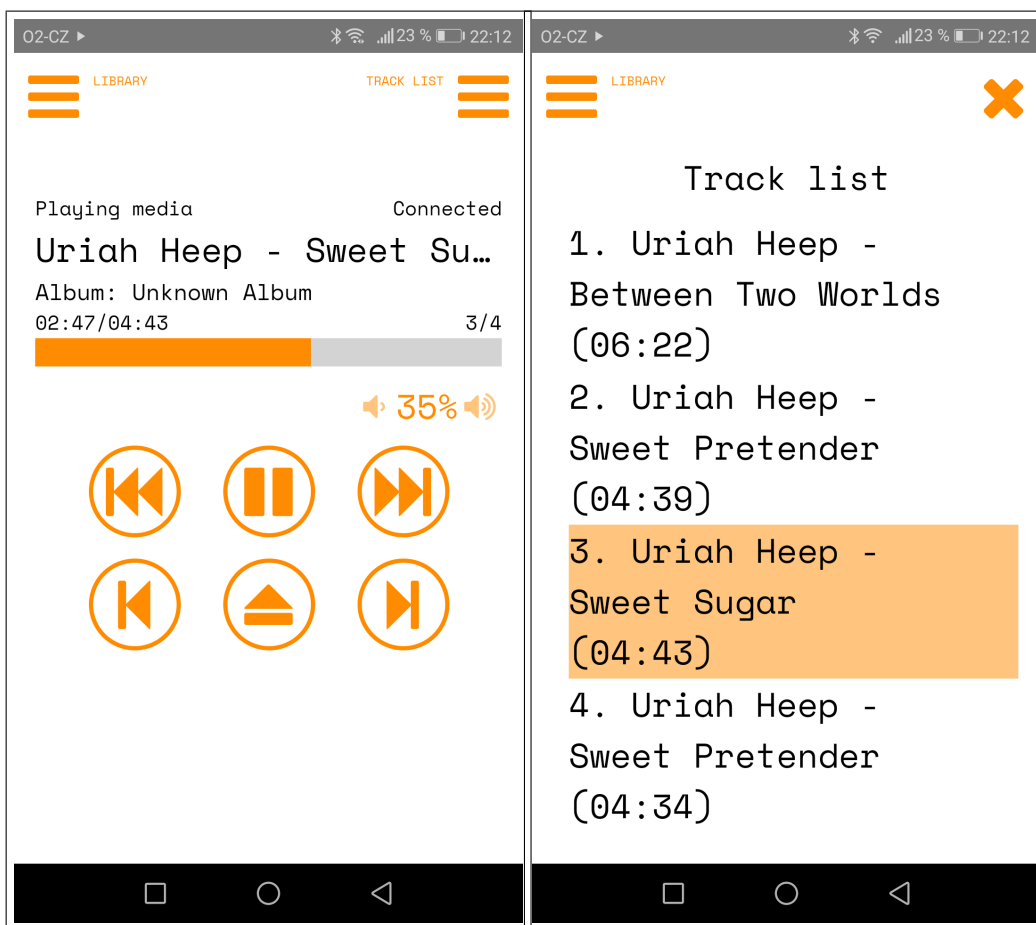
Obrázek B.3: webové rozhraní otevřené na stolním počítači

# Příloha C

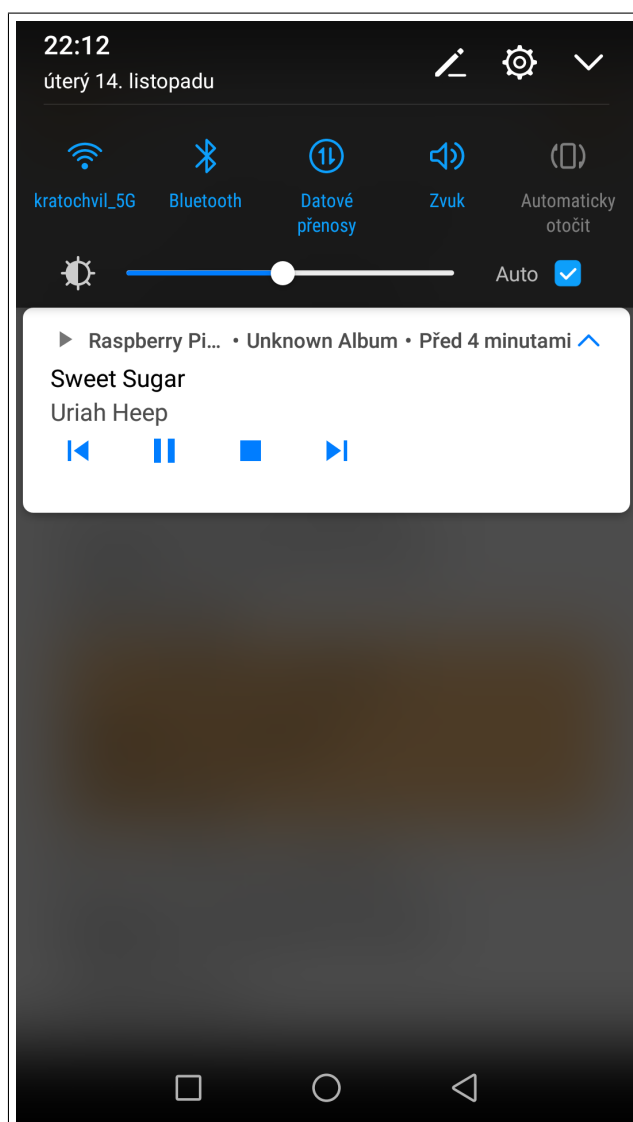
## Screenshots z mobilní aplikace



Obrázek C.1: úvodní obrazovka aplikace a úprava záznamu



Obrázek C.2: mobilní aplikace připojená k přehrávači – vpravo s otevřeným seznamem skladeb



Obrázek C.3: ukázka ovládání přehrávače z notifikace