

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Bakalářská práce

Mobilní virtuální turistický průvodce

Prohlášení

Prohlašuji, že jsem bakalářskou práci vypracoval(a) samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 2. 5. 2016

Milan Kuda

Poděkování

Chtěl bych poděkovat panu prof. Ing. Václavu Skalovi, CSc. za odborné vedení práce a cenné rady, které mi pomohly tuto práci zkompletovat.

Abstract

This bachelory work deals with problem of creation portable mobile virtual guide for touristic purposes. The main aim of this work is to construct system for guide, which allows us to build sights and from them composes a tour. The system counts with using multimedia files about the monuments. Another aim of the work is a data distribution for a tourists through their mobile phones or through other portable devices. The base of the system is a platform Raspberry Pi with linux operating system. An application uses web server Apache, database server MySQL and combination of programming language as PHP and JavaScript. Appearance and rendering of particular sites is solved by HTML and CSS.

Abstrakt

Tato práce se zabývá problematikou tvorby přenosného mobilního virtuálního průvodce pro turistické účely. Cílem práce je vytvořit systém pro průvodce, který umožňuje vytvářet památky a z nich následně skládat trasu. Systém přitom počítá s využitím multimediálních souborů o památkách. Dalším cílem je distribuce dat turistům skrze jejich mobilní telefony nebo přes ostatní přenosná zařízení. Základem systému je platforma Raspberry Pi s linuxovým operačním systémem. Aplikace využívá webového serveru Apache, databázového serveru MySQL a kombinaci programovacích jazyků PHP a JavaScript. Vzhled a výpis jednotlivých stránek je řešen pomocí HTML a CSS.

Obsah

1.	Úvod.....	6
2.	Výběr hardware a architektury.....	7
2.1.	Výběr počítače	7
2.2.	Výběr WiFi modulu	8
2.3.	Výběr baterie.....	8
2.4.	Výběr operačního systému.....	9
2.5.	Výběr paměti.....	9
2.6.	Výběr software.....	9
2.7.	Architektonický vzor.....	10
2.7.1.	Rozdělení do vrstev.....	10
2.7.2.	Třívrstvá architektura.....	10
2.8.	Architektura MVC	11
2.8.1.	Model	11
2.8.2.	View (pohled).....	11
2.8.3.	Controller	12
2.9.	Ostatní architektury.....	12
2.9.1.	MVP (Model – view – presenter).....	12
2.9.2.	Autonomous View	12
2.9.3.	Presentation Model (Model – view – viewmodel)	12
2.10.	Responzivní web design.....	13
3.	Instalace systému a tvorba aplikace	13
3.1.	Tvorba instalačního media	13
3.2.	Instalace Raspbianu.....	13
3.3.	Konfigurace Raspberry jako access point	14
3.3.1.	Konfigurace Wi-Fi	14
3.3.2.	Konfigurace síťové části	16
3.4.	Příprava webového serveru	18
3.4.1.	Instalace Apache	18
3.4.2.	Instalace PHP	18
3.4.3.	Instalace MySQL	18
3.4.4.	Instalace phpMyAdmin.....	18
3.4.5.	Aktualizace na PHP 7.....	19
3.5.	Aktualizace na Raspberry Pi 2 B	19
3.6.	Stream zvuku.....	19
3.6.1.	Instalace DarkIce.....	20
3.7.	Nástin webové aplikace.....	23

3.8.	Tvorba databáze	24
3.9.	Příprava adresářové struktury	25
3.10.	.htaccess	26
3.11.	Bootstrap	26
3.12.	Webová aplikace	27
3.12.1.	index.php.....	27
3.13.	Kontrolery	28
3.13.1.	Controller	28
3.13.2.	RouterController	29
3.13.3.	Třídy admin.....	29
3.13.4.	Třídy user	29
3.13.5.	Ostatní třídy.....	30
3.14.	Modely	31
3.14.1.	Database	31
3.14.2.	UserModel.....	31
3.14.3.	AreaModel.....	31
3.14.4.	MonumentModel.....	32
3.14.5.	GuideModel.....	32
3.15.	Pohledy.....	33
3.15.1.	Speciální pohledy	33
3.15.2.	Administrátorské pohledy	33
3.15.3.	Uživatelské pohledy	34
3.15.4.	Ostatní pohledy	35
3.16.	Java skripty.....	36
3.17.	Testování	38
3.17.1.	Měření signálu WiFi	38
3.17.2.	Měření přímé viditelnosti	38
3.17.3.	Měření v lesním porostu.....	39
3.17.4.	Měření ve městě	40
3.17.5.	Měření výdrže baterie	41
3.17.6.	Testování v reálném provozu	41
3.17.7.	Testování funkčnosti systémů	41
4.	Závěr	43
	Přehled zkratk.....	44
	Literatura.....	45
	Přílohy.....	47

1. Úvod

Tato bakalářská práce Mobilní virtuální turistický průvodce si klade za cíl realizovat systém pro tvorbu mobilních průvodcovských tras, který by uživateli umožnil jednoduchý přístup k informacím o památce včetně multimédií. Výsledný systém má za cíl implementaci přenosného webového serveru postaveného na platformě Raspberry Pi, ke kterému bude přistupováno pomocí webového prohlížeče na mobilním telefonu či tabletu.

V úvodu této práce bude popsán výběr jednotlivých částí hardwaru a porovnání obou použitých verzí Raspberry Pi. Čtenář se dozví informace o výběru WiFi modulu pro realizaci přístupového bodu nebo například o dostupných operačních systémech. Následně jsou v kapitole popsány vrstvené architektonické vzory. Také se seznámíme s architekturou Model-View-Controller a dalšími. Právě pomocí architektury MVC bude vytvořena aplikace.

V následující kapitole se budeme zabývat instalací operačního systému, konfigurací Apache a MySQL serveru a instalací překladače pro PHP. Také je zde popsána konfigurace přístupového bodu pomocí WiFi a přenos zvuku z mikrofonu k uživatelům. Po konfiguraci serveru je popsána tvorba databáze, nastavení htaccess a samotná tvorba webové aplikace. Na závěr se čtenář seznámí s testováním vytvořeného systému.

2. Výběr hardware a architektury

2.1. Výběr počítače

Ze zadání bakalářské práce vyplývá, že základ systému bude tvořit Raspberry Pi. Raspberry Pi je levný počítač s velikostí kreditní karty, který se dá připojit k monitoru nebo televizi a používá standardní myš a klávesnici [1]. Vývoj práce započal na verzi Raspberry Pi B+ a později byl vyměněn za novější verzi Raspberry Pi 2 model B.

Výhodou tohoto počítače je rozsáhlá základna uživatelů a různých návodů. Raspberry se dá použít pro spoustu zajímavých projektů, např.: stanice pro sledování počasí, cloudový server nebo ovládání robotů [2]. Další výhodou je velký výběr operačních systémů od upraveného Debianu až po Windows 10 IOT Core.

Jelikož vývoj probíhal na dvou generacích počítače, porovnáme si jednotlivé parametry a podíváme se na výkon. Pro oba počítače použijeme test sysbench, který je vhodný pro měření vícevláknového zatížení.

Zařízení	Raspberry Pi B+	Raspberry Pi 2 B
Chipset	Broadcom BCM2835	Broadcom BCM2836
Procesor	ARM1176JZF-S	ARM Cortex-A7
Počet jader	1	4
Velikost paměti	512 MB	1 GB LPDDR2

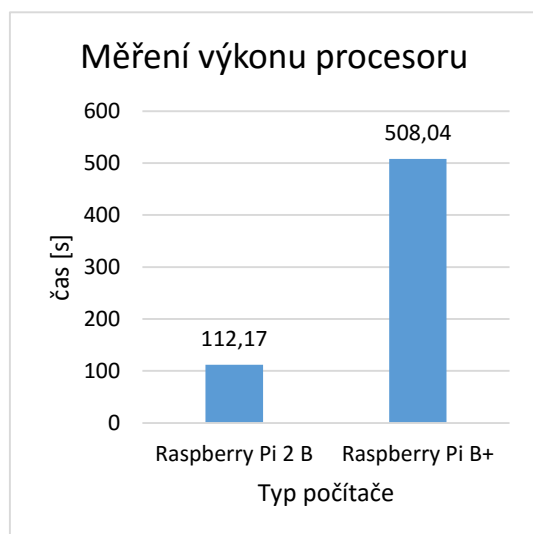
Tab. 2.1 - Popis specifikací

V tabulce tab. 2.1 můžeme vidět největší rozdíly mezi oběma verzemi. Hlavní rozdíl by měl spočívat ve vyšším výpočetním výkonu a velikosti paměti. Oba tyto aspekty jsou v této práci velmi důležité. Podíváme se tedy na slíbené porovnání výkonu. První test slouží k porovnání výkonu procesorů a druhý pro test paměti.

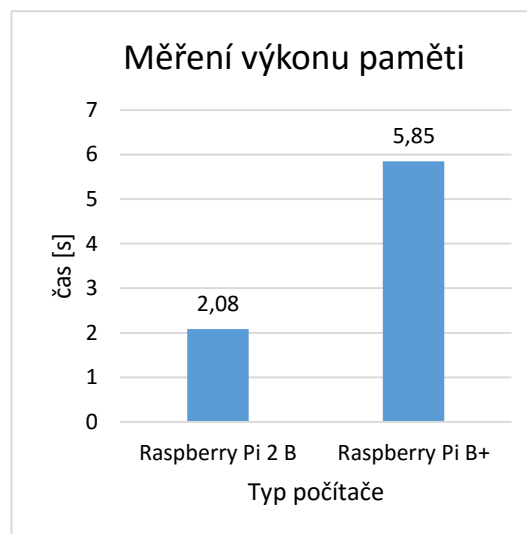
```
sysbench --test=cpu --num-threads=4 run
```

```
sysbench --test=memory --memory-block-size=1M --memory-total-size=10G --num-threads=4 run
```

Z výsledků uvedených v grafech obr. 2.1 a obr. 2.2 je patrné, že novější model představuje velký krok vpřed a mohl by vyřešit naše problémy s výkonem, které nastaly při testování minulý rok během WSCG 2015.



Obr. 2.1 - Test procesorů



Obr. 2.2 - Test paměti

2.2. Výběr WiFi modulu

Následující důležitou částí je výběr správného WiFi modulu. Na trhu se jich vyskytuje nepřehledné množství, ovšem pro tuto práci je potřeba specifický. Náš adaptér musí splňovat několik kritérií a to hlavně velikost a dostatečnou kapacitu pro co nejvyšší možný počet připojených uživatelů. Také se ukázalo, že vytvořit připojení pomocí klasického WiFi modulu není jednoduché. Klasické moduly nejsou koncipovány na vysílání jako access point a budeme potřebovat i upravený software, který nepodporují všechny výrobky na trhu. Díky široké Raspberry komunitě jsem našel stránku, kde se již podobný problém řeší [3]. Po nastudování jsem se rozhodl pro malý a levný modul EW-7811Un podporující IEEE 802.11n.

2.3. Výběr baterie

Požadavkem pro bakalářskou práci je i přenositelnost systému. Proto bude zapotřebí mít k Raspberry připojenou i externí baterii, která nám bude dodávat dostatek elektrické energie pro provoz. V poslední době se na trhu objevilo mnoho baterií pro nabíjení mobilních telefonů. Naštěstí mobilní telefony i Raspberry sdílejí stejný konektor pro napájení a to Universal Serial Bus (USB) typu micro-B. Výběr baterie byl tedy značně zjednodušený. Jako nejvýhodnější se jevila nabídka baterie XIAOMI

PowerBank 10400 mAh. Tato baterie má kapacitu 10400 mAh při 3,6 V, tj. 37,44 Wh. Baterie je schopná dodávat výstupní napětí 5,1 V a proud 2,1 A. Na stránkách Raspberry je uvedeno, že zdroj by měl být schopen dodat 1,8 A, což tato baterie splňuje. Nabíjení baterie lze provést nabíjecím adaptérem, například pro mobilní telefon s konektorem USB micro-B. Zde je limit pro napětí 5 V a pro proud 2 A. Nabíjení baterie z úplného vybití by mělo trvat kolem 5 a půl hodiny.

2.4. Výběr operačního systému

Výběr operačního systému je u Raspberry díky jeho rozšíření a komunitě poměrně široký. O tom svědčí i zapojení Microsoftu a jeho verze Windows 10 IOT (Internet of Things) Core. Základní linuxovou distribucí je Raspbian postavený na Debianu. Nalezneme ovšem i dvě distribuce, postavené na Ubuntu a to Ubuntu Mate s klasickým desktopem a Snappy Ubuntu Core, upravenou pro vývojáře na Raspberry. Velmi oblíbená je aplikace Raspberry jako domácí multimediální centrum. Existují tedy hned dvě distribuce a to Open Source Media Centre (OSMC) a Open Embedded Linux Entertainment Centre (OPENELEC). Posledním systémem je RISC OS, tento systém je vyvíjen speciálně pro procesory typu ARM. Na všechny tyto operační systémy můžeme získat odkaz na oficiálních stránkách Raspberry. Já jsem zvolil Raspbian, protože se jedná o nejrozšířenější operační systém a budu moci využít své znalosti Linuxu. Rozhodl jsem se použít starší verzi s kódovým jménem Wheezy, jelikož novější verze Jessie není zcela kompatibilní, viz kapitola 3.4.5..

2.5. Výběr paměti

Pro operační systém a data budeme potřebovat dostatek rychlé paměti. U Raspberry se vše ukládá na micro Secure Digital paměťovou kartu. Existuje seznam funkčních karet na stránkách eLinux, ovšem tento seznam je testován uživateli a nelze ho vydávat za směrodatný. Na druhé straně je dobré pořídit spíše kartu, která byla uživateli označena jako funkční. Maximální rychlost je omezena na 23,84 MB/s (řadič SD karty pracuje na 50 MHz po 4 bitech), nemá tedy smysl kupovat o moc rychlejší kartu. Já jsem zvolil kartu Maxwell Micro SDHC 32GB class 10.

2.6. Výběr software

Operační systém máme vybrán a musíme tedy zvolit způsob, jak předat informace o památkách uživatelům. Problém tkví v roztržitosti platform. Bylo by nesmyslné tvořit zvláštní aplikace pro Android, iOS a Windows Mobile. Vycházel jsem

z toho, že každé přenosné zařízení má vlastní webový prohlížeč. Proto bude nejjednodušší vytvořit responzivní webové aplikace. Při tvorbě aplikace využiji získaných poznatků z předmětu KIV/WEB. Rozhodl jsem se web postavit na osvědčené platformě LAMP tj. Linux, Apache, MySQL a PHP. Další výhodou bude, že při vývoji HTML 5 byly přidány tagy pro audio a video. Nebude tedy problém splnit multimediální podmínku práce. Dalším úkolem je potřeba zvolit vhodný architektonický vzor.

2.7. Architektonický vzor

„Architektura je subjektivní pohled na systém, na němž se shodli experti pracující na projektu. Popisuje podobu a komunikaci hlavních komponent. Ovlivňuje řadu detailů, takže je třeba ji mít navrženou co nejdříve, abychom nemuseli své programy později zásadně měnit“ [4].

2.7.1. Rozdělení do vrstev

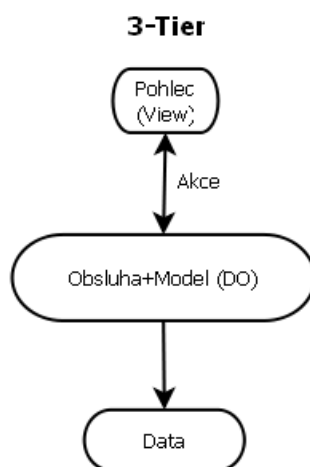
Navržený systém nejčastěji rozkládáme do vrstev. Pro každou vrstvu platí, že je definována pomocí nižší vrstvy a poskytuje služby a rozhraní pro vrstvu vyšší. Vrstvy jsou většinou black boxy, vyšší vrstvy tedy nevidí, jak jsou aplikovány použité funkce v nižší vrstvě. Architektury jsou buď vrstveny silně (vyšší vrstva využívá pouze vrstvu o jednu nižší) nebo slabě (zde můžeme využít i služby poskytované nižšími vrstvami). Nejznámější použitou vícevrstvou architekturou je sedmivrstvá ISO/OSI architektura používaná v počítačových sítích.

Proč používat rozdělení do vrstev? Vrstvy můžeme v silně vrstvených architekturách snadno nahrazovat. Jde to i v slabě vrstvených architekturách, ovšem je to pracnější. Další výhodou je snadnější pochopení. Pokud máme například funkci pro vymazání složky, tak nás ve vyšší vrstvě nemusí zajímat, zda je složka prázdná a půjde vymazat. Stačí nám pouze zadat jméno složky. Mezi nevýhody patří například snížení výkonu (každá vrstva převádí data z jedné reprezentace do druhé) a nutnost občas provádět kaskádové změny [4].

2.7.2. Třívrstvá architektura

Jak již nadpis napovídá, tato architektura se skládá ze tří vrstev a to prezentační, datové a aplikační, viz obr. 2.3. Jednotlivé vrstvy jsou navzájem nezávislé. Toho je možno využít pro vývoj, kdy se pro změnu nemusíme ohlížet na ostatní vrstvy, pokud dodržíme podmínku neměnného rozhraní mezi vrstvami [5].

Prezentační vrstva slouží ke komunikaci s uživatelem, tj. pomocí grafického rozhraní nebo webového klienta. Datová vrstva se stará o propojení s ostatními systémy, většinou s úložištěm dat. V aplikační vrstvě jsou data zpracována [4].



Obr. 2.3 - Třívrstvá architektura [6]

2.8. Architektura MVC

Předchozí kapitoly sloužily jako přiblížení problematiky, abychom mohli pochopit architekturu MVC, která bude použita v této práci. Architektura MVC je také rozdělena na tři části a to model, view a controller. V MVC jde v zásadě o oddělení logiky od výstupu, nechceme-li mít vedle sebe HTML značky pro vykreslování a logiku z PHP. Takový kód je špatně čitelný a určitě nebude snadné ho dále rozšiřovat. To ovšem bývá složité dodržet, proto i v pohledech (view) můžeme nalézt části PHP kódu, například pro výpis více řádků tabulky. Problémem MVC je, že není naprosto pevně definována a je snadno zaměnitelná s MVP [7, 8].

2.8.1. Model

V modelu je popsána samotná logika, tj. výpočty, komunikace s databází a kontrola formulářů. Jsou zde uloženy funkce, které voláme z kontrolerů. Důležité je, že model neví, jak bude výstup vypadat a odkud získal vstupní parametry [8].

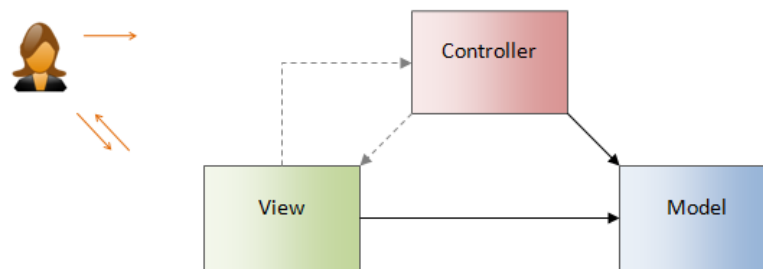
2.8.2. View (pohled)

View slouží k vykreslení výstupu pro uživatele. Snažíme se co nejvíce omezit jiný jazyk než vykreslovací (v našem případě HTML). View by měl obsahovat co nejmenší možné množství logiky a rovněž, stejně jako model, se nezajímat o původ dat [8].

2.8.3. Controller

Controller je poslední prvek tohoto vzoru. Slouží jako spojka, která komunikuje s uživatelem, pohledem a modelem. Od uživatele získává vstupní data, která následně zpracovává pomocí funkcí z modelu. Následně výsledek zobrazí pomocí pohledu [8].

Uživatel by měl podle tohoto rozložení komunikovat pouze s controllerem, ale to nemusí platit vždy. Pokud si například ošetříme pomocí Javascriptu vstup do formuláře, komunikuje uživatel přímo s pohledem. Proto výsledný diagram bude vypadat takto [7].



Obr. 2.4 - MVC architektura [7]

2.9. Ostatní architektury

2.9.1. MVP (Model – view – presenter)

V této architektuře se předpokládá přítomnost widgetových systémů (Flex, JavaFX). Hlavním rozdílem oproti architektuře MVC je, že uživatel komunikuje pouze s pohledem a pohled má přímou vazbu na presenter. Pohled po akci uživatele zavolá metodu presenteru, který obsahuje aplikační a prezentační logiku. Model zůstává nezměněn [7].

2.9.2. Autonomous View

Tato architektura kombinuje v jednom souboru uživatelské rozhraní a logiku. Vzniká tak často nepřehledný kód, který se používá pro jednodušší aplikace, kde by bylo použití složitějšího vzoru spíše na obtíž. Pokud bychom se rozhodli napsat složitější aplikaci s využitím Autonomous View, byl by kód jistě neúměrně dlouhý a také testování velmi složité [7].

2.9.3. Presentation Model (Model – view – viewmodel)

Presentation Model je odlišný model od MVC a MVP. Rozdíl je v tom, že view nepřístupuje k modelu přímo. Jako prostředník ke komunikaci slouží vrstva presentation

model (prezentační model), což je vlastně kombinace modelu a presenteru. Tato třída udržuje stav aplikace. View se dotazuje viewmodelu a podle toho vykresluje ovládací prvky. Na druhou stranu proudí informace od uživatele. Model opět poskytuje aplikaci data a o vrstvě view neví nic [9].

2.10. Responzivní web design

Tvorba webové aplikace má jednu nevýhodu. Musí vypadat dobře na všech zařízeních s různou úhlopříčkou obrazovky, rozlišením nebo nastavením velikosti textu. Vzhled webu je určen kaskádovými styly CSS a JavaScriptem. Naštěstí nemusíme vytvářet vzhled aplikace a testovat web na různých zařízeních. Existuje poměrně velké množství frameworků, které nám umožňují vytvořit responzivní vzhled stránky. Asi nejznámější front-end frameworkem pro webové aplikace je Bootstrap. Má výhodu v rozšířené komunitě a různých doplňcích. Tento framework bude využit v mé práci.

3. Instalace systému a tvorba aplikace

3.1. Tvorba instalačního media

Pro tvorbu instalačního media potřebujeme microSD kartu, čtečku karet, program Win32DiskImager a obraz instalačního média, který získáme na stránkách Raspberry. Pro instalaci si microSD kartu připravíme tak, že ji naformátujeme a jako souborový systém zvolíme FAT32. Tento souborový systém volíme proto, abychom se vyhnuli problémům při nahrávání obrazu disku. Obraz disku po stažení otevřeme ve výše zmíněném programu a obraz nahrajeme na připravenou SD kartu.

3.2. Instalace Raspbianu

V instalaci pokračujeme tak, že připojíme Raspberry pomocí rozhraní HDMI k monitoru a také připojíme USB klávesnici pro ovládání uživatelského rozhraní. SD kartu vložíme do Raspberry a zapneme. Po chvilce čekání se objeví instalační obrazovka s 9 možnostmi.

1. Expand Filesystem – tuto možnost zvolíme, jelikož jsme si obraz disku nahráli sami. Jinak bychom se připravili o místo, které na SD kartě zbývá. Pokud se rozhodneme časem vyměnit paměťovou kartu za větší, lze pomocí této funkce rozšířit souborový systém i na zbytek nové paměti

2. Change User Password – zde nastavíme heslo uživatele, základní heslo pro uživatele pi je Raspberry. V našem případě budu všude používat heslo „leleazta“
3. Enable Boot To Desktop/Scratch – nastavení zavaděče, vybereme možnost Console Text (při startu se objeví terminál), v této aplikaci není důvod mít zapnuté grafické rozhraní
4. Internationalisation Options – regionální nastavení pro čas, kódování a rozložení klávesnice
5. Enable Camera - zapnutí kamery, pro naše účely jej vynecháme
6. Add to Rastrack – mapa funkčních Raspberry po světě, opět vynecháme
7. Overclock – přetaktování CPU a GPU, pro úsporu energie vynecháme
8. Advanced Options – ostatní nastavení, zde vybereme SSH a zapneme
9. About raspi-config – informace o této konfigurační utilitě.

Na závěr zvolíme finish a Raspberry se restartuje. Následně připojíme Raspberry do sítě s DHCP serverem. Pokud se vše podařilo, můžeme nyní ovládat Raspberry přes protokol SSH (uživatelské jméno pi a heslo leleazta). V následujících krocích budeme potřebovat připojení k Internetu pro stahování balíčků programů. Pokud bychom chtěli kdykoliv vyvolat konfigurační obrazovku, stačí zadat příkaz `sudo raspi-config`.

3.3. Konfigurace Raspberry jako access point

V následujících kapitolách je popsána konfigurace přístupového bodu pro Raspberry. Nalezneme zde konfiguraci WiFi a nastavení sítě. Testování vytvořené sítě můžeme nalézt v kapitole 3.17.

3.3.1. Konfigurace Wi-Fi

Jak již bylo zmíněno v teoretické části, zvolil jsem WiFi modul Edimax EW7811-UN s čipem Realtek RTL8188CUS. Návod pro tvorbu hotspotu bude vycházet ze zdrojů [3, 10].

V prvním kroku provedeme aktualizaci softwaru Raspberry.

```
sudo apt-get update && sudo apt-get upgrade
```

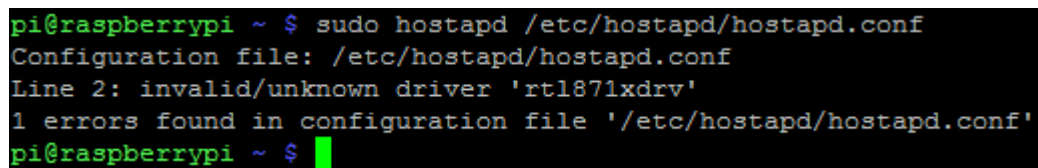
Následně nainstalujeme potřebné aplikace `hostapd`, `hostapd-utils` a `iw`.

```
sudo apt-get install hostapd hostapd-utils iw
```

Nyní si zapíšeme konfiguraci WiFi sítě. Nejprve vytvoříme soubor `hostapd.conf`, kde bude konfigurace uložena a následně vložíme připravenou konfiguraci zapsanou v příloze 1. V konfiguraci můžeme vidět nastavení SSID, kanál WiFi, typ zabezpečení a heslo.

```
sudo nano /etc/hostapd/hostapd.conf
```

Pokud bychom se nyní pokusili spustit hotspot, program nám vypíše tuto chybovou hlášku jako na obr. 3.1.



```
pi@raspberrypi ~ $ sudo hostapd /etc/hostapd/hostapd.conf
Configuration file: /etc/hostapd/hostapd.conf
Line 2: invalid/unknown driver 'rtl871xdrv'
1 errors found in configuration file '/etc/hostapd/hostapd.conf'
pi@raspberrypi ~ $
```

Obr. 3.1 - Chyba při spuštění `hostapd`

Budeme tedy muset nainstalovat speciálně upravenou verzi `hostapd`, která nám umožní vytvoření access pointu i s tímto adaptérem. Nejprve tedy odinstalujeme `hostapd` a poté stáhneme upravenou verzi a tu zkompilujeme. Mohlo by se zdát, že první instalace programu `hostapd` byla zbytečná, ovšem není tomu tak. Pro správnou funkci tohoto programu se musí nainstalovat i několik balíčků, na jejichž funkci tento program závisí. Zde je vidět stažení balíčku a překlad programu.

```
sudo apt-get autoremove hostapd

wget https://github.com/jenssegers/RTL8188-hostapd/archive/v2.0.tar.gz

tar -zxvf v2.0.tar.gz

cd RTL8188-hostapd-2.0/hostapd

sudo make
```

Po kompilaci budeme schopni program nainstalovat a spustit. Pokud jsme vše provedli správně, měla by se nám zobrazit nová bezdrátová síť na přijímači. Samotná instalace proběhne pomocí `make` níže.

```
sudo make install

sudo service hostapd restart
```

Konfigurace vysílače WiFi je nyní kompletní, ale nebude možné se k síti jednoduše připojit. Pro připojení klientů je vhodné nastavit DHCP server a jméno hostitele, na které bude server reagovat. Na oba tyto problémy se zaměříme v další části.

3.3.2. Konfigurace síťové části

WiFi síť již nyní vysílá, musíme si tedy nastavit síťové rozhraní. K tomu použijeme aplikaci dnsmasq, která nám umožní nastavení DHCP serveru i hostname. Pro uživatele bude značně pohodlnější zadávat jako URL text místo IP adresy. Nejprve však nastavíme síťové rozhraní WiFi. Musíme mu přiřadit IP adresu a masku. Otevřeme si tedy soubor pro nastavení síťových rozhraní.

```
sudo nano /etc/network/interfaces
```

Nalezneme odstavec s wlan0. Pro konfiguraci síťového rozhraní použijeme následující nastavení.

```
iface wlan0 inet static
    address 192.168.254.1
    netmask 255.255.255.0
wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf
iface default inet dhcp
```

Nastavíme tedy statickou IP adresu serveru a masku typu C. Toto nastavení je nutné. V původním nastavení sloužila síťová karta pouze jako přijímač a tudíž ve většině případů obdržela parametry pro připojení do sítě od DHCP serveru. Také se podíváme na nastavení rozhraní eth0. Po konfiguraci dnsmasq by se mohlo stát, že síťové rozhraní eth0 nebude správně nakonfigurované pro přijímání dat z DHCP serveru. Pokud bychom ho s původními parametry připojili do sítě, nepřijalo by nastavení a server by se stal pro nás nedostupným. To by bylo nepříjemné v tom, že bychom nemohli instalovat aktualizace ani další programy. Řádek s nastavením pro eth0 nalezneme níže.

```
iface eth0 inet dhcp
```

Další změnu provedeme v souboru dnsmasq.conf. Zde nastavíme název hostitele a DHCP server. Do konfiguračního souboru se dostaneme takto:


```
sudo nano /etc/dnsmasq.conf
```

Řádek s `dhcp-option` značí přiřazení adresy. Rezervujeme si IP adresy v rozsahu 100 až 200 s dobou platnosti adresy na 12 hodin. Možnost `dhcp-option-3` slouží jako výchozí brána a možnost `dhcp-option-6` jako DNS server. Obě tyto možnosti nastavíme na základní adresu serveru. Doménové jméno zvolíme `server`, následně jej změňíme až v souboru `hostname`. Do souboru vložíme následující řádky.

```
interface=wlan0  
  
domain=server  
  
dhcp-range=192.168.254.100,192.168.254.200,255.255.255.0,12h  
  
dhcp-option=3, 192.168.254.1  
  
dhcp-option=6, 192.168.254.1
```

Poslední dvě úpravy se budou týkat `hostname`. Jak již bylo řečeno, pro uživatele je příjemnější zadávat text místo IP adresy. Nejprve tedy změňíme `hostname` našeho serveru, do konfigurace se dostaneme tímto příkazem.

```
sudo nano /etc/hostname
```

Zde pouze přepíšeme původní název. Já jsem vybral jméno `guide` jako průvodce. V této části nám tedy zbývá pouze nastavit směrování v `hosts`.

```
sudo nano /etc/hosts
```

Zde vložíme dva řádky. První je vnitřní směrování. Na druhém řádku nalezneme dva názvy pro stejnou IP.

```
127.0.1.1    server  
192.168.254.1  guide.raspi guide
```

Pokud v prohlížeči zadáme pouze adresu `guide`, mají některé prohlížeče (např. Chrome) tendenci vyhledávat tento název ve zvoleném vyhledávači a stránku nezobrazí. Proto jsou zde dva názvy. Pokud zadáme `guide.raspi`, prohlížeč se zeptá DNS serveru na Raspberry, zda nezná takovou adresu. Pokud by stále prohlížeč nechtěl stránku zobrazit, přidáme za adresu `/`. Tím se zamezí vyhledávání názvu v internetovém vyhledávači.

3.4. Příprava webového serveru

Pro naši aplikaci budeme potřebovat Apache webový server, MySQL databázi, PHP a phpMyAdmin správce databáze. Instalace je inspirována zdrojem [11].

3.4.1. Instalace Apache

Začneme instalací webového serveru. Jako první musíme nainstalovat balíčky pro Apache. Tato instalace je velice jednoduchá a bude nám na ni stačit jeden příkaz.

```
sudo apt-get install apache2 apache2-doc apache2-utils
```

Protože aplikace bude uložena v základním adresáři webových stránek, nebudeme potřebovat žádnou konfiguraci Apache. Vše by mělo být připraveno k používání. Následnou konfiguraci pro velikost nahrávaného souboru pomocí POST vložíme do souboru `.htaccess`.

3.4.2. Instalace PHP

Druhou částí je instalace PHP, opět se bude jednat o jeden řádek a nebudeme muset provádět žádnou konfiguraci.

```
sudo apt-get install libapache2-mod-php5 php5 php-pear php5-xcache
```

Následně nainstalujeme balíček pro konektivitu mezi PHP a MySQL.

```
sudo apt-get install php5-mysql
```

3.4.3. Instalace MySQL

Předposledním krokem bude instalace databázového serveru MySQL. Zde budeme při instalaci vyzváni k zadání správcovského hesla databáze.

```
sudo apt-get install mysql-server mysql-client
```

3.4.4. Instalace phpMyAdmin

Poslední instalací bude webová aplikace phpMyAdmin pro správu databáze. Jako první budeme potřebovat nainstalovat balíček aplikace.

```
sudo apt-get install phpmyadmin
```

Během instalace musíme zvolit webový server, pro který se má nastavit běh phpMyAdmin. Zvolíme Apache2. Další obrazovka se při instalaci zeptá, zda chceme

pro balík nastavit a nainstalovat databázi. Zvolíme ano a budeme vyzváni k zadání MySQL hesla a vytvoření nového hesla pro phpMyAdmin.

Pokud je vše nainstalováno správně, měli bychom pod adresou `guide.raspi/` v prohlížeči nalézt informační stránku s nápisem „It works!“. K phpMyAdmin se připojíme přes `guide.raspi/phpmyadmin` s přihlašovacím jménem `root` a heslem, jenž jsme zadali při instalaci (v tomto případě opět `leleazta`).

3.4.5. Aktualizace na PHP 7

Pro server jsem chtěl zvolit PHP nejnovější verze 7. Ovšem tato verze funguje pouze pod novější distribucí Raspbianu Jessie. Po provedené aktualizaci se povedlo novější verzi nainstalovat. Poté se však vyskytl problém s aplikací `hostapd`, která při aktualizaci naprosto zmizela. Reinstalace této aplikace nepomohla a k Raspberry se nebylo možno připojit pomocí WiFi. Tuto možnost jsem tedy zamítl a vrátil jsem se k Raspbianu Wheezy. Tento problém znamená i nemožnost použití nejnovějšího Raspberry Pi 3, které funguje správně pouze pod nejnovějším firmwarem, který není obsažen v této starší verzi operačního systému.

3.5. Aktualizace na Raspberry Pi 2 B

Protože se původní Raspberry při testování příliš neosvědčilo, rozhodl jsem se pro výměnu za novější a silnější model. Tento model ovšem není zpětně kompatibilní se starším Raspberry. Je nutné tedy provést aktualizaci na starším Raspberry a poté vložit paměťovou kartu do novějšího. Aktualizace je jednoduchá, jako například když aktualizujeme Ubuntu na novější verzi distribuce. Stačí zadat tyto tři příkazy.

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get dist-upgrade
```

Tato aktualizace ovšem nahrála z repositářů i novější verzi `hostapd` a přestal tedy fungovat hotspot. To lze snadno opravit reinstalací `hostapd` stejným způsobem, jak je uvedeno v kapitole 3.3.1.

3.6. Stream zvuku

Pro lepší komunikaci s průvodcem by měl být server schopen přenášet zvuk z mikrofону přes WiFi připojení do mobilních zařízení. Po hledání jsem objevil aplikaci

Iccast2, která umí přenášet stream audio souborů. Našel jsem i návod, jak tohoto programu využít pro tuto potřebu [12].

3.6.1. Instalace DarkIce

DarkIce bude sloužit k zaznamenávání zvuku z mikrofonu a jeho přeposílání programu Icecast, který provede samotný stream. Formát MP3 není otevřený a není tedy nabízen v základní instalaci Raspberry. Musíme tedy přidat repozitář, který tyto balíčky obsahuje. Následně tyto balíčky nainstalujeme.

```
sudo sh -c "echo 'deb-src http://mirrordirector.raspbian.org/raspbian/ wheezy main contrib non-free rpi' >> /etc/apt/sources.list"

sudo apt-get update

sudo apt-get --no-install-recommends install build-essential devscripts autotools-dev fakeroot dpkg-dev debhelper autotools-dev dh-make quilt ccache libsamplerate0-dev libpulse-dev libaudio-dev lame libjack-jackd2-dev libasound2-dev libtwolame-dev libfaad-dev libflac-dev libmp4v2-dev libshout3-dev libmp3lame-dev
```

Po nainstalování potřebných balíčků vytvoříme složku src a přesuneme se do ní. Do této složky stáhneme zdrojové soubory programu DarkIce a budeme v nich muset udělat pár změn pro podporu formátu MP3. Také musíme nastavit práva ke složce a ostatním souborům, abychom mohli změny provést.

```
mkdir src && cd src/

sudo apt-get source darkice

sudo chmod -R 777 darkice-1.0/
```

Jako první upravíme soubor rules, abychom mohli zkompilevat program v našem prostředí. Soubor stáhneme pomocí wget a pak nahradíme starý. Do souboru lze úpravy zapsat i ručně, ale je zde nebezpečí, že se program nezkompiluje, například kvůli špatně umístěné mezeře.

```
wget https://stmlr.net/files/linux/rules

mv rules darkice-1.0/debian/
```

Následně změníme verzi balíku tak, aby reflektovala podporu MP3 a program nainstalujeme.

```
cd darkice-1.0
debchange -v 1.0-999~mp3+1
dpkg-buildpackage -rfakeroot -uc -b
sudo dpkg -i ../darkice_1.0-999~mp3+1_armhf.deb
```

Nyní přeskochíme konfiguraci DarkIce a vrátíme se k ní později. Pustíme se tedy do instalace Icecast2. Pro instalaci musíme použít nástroj aptitude. Samotný balíčkovací systém APT není z neznámého důvodu schopen stáhnout požadovaný balíček aplikace. Při instalaci zvolíme ruční konfiguraci, server nastavíme jako localhost a hesla dle uvážení.

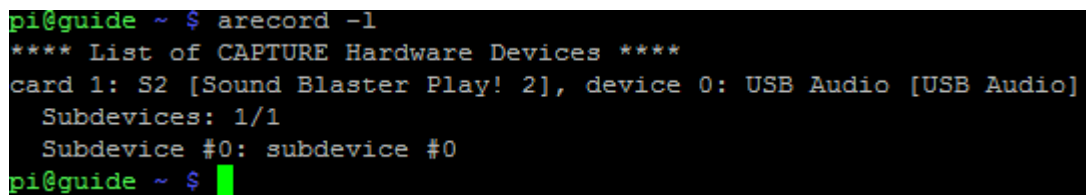
```
sudo aptitude install icecast2
```

Icecast nyní nalezneme po zadání adresy guide.raspi:8000 do webového prohlížeče.

Po instalaci je potřeba DarkIce nakonfigurovat. Nejprve si vypíšeme údaje o naší zvukové kartě.

```
arecord -l
```

Zde uvidíme, kde se zvuková karta nachází, tj. card 1 a device 0 dá adresu hw:1,0 pro konfigurace jak je vidět na obr. 3.2.



```
pi@guide ~ $ arecord -l
**** List of CAPTURE Hardware Devices ****
card 1: S2 [Sound Blaster Play! 2], device 0: USB Audio [USB Audio]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
pi@guide ~ $
```

Obr. 3.2 - Arecord, výpis zvukových karet

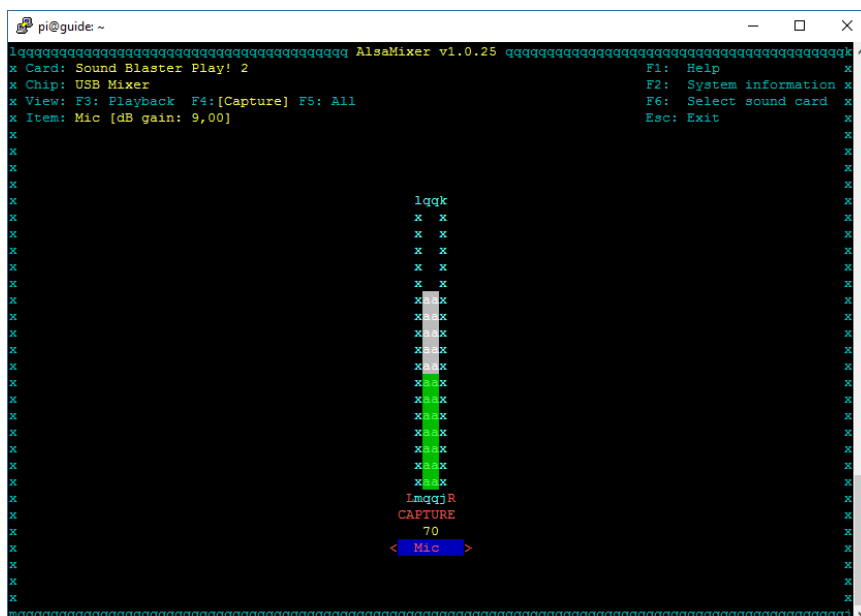
Vytvoříme tedy konfigurační soubor v /etc. Použitá konfigurace je v příloze 3.

```
sudo nano /etc/darkice.cfg
```

Při nastavování si musíme dát pozor, zda naše zvuková karta nahrává stereo nebo mono zvuk. Při nesprávném nastavení se DarkIce nespustí.

Poslední část se týká nastavení mikrofону. Přes program alsamixer můžeme nastavit zvýšení hlasitosti, viz obr. 3.3. Osvědčilo se mi nastavení kolem 70 %. Při vyšším nastavení docházelo k deformaci hlasu. Následně uložíme nastavení.

```
sudo alsactl store
```



Obr. 3.3 - AlsaMixer

Vše je nyní nastaveno a můžeme spustit vysílání. Použijeme příkaz nohup, který zajistí běh programu i při vypnutí terminálu. Výpis z aplikace bude přeměrován do souboru nohup.out.

```
sudo nohup darkince &
```

Takto vytvořený stream má jednu nevýhodu. Odezva od hlasu z mikrofonu k příjemci je 10 až 15 sekund. Pokud bychom se pokusili odezvu snížit, například v konfiguraci DarkIce, tento program by byl během svého běhu ukončen. Nejprve jsem chtěl v aplikaci vytvořit stránku pro administrátora, kde by bylo možné stream zapnout a vypnout. Bohužel se tato možnost ukázala jako nevhodná, protože DarkIce se z neznámých důvodů občas nechtěl zapnout, případně způsobil pád síťového rozhraní. Proto jsem se rozhodl pro jiný přístup a to takový, že se DarkIce spustí při zapnutí serveru, pokud je připojen mikrofon. Postup, jak upravit program pro spuštění, jsem našel na fóru pro podporu Ubuntu ubuntuforums.org. Nejprve jsem v souboru `/etc/init.d/darkice` upravil řádky tak, aby odpovídaly níže uvedeným (nejsou uvedeny celé odstavce, ale pouze části do změny).

```

start-stop-daemon --start --quiet -m --pidfile $PIDFILE \
stop_server() {
# Stop the process using the wrapper
    start-stop-daemon --stop --quiet --pidfile $PIDFILE \
        --exec $DAEMON
    errcode=$?
    rm $PIDFILE
running() {
    # Check if the process is running looking at /proc
    # (works for all users)
    sleep 1

```

Následně upravíme soubor `/etc/default/darkice` tak, že přepíšeme řádku `RUN=NO` na `RUN=yes`. Pro spuštění bez přihlášení je potřeba přidat nového uživatele příkazem `adduser nobody audio`. Jako poslední část opravíme čas spuštění. `DarkIce` se zřejmě po zapnutí serveru spouští brzy a má problémy s načtením zvukové karty.

```

update-rc.d -f darkice remove
update-rc.d darkice defaults 99

```

Tímto končí konfigurace serveru a přistoupíme k tvorbě webové aplikace.

3.7. Nástin webové aplikace

Aplikace by měla být pro uživatele i průvodce co nejjednodušší na používání. Uživatel tedy bude mít možnost zvolit trasu a procházet seznamem památek tak, jak průvodce danou trasu vytvoří. U každé památky by se měl dozvědět základní informace, GPS souřadnice a pozici na mapě. Následně, pokud průvodce tyto údaje přidal, si může uživatel zobrazit fotografie, přehrát přiložené audio nebo pustit video. Při pohybu mezi památkami bude uživatel informován, k jaké památce se přesouvá a jak dlouho bude přesun trvat.

Průvodce může po přihlášení vytvářet, měnit nebo mazat památky, trasy a lokace. Důležitá je tvorba lokace, bez ní nepůjde vytvořit památka ani trasa. Lokace slouží jako oblast, ve které se bude průvodce s návštěvníky pohybovat. Může se jednat

o město, krajinu nebo třeba budovu. V rámci lokace se vloží mapa, na kterou bude průvodce vkládat jednotlivé památky. Co se v památce může zobrazit, bylo popsáno výše. Následně lze z památek vytvořit trasu. Trasa bude obsahovat oblast, seznam památek a délku přesunů mezi jednotlivými památkami.

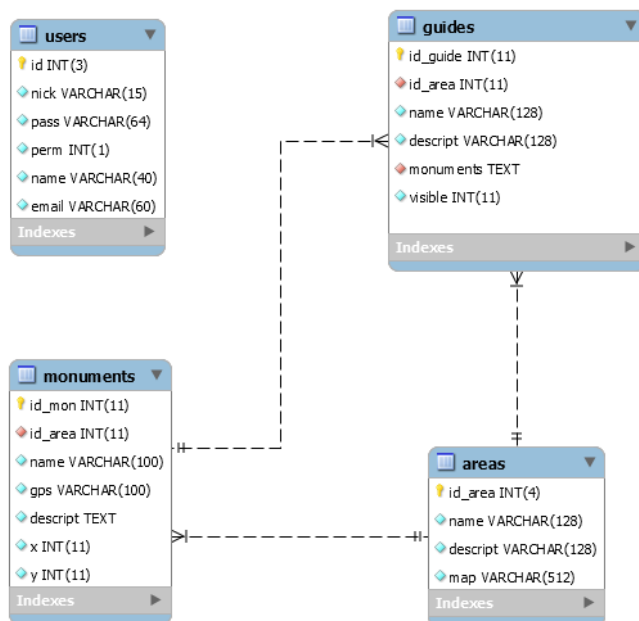
Poslední část se bude týkat administrátorského rozhraní. Administrátor bude mít přístup ke všem činnostem a navíc může upravovat uživatelské účty.

3.8. Tvorba databáze

Pro ukládání dat aplikace jsem zvolil databázový server MySQL. V databázi budou vytvořeny čtyři tabulky, jak je uvedeno na obr. 3.4. Jako znaková sada byla zvolena utf8_czech_ci. V tabulce users budou uloženi uživatelé. Heslo je uloženo pomocí hašování sha256. Tato tabulka nemá žádné další propojení s dalšími tabulkami.

Následující tři tabulky budou sloužit pro ukládání dat o trase. Abychom rozložení tabulek lépe pochopili, uvedeme si příklad. Průvodce bude chtít vytvořit trasu historickým centem Plzně a vytvoří tedy novou oblast v tabulce areas. Zde budou uloženy informace o jménu oblasti, popisu a cesta k mapě. Žádný průvodce se neobejde bez památek. Památky mají svoji tabulku se jménem monuments. Opět zde nalezneme údaje o památce, ale také identifikátor oblasti. Čísla x a y jsou souřadnice památky v mapě z oblasti.

Průvodce určitě bude chtít uložit seznam památek v určitém pořadí do trasy, pro to slouží tabulka guides. Nalezneme zde údaje o trase, identifikátor oblasti a dále například dvourozměrné pole monuments. Jak název napovídá, jsou v tomto poli uloženy identifikátory památek a „čekací doby“. Čekací dobu si můžeme představit jako čas, který je nutný k přechodu mezi památkami.



Obr. 3.4 - Návrh databáze

3.9. Příprava adresářové struktury

Webová aplikace bude pro lepší přehlednost rozdělena do složek. Podle architektury MVC budeme potřebovat tři základní složky pro všechny vrstvy. Naše aplikace také potřebuje složku pro data, CSS, JavaScript a ikony. Zde je uveden abecední seznam složek a jejich význam.

- controllers – zdrojové soubory pro stejně pojmenovanou MVC vrstvu
- css – kaskádové styly pro vzhled aplikace
- data – soubory pro památky rozdělené do složek
 - area – oblasti s mapou
 - monument – multimediální soubory pro památku
- fonts – Glyphicons (obrázkové ikony) pro Bootstrap
- js – složka pro JavaScripty
- models - zdrojové soubory pro stejně pojmenovanou MVC vrstvu
- views - zdrojové soubory pro šablony

3.10. .htaccess

Soubor .htaccess bude obsahovat každá vytvořená složka. Soubor je určený ke konfiguraci Apache serveru a říká, co se má stát, když uživatel zadá specifickou URL adresu [8]. Pro aplikaci použijeme dva různé soubory. První z nich bude uložen v kořenovém adresáři a použijeme ho pro hezké URL adresy, například ve tvaru uvedeném níže.

```
http://guide.raspi/login
```

Zpracování adres tedy bude plně probíhat pod kontrolou PHP a všechny dotazy přeměrujeme na základní soubor index.php, ve kterém se URL zpracuje podle našeho přání. Součástí souboru je i velikost souborů odeslaných pomocí formuláře. Nastavení .htaccess si lze prohlédnout v příloze 2.

Druhým souborem zamezíme uživatelům vstup do složek. Pokud by uživatel zadal do URL název složky, zobrazil by se mu seznam souborů a to nechceme. Proto každá složka, která je uvedena výše, obsahuje nový .htaccess soubor s následujícím nastavením.

```
Deny from all
```

Apache server tedy neobslouží žádný příchozí dotaz kromě dotazů z naší aplikace.

3.11. Bootstrap

Před začátkem tvorby aplikace ještě vyřešíme problém vzhledu. Bootstrap v základní konfiguraci můžeme stáhnout z oficiálních stránek getbootstrap.com. Takto stažený Bootstrap bude mít základní vzhled. Zde se dá využít služeb stránky painstrap.com. Máme možnost importovat výběr barev, které jsme vytvořili zde: color.adobe.com. Následně barvy můžeme upravit a vybrat pro jednotlivé prvky webu. Na závěr nám web vygeneruje připravený archiv s již upravenými CSS soubory. Ohromná výhoda tohoto přístupu tkví v tom, že můžeme vytvořit vzhled stránky a nemusíme sahat přímo do CSS souborů Bootstrapu.

Následně do Bootstrapu přidáme Glyphicons. Jedná se o speciálně vytvořené obrázky pro použití právě v Bootstrapu. Budou nám sloužit hlavně jako grafické dokreslení k textu. Například pro varování můžeme přidat obrázek vykřičníku pouze upravením divu pro výpis. Glyphicons jsou součástí archivu Bootstrap a nalézají se ve složce fonts. Bootstrap vyžaduje pro svoji funkci jQuery, tato JavaScriptová knihovna

slouží pro interakci mezi JavaScriptem a HTML. Stáhnout ji můžeme například zde: jquery.com/download. Poslední věcí, která se týká této kapitoly, bude přidání validátoru [13] pro formuláře. Velice nám to usnadní práci při kontrole formulářů, nemusíme totiž vytvářet svůj script pro kontrolu formuláře během psaní. Stačí rovnou při tvorbě formuláře vložit podmínky, pro které bude toto políčko platné.

3.12. Webová aplikace

Aplikace budu tvořit podle MVC návodu Davida Čápka [8]. Základem aplikace bude již zmíněný soubor `index.php`. Zde zpracujeme jednotlivé URL adresy.

3.12.1. `index.php`

Tento soubor si popíšeme podrobněji. Bude nám sloužit pro načítání kontrolerů a modelů. Nejprve v souboru vytvoříme session a nastavíme kódování.

```
session_start();  
  
mb_internal_encoding("UTF-8");
```

Následně si zjistíme a uložíme složku, ve které se data aplikace nachází. Toto je důležité pokud aplikaci uložíme do jiné složky než kořenové.

```
$cfgvars = array();  
if ($_SERVER['SERVER_NAME'] == 'localhost') {  
    $dir = explode("\\", dirname(__FILE__));  
    $cfgvars['rootDir'] = '/' . array_pop($dir) . '/';  
}  
else  
    $cfgvars['rootDir'] = "  
global $cfgvars;
```

Pokračovat budeme vytvořením funkce pro načtení příslušné třídy pomocí funkce `autoloadFunction`. Využijeme toho, že názvy tříd kontroleru budou končit řetězcem `Controller`. Pokud v URL dostaneme takovou třídu, odkážeme do složky `Controllers`, jinak se podíváme do složky `Models`, zda se zde nenachází požadovaná stránka. Funkci přidáme jako `autoloader`.

```

function autoloadFunction($class)
{
    if (preg_match('/Controller$/', $class))
        require("controllers/" . $class . ".php");
    else
        require("models/" . $class . ".php");
}
spl_autoload_register("autoloadFunction");

```

Následně se v indexu přihlásíme k databázi a vytvoříme router pomocí funkcí, které si vytvoříme později. Jako poslední zavoláme funkci na vykreslení šablony.

```

$connection = new Database("localhost","root","lelazta","bp");

$router = new RouterController();

$router->process(array($_SERVER['REQUEST_URI']), $connection);

$router->createView();

```

3.13. Kontrolery

V teoretické části bylo zmíněno, že kontrolery se starají o komunikaci mezi pohledy, modely a uživatelem. Všechny soubory v této složce budou dědit hlavní třídu a implementovat její abstraktní funkci. Třídy budou ve jméně končit názvem Controllers a implementovat funkce vytvořené v třídách obsažených ve složce models. Zdrojové kódy zde až na výjimky uvádět nebudu.

3.13.1. Controller

Ve složce Controllers si vytvoříme abstraktní třídu, od které budou dědit všechny ostatní kontrolery. Třída obsahuje funkce pro vykreslení správné šablony, hlavičky a patičky ze složky views, následně obsahuje též funkci pro přesměrování. Třída také obsahuje ochranu proti Cross-site scripting útokům. Jedná se o to, že můžeme na stránku vložit škodlivý kód. Tento kód nám například místo uživatelského jména vytvoří tabulku pro přihlášení, kam některý uživatel vyplní své jméno a heslo. Takto se hacker dostane k uživatelskému účtu. Vytvoříme si funkci specialChars pomocí PHP funkce htmlspecialchars, která zamezí použití HTML znaků a tím vložení nebezpečného kódu [8]. Hlavní funkcí této třídy je abstraktní funkce process, kterou musí implementovat všechny ostatní kontrolery. Poslední součástí třídy jsou funkce pro

výpis zpráv. Zprávy jsou komunikačním mechanismem mezi uživatelem a aplikací. Uživatel se tak dozví například potvrzení o vytvoření památky nebo chybová hlášení.

3.13.2. RouterController

Další důležitou součástí složky Controllers je třída RouterController, která dědí Controller. V ní nalezneme funkci `getClassName` pro zjištění názvu třídy z adresy a `parseURL`, která rozdělí adresu podle lomítek. Poslední funkcí, která musí být implementována, je `process`. Zde jsou využity výše zmíněné funkce pro zpracování adresy a zavolání příslušného kontroleru. Dále zde nalezneme předání informací o stránce, předání zpráv a zavolání výpisu stránky.

3.13.3. Třídy admin

V této části se budu zabývat soubory, které mají na začátku jména admin. Tyto soubory obsahují třídy, které jsou dostupné pouze administrátorovi, tj. uživateli s právy nastavenými na číselnou hodnotu 1. První třída `AdminController` je úvodní stránkou administrátora. Třída `adminUsersController` je určena k výpisu všech uživatelů a konečně `AdminUserEdit` pro formulář k editaci informací zvoleného uživatele.

Tímto rozdělením se budou řídit všechny ostatní části aplikace, aby ovládání pro uživatele bylo povědomé a nemusel se pokaždé učit nové návyky. Následně se třídy budou podobat v základním rozložení kódu. Nejprve se načtou modely a zkontroluje se přihlášení uživatelů, poté následuje samotný kód zpracování stránky.

3.13.4. Třídy user

Nejvíce tříd začíná jménem user a jsou přístupné pouze přihlášeným uživatelům a administrátorům. Zde nalezneme správu oblastí, památek a tras. Po přihlášení se setkáme s třídou `UserController`, což je podobná třída jako v administrátorské části a opět slouží jako úvodní obrazovka uživatele. Následně si rozdělíme třídy do třech částí.

Zde budeme postupovat jako při popisu databáze, tj. nejprve vytvoříme oblast nebo ji upravíme. K tomuto účelu slouží třídy s počátkem jména `UserArea`. Zde opět nalezneme třídu pro výpis oblastí `UserAreasController`, třída pro tvorbu oblastí má název `UserAreaCreateController`. Zde nalezneme formulář pro tvorbu památky a její mapu. Mapa musí být ve formátu PNG nebo JPEG ve velikosti nejlépe kolem 1000 pixelů na šířku. Mapu oblasti nalezneme ve složce `data/area/ID oblasti/`. A konečně

třída pro úpravu oblasti `UserAreaEditController`. Tyto třídy jsou nejjednodušší ze všech třech a také nejkratší.

Jako další budeme potřebovat vytvořit správu památek. Třídy k tomu určené nesou jméno začínající na `UserMonuments`. Jako první si opět vypíšeme seznam památek, k jejich výpisu slouží třída `UsersMonumentsController`. Podle návrhu databáze potřebujeme při tvorbě identifikátor oblasti. Proto v první části tvorby památky vybereme oblast z tabulky. Po výběru následuje formulář pro památku. Zde vložíme základní údaje, média a vybereme polohu na mapě. Toto vše je zpracováno ve třídě `UserMonumentsCreateController`. Soubory s multimédií nalezneme ve složkách `audio`, `picture` a `video` v adresáři `data/monuments/ID památky/`. Následují dvě třídy pro úpravu památek a to `UserMonumentsEditController` pro úpravu textových detailů a `UserMonumentsMediaController` pro úpravu multimédií.

Poslední částí uživatelského rozhraní pro památky je tvorba trasy. Třídy pro tvorbu trasy mají začátek jména `UserGuide`. První třída, se kterou se setkáme, bude `UserGuidesController` a opět slouží k výpisu tras. Trasa bude tvořena dvourozměrným polem, vždy zde nalezneme identifikátor památky a čas potřebný k přesunu k památce. Nejprve však vybereme oblast, kde se bude trasa nacházet. Z oblasti vybereme památky a ty následně seřadíme. Seřazení památek funguje pomocí `drag and drop` a je použito rozšíření `jQueryUI`. Toto vše je obslouženo ve třídě `UserGuideCreate`. Předaná data se zpracují a vytvoří se formulář pro podrobnosti trasy a zápis doby přesunů mezi památkami, který je napsán ve třídě `UserGuideFinalController`. Takto připravená data jsou vložena do databáze pomocí `UserGuideSaveController`. Ostatní třídy slouží k úpravě trasy. Úprava je opět rozdělena na dvě části a to úpravu textových detailů a úpravu seznamu památek. Třídy pro úpravu jsou odvozeny od tříd pro tvorbu a mají název začínající na `UserGuideEdit`.

3.13.5. Ostatní třídy

Ostatní třídy nepatří do žádné z výše uvedených kategorií. Nalezneme zde například třídu pro výpis chybové stránky `ErrorController` a hlavní stránku se seznamem viditelných tras `IndexController`. Nalezneme zde tři třídy pro přihlášení, odhlášení a registraci uživatelů. Poslední nezmíněná třída `GuideController` slouží k prohlížení trasy.

3.14. Modely

Modely se starají o komunikaci s databází, kontrolu formulářů, tvorbu složek a výpočty. Ke každému modelu je přiřazena chybová třída pro zachytávání chybových hlášení. Tato chybová hlášení můžeme v kontrolorech dále vypsat uživateli. Mimo klasické modely zde nalezneme obalovací třídu pro práci s databází.

3.14.1. Database

Tato třída je určena čistě jako wrapper pro komunikaci s databází. Nalezneme zde všechny potřebné funkce, které budeme při práci potřebovat pro vkládání řádků, aktualizaci a ovlivňování dat. Třída byla napsána pomocí návodu [8]. Při inicializaci dojde k připojení k databázi a to v souboru index.php.

```
$connection = new Database("localhost","root","leazta","bp");
```

3.14.2. UserModel

Třída obsahuje funkce pro správu uživatelů. Mimo funkce pro registraci, přihlášení, odhlášení a aktualizaci dat uživatele obsahuje třída například funkci checkTime. V aplikaci má uživatel půl hodiny na provedení akce, jinak je odhlášen.

3.14.3. AreaModel

Model slouží k práci s oblastmi. Zde již nalezneme vytváření složek a kontrolu formátu vložených map. K této kontrole je využito tzv. „typu internetového média“, které je PHP schopno rozpoznat. Tato ukázka slouží k ověření, zda je mapa typu PNG nebo JPEG.

```
$finfo = finfo_open(FILEINFO_MIME_TYPE);  
  
$ext_pic = finfo_file($finfo, $map['tmp_name']);  
  
if($ext_pic != "image/jpeg" && $ext_pic != "image/png"){  
    throw new AreaError("Area map must be jpeg or png");  
}
```

Pro tvorbu složky s mapou také budeme potřebovat identifikátor příštího záznamu v tabulce. Číslo identifikátoru použijeme jako název složky. Pro jeho zjištění použijeme funkci getNextId.

```
private function getNextId($connection){  
  
    return $this->area=$connection->queryOne('SHOW TABLE STATUS WHERE  
  
    name = ?', array('areas'));  
  
}
```

3.14.4. MonumentModel

Model určený pro správu památek. Obsahuje podobné funkce jako předchozí modely. V tomto modelu je zpracováno více multimédií a je nutné se vyrovnat se stejnými jmény souborů. V databázi nemáme speciální tabulku s jednotlivými médii a místo toho používáme načítání přímo ze složek, kam soubory ukládáme. Mohlo by se tedy stát, že budeme chtít do složky uložit soubor se stejným jménem. Aby uživatel nemusel řešit přejmenování, uděláme to za něj. K tomu nám budou sloužit dvě funkce a to checkFilename, pomocí které zjistíme, zda se ve složce již stejné jméno nenachází.

```
$newName = $this->renameFile($name);  
  
$return = $this->checkFilename($path, $newName);
```

Pokud nalezneme soubor se stejným názvem, zavoláme funkci renameFile a nové jméno pomocí předchozí funkce rekurzivně zkontrolujeme. Jméno souboru je zvoleno náhodně a to s prefixem file_ a čtyřmístným náhodným číslem.

```
$ext = pathinfo($file, PATHINFO_EXTENSION);  
  
$rand = rand(0, 9999);  
  
$newName = "file_{$rand}.$ext";
```

3.14.5. GuideModel

Posledním modelem je třída pro zpracování trasy. Obsahuje obvyklé metody pro tvorbu, smazání a aktualizaci trasy. Pro ukládání pole památek do databáze je použita funkce serialize, která převede připravené dvourozměrné pole do reprezentace vhodné pro uložení. Také je důležité poznamenat, že pokud smažeme oblast, smažou se i všechny památky a trasy v této oblasti. Stejně je to s památkou, která je následně smazána ze všech tras. Pro tento účel máme dvě speciální funkce. První z nich se jmenuje deleteMonFromGuides a prohledává všechny trasy, pokud nalezne shodu identifikátoru smazané památky a památky v poli, předá klíč v poli funkci

deleteFromArray a památku smaže. V poli následně obnovíme pořadí indexů. Zde nalezneme ukázkou kódu.

```
$array = unserialize($guide['monuments']);  
unset($array[$key]);  
$save = array_values($array);
```

3.15. Pohledy

Pohledy slouží k vykreslení výstupu ve webovém prohlížeči. Je vhodné co nejvíce omezit použití logiky, ale nelze se jí zcela ubránit. Proto nalezneme v pohledech i jiný jazyk než HTML. Také v pohledech nalezneme základní pohled a to site.phtml. Zde voláme ostatní pohledy pro vytvoření stránky. Pohledy dělíme podobně jako kontrolery na pohledy pro administrátory, uživatele a ostatní. Navíc zde nalezneme speciální pohledy s prefixem pro menu či zprávy.

3.15.1. Speciální pohledy

První z těchto souborů `_head.phtml` je určen pro hlavičku stránky. Nalezneme zde údaje o stránce, načtení kaskádových stylů a JavaScriptů. Následně je načtena navigace, která je rozdělena do čtyř souborů začínajících na `_nav`. Je tak učiněno pro rozdílná použití, například pro administrátora či prohlížení trasy. Předposledním souborem je `_message.phtml`. Použití souboru spočívá ve vypsání hlášení uživateli. Závěrečným souborem je `_footer.phtml`.

3.15.2. Administrátorské pohledy

Název pohledů začíná stejně jako u kontrolerů názvem `admin` a jsou určeny pouze pro administrátora. Slouží k výpisu základní stránky administrátora (`admin.phtml`), výpisu seznamu uživatelů (`admin_users.phtml`), viz obr. 3.5 a editaci uživatele (`admin_user_edit.phtml`).



ID	Nick	Name	Email	Edit
39	admin	Admin	admin@admin.com	Edit user Delete
40	testovací	Admin	as@as.com	Edit user Delete

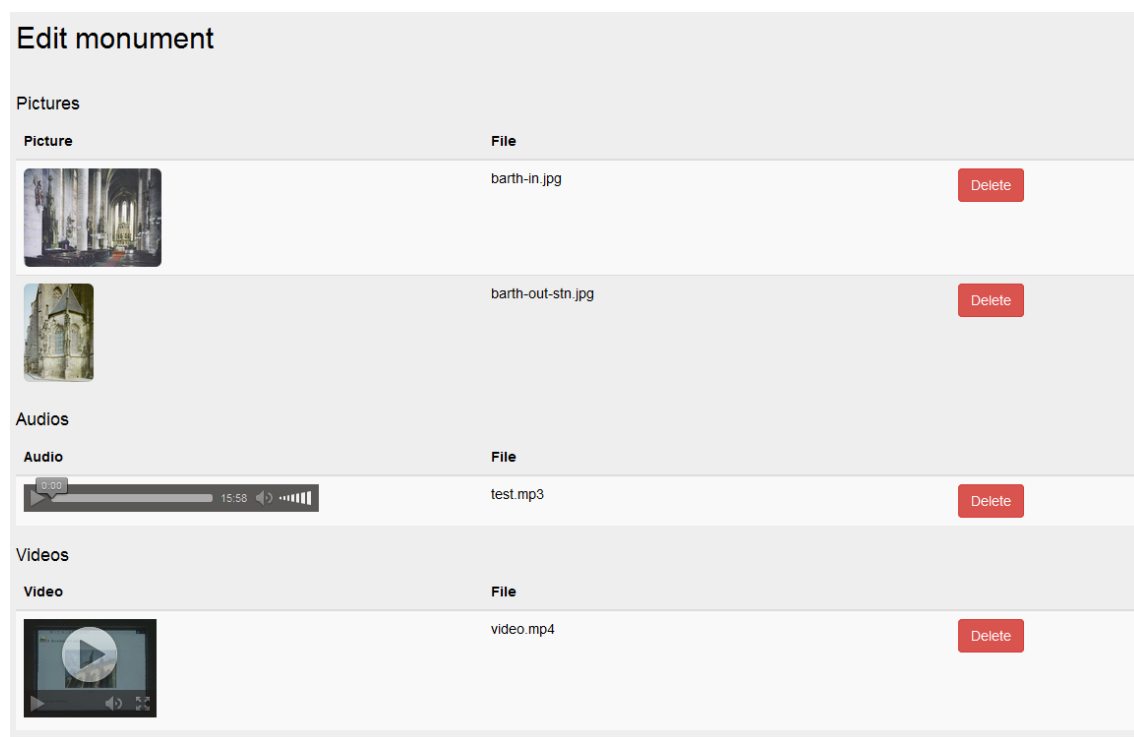
Obr. 3.5 - Výpis uživatelů

3.15.3. Uživatelské pohledy

Jsou podobně jako u kontrolerů největší částí pohledů. Základním pohledem je `user.phtml` a jedná se o základní stránku uživatele. Uživatelské pohledy můžeme opět rozdělit na tři části.

První částí jsou pohledy pro oblasti. Používají se k vypsání oblastí (`user_areas.phtml`), jako formulář pro vytvoření oblasti (`user_area_create.phtml`) a pro editaci (`user_area_edit.phtml`).

Podobnou částí jsou pohledy památek. Vypsání památek zobrazuje pohledy `user_monuments.phtml`. Vypsání formuláře pro tvorbu památky zajišťuje `user_mon_create.phtml`. Zajímavostí tohoto souboru je JavaScript pro výběr bodu z mapy. Více o tomto JavaScriptu se dozvíme v následující kapitole. Je zde také možnost vložit GPS údaje o poloze památky. Tyto údaje nejsou kontrolovány na správnost údajů kvůli různým možnostem zápisů těchto údajů. V památce je také možnost přidat multimediální soubory. Povoleny jsou typy JPEG a PNG pro obrázky, OGG, WAV a MP3 pro zvukové soubory a MP4 a WEBM pro video. Poslední soubory začínající na `user_mod_edit` jsou určeny k úpravám informací o památce. Část editace multimédií u památky můžeme vidět u obr. 3.6.

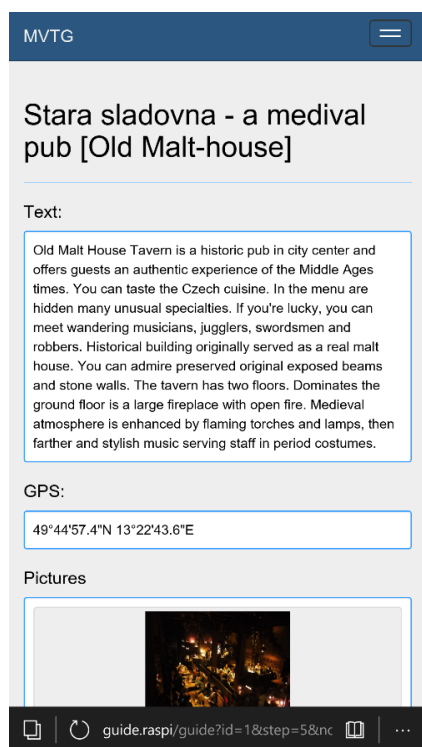


Obr. 3.6 - Editace multimédií

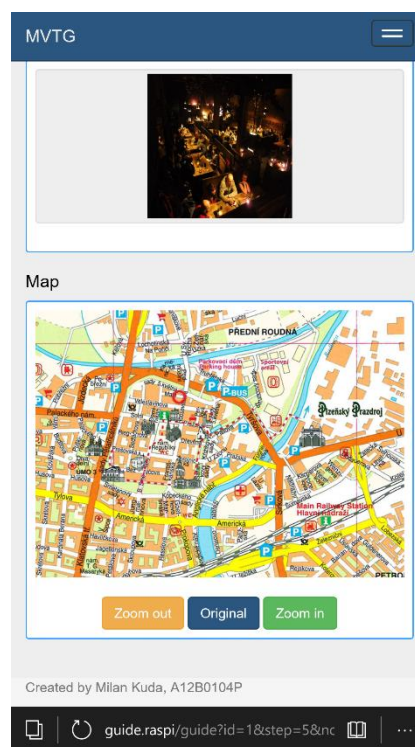
Poslední z uživatelské části jsou pohledy pro tvorbu a správu tras. Nalezneme zde opět pohled pro výpis všech tras `user_guides.phtml`. Jelikož je zde více souborů, vysvětlíme si je postupně jako při vytváření trasy. V první části vybereme všechny památky, které v trase použijeme (`user_guide_sel_mon.phtml`). Památky musíme seřadit tak, jak se nám následně zobrazí v trase (`user_guide_sort.phtml`). Tento pohled využívá systému táhni a pusť pro seřazení památek v seznamu. K uložení tohoto seznamu je potřeba skript pro vytvoření a aktualizaci pole po každé změně seřazení památek. V poslední části se nám zobrazí seřazená trasa a možnost přidat název a popis trasy (`user_guide_final.phtml`). Nalezneme zde i pohledy pro úpravu trasy, které jsou obdobou předchozích pohledů s drobnými úpravami. Názvy těchto pohledů začínají na `user_guide_edit`.

3.15.4. Ostatní pohledy

Jedná se o pohledy, které jsou přístupné bez přihlášení a zároveň obsahují data pro tělo stránky. Patří mezi ně výpis chybové stránky (`error.phtml`), zobrazení seznamu tras pro návštěvníky a hlavní stránka (`index.phtml`), která je rovněž základní stránkou aplikace. Nalezneme zde i stránky pro registraci (`register.phtml`) a stránku pro přihlášení uživatele (`login.phtml`). Posledními pohledy jsou výpis památky v trase (`guide.phtml`, viz obr. 3.7 a 3.8) a stránka pro čekání při přesunu mezi památkami (`wait.phtml`).



Obr. 3.7 - Výpis památky z trasy 1

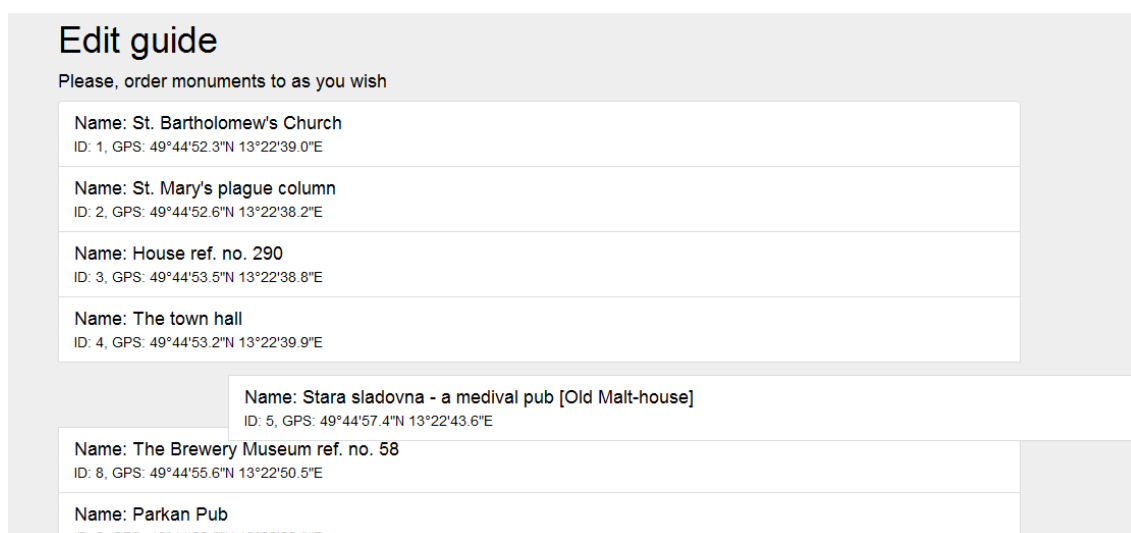


Obr. 3.8 - Výpis památky z trasy 2

3.16. Java skripty

Projekt využívá několik Java skriptů pro zpříjemnění uživatelského rozhraní a zpracování dat. Knihovny Java skriptů nalezneme ve složce js, ale některé funkce jsou zahrnuty přímo v souborech pohledů. Základ pro naši aplikaci tvoří knihovna jQuery ve verzi 1.11.3, určená pro Bootstrap a další funkce této aplikace. Jednou z těchto funkcí je již zmíněné seřazení památek pomocí funkce táhni a pusť. K tomu je využita knihovna jQuery UI rozšířená o následující skript. Funkce skriptu je vidět na obr. 3.9.

```
$("#sortable").sortable({  
  
    create: function () {  
  
        var order = $("#sortable").sortable("toArray");  
  
        document.getElementById('text').value = order;  
  
    },  
  
    update: function (event, ui) {  
  
        var order = $("#sortable").sortable("toArray");  
  
        document.getElementById('text').value = order;  
  
    }  
  
});
```



Obr. 3.9 - Přesun památky v pořadí trasy

Skript je určen pro vytvoření seřazeného pole památek, které následně využijeme při kompletaci trasy. Nalezneme zde knihovny pro Bootstrap a validátor hodnot ve

formuláři. Použití validátoru je velmi snadné a ušetří mnoho času při tvorbě vlastního skriptu pro kontrolu zadaných dat. Níže vidíme ukázkou použití. Stačí tedy v tagu pro vstupní hodnotu zadat parametry a validátor se již postará o kontrolu. Můžeme také vypsat uživateli nápovědu, jak je vidět v tagu span.

```
<input type="text" class="form-control" name="name" placeholder="Insert monument name" data-minlength="3" maxlength="100" required />
<span class="help-block with-errors">Minimum 3 characters</span>
```

Pokud formulář nespĺňuje daná kritéria, není možno jej odeslat.

Máme zde také skript pro obsluhu mapy. Mapu jsem se rozhodl vytvořit co možná nejjednodušší. Naše aplikace je velmi limitována tím, že klient není během provádění připojen k Internetu a nemůže tedy stahovat data ze třetích stran. Pokud by toto bylo možné, lze velmi jednoduše použít mapy od firmy Google a zadávat polohu památky pouze pomocí GPS souřadnic. Rozhodl jsem se vytvořit mapu z jediného obrázku s funkcemi pro přibližování, oddalování a zakreslení polohy památky. K tomuto účelu velmi dobře posloužil zásuvný modul pro jQuery ImageMapster [14]. Nejprve bude ovšem nutné získat polohu památky na mapě. K tomu dochází při tvorbě památky, kdy se využije mapy pro oblast a uživatel na ní pomocí myši zvolí souřadnice na obrázku. Celý skript pro vložení mapy a získání souřadnic lze najít v pohledu user_mon_create.phtml. Zde si ukážeme pouze řádky pro získání souřadnic, které jsou dále zpracovány.

```
x = event.clientX + document.body.scrollLeft + document.documentElement.scrollLeft;
y = event.clientY + document.body.scrollTop + documentElement.scrollTop;
x -= canvas.offsetLeft;
y -= canvas.offsetTop;
```

Takto získaná pozice je automaticky doplněna do formuláře památky a uložena v databázi. Pro samotné vykreslení polohy a funkce pro zoom je vytvořen skript map.js. V něm lze nalézt čtyři připravené funkce. Postup vykreslení mapy je teď poměrně jednoduchý. Následující řádky ukazují mapu a koordináty pro vykreslení bodu. Poslední složkou koordinátu je velikost bodu.

```



<map name="mapa">

<area shape="circle" coords="<?=$details['x'] ?>, <?=$details['y'] ?>, 10"/>

</map>

```

Nyní se v mapě vykreslí červená kružnice kolem bodu, který jsme zvolili. Zbývá tedy přidat tlačítka pro zoom. Jako ukázkou zde uvedu skript pro přiblížení mapy.

```

$('#zvetsit').bind('click', function () {

    width = document.getElementById('mapa').offsetWidth;

    zoom = width * 1.25;

    $('#mapa').mapster('resize', zoom, 0, 1000);

});

```

Jak lze vyčíst ze skriptu, po zavolání funkce se mapa zvětší o 25% své velikosti. Tyto funkce lze najít na obr. 3.8.

3.17. Testování

V následujících kapitolách jsou popsána různá měření a testování, která byla provedena na vytvořeném systému. Hlavní část testování byla věnována měření signálu WiFi, jelikož distribuce signálu ke klientům je jedna z nejdůležitějších částí systému.

3.17.1. Měření signálu WiFi

Měření signálu WiFi mělo demonstrovat různé situace, do kterých se průvodce může při své činnosti dostat. Vysílač byl spolu s Raspberry ukryt v batohu a měření probíhalo mobilním telefonem LG Nexus 5 s programem WiFi analyzer. Měřilo se v jednotkách decibel miliwatt (dBm). Jedná se o poměr síly přijatého signálu s 1 mW (rov. 3.1).

$$\text{dBm} = 10 * \log_{10} \frac{P_1}{1 \text{ mW}}$$

Rov. 3.1 – Výpočet síly přijatého signálu

3.17.2. Měření přímé viditelnosti

Toto měření je podle mého uvážení nejdůležitější. Pohybující se skupina by měla mít většinu času přímý výhled na průvodce, a pokud se ve skupině nebude

pohybovat více lidí, nemělo by signálu nic překážet v cestě. Toto měření je tedy nejpodrobnější a bude nám sloužit jako referenční.

Vzdálenost	1. měření	2. měření	3. měření	průměr
0 metrů	- 35 dBm	- 37 dBm	- 31 dBm	- 34,33 dBm
10 metrů	- 64 dBm	- 62 dBm	- 64 dBm	- 63,33 dBm
20 metrů	- 67 dBm	- 72 dBm	- 68 dBm	- 69 dBm
30 metrů	- 71 dBm	- 70 dBm	- 69 dBm	- 70 dBm
40 metrů	- 75 dBm	- 74 dBm	- 81 dBm	- 76 dBm
50 metrů	- 79 dBm	- 77 dBm	- 81 dBm	- 79 dBm

Tab. 3.1 - Měření WiFi, přímá viditelnost

Vzdálenost 50 metrů by měla být pro skupinu víc než dostačující, ale otestoval jsem i maximální vzdálenost, při které dokázal mobilní telefon udržet stabilní spojení. Tato hodnota činila 116 metrů při signálu - 85 dBm. Tato hodnota mě velice překvapila.

3.17.3. Měření v lesním porostu

V lese jsem simuloval zakrytý přímý výhled na vysílač stromy a také dřevěnou překážkou. Měření probíhalo v řídkém porostu. Výsledky můžeme vidět v následující tabulce 3.2.

Vzdálenost	překážka	1. měření	2. měření	3. měření	průměr
10 metrů	1 strom	- 65 dBm	- 64 dBm	- 66 dBm	- 65 dBm
20 metrů	2 stromy	- 75 dBm	- 80 dBm	- 77 dBm	- 77,33 dBm
30 metrů	3 stromy	- 76 dBm	- 85 dBm	- 82 dBm	- 81 dBm
30 metrů	1 strom + listí	- 83 dBm	- 89 dBm	- 84 dBm	- 85,33 dBm
30 metrů	Strom + dřevěná překážka	- 82 dBm	- 82 dBm	- 92 dBm	- 88,33 dBm

Tab. 3.2 - Měření WiFi, lesní porost

Zde můžeme vidět velký úbytek signálu při překážkách. Při vzdálenosti 30 metrů s více překážkami se již můžeme setkat s výpadky signálu. Změřil jsem také signál na lesní cestě, zde se hodnota pohybovala v hodnotách z měření přímé viditelnosti s úbytkem 3 až 5 dBm. Změna signálu byla zřejmě způsobena odrazy signálu od blízkých stromů.

3.17.4. Měření ve městě

Abych simuloval i druhou část používání průvodce, vydal jsem se změřit signál do města. V prvním měření jsem zopakoval měření přímé viditelnost v nových podmínkách. Podle skenování se na kanálu 4 nenacházela žádná WiFi síť, ale na sousedních kanálech jsem zachytil několik sítí se slabým signálem. Výsledky měření jsou opět uvedeny v tabulce 3.3.

Vzdálenost	1. měření	2. měření	3. měření	průměr
10 metrů	- 66 dBm	- 64 dBm	- 67 dBm	- 65,66 dBm
20 metrů	- 69 dBm	- 75 dBm	- 69 dBm	- 71 dBm
30 metrů	- 75 dBm	- 74 dBm	- 73 dBm	- 78,33 dBm
40 metrů	- 77 dBm	- 78 dBm	- 80 dBm	- 78,33 dBm
50 metrů	- 82 dBm	- 85 dBm	- 83 dBm	- 83,33 dBm

Tab. 3.3 - Měření WiFi, město

Při měření ve městě jsem také otestoval dva druhy překážek a to automobil a roh domu, viz tab. 3.4. Vysílač byl vždy umístěn tak, aby na něj nebyl přímý výhled.

Vzdálenost	překážka	1. měření	2. měření	3. měření	průměr
10 metrů	automobil	- 81 dBm	- 83 dBm	- 81 dBm	- 81,66 dBm
15 metrů	roh domu	- 89 dBm	- 90 dBm	- 89 dBm	- 89,33 dBm

Tab. 3.4 - Měření WiFi, městské překážky

Při testu s rohem domu již nebylo připojení stabilní. Zde může nastat problém při pohybu skupiny mezi zaparkovanými auty. Doporučuji tedy síť na připojených zařízeních si „zapamatovat“, aby po přerušení došlo k automatickému připojení. Poslední provedené měření proběhlo v budově. Zde mě zajímala průchodnost signálu stěnami, přičemž tloušťka zdi byla kolem 50 centimetrů.

Vzdálenost	překážka	1. měření	2. měření	3. měření	průměr
2 metry	1 zeď	- 71 dBm	- 70 dBm	- 65 dBm	- 68,66 dBm
5 metrů	2 zdi	- 74 dBm	- 72 dBm	- 79 dBm	- 75 dBm
8 metrů	3 zdi	- 89 dBm	- 92 dBm	- 93 dBm	- 91,33 dBm

Tab. 3.5 - Měření WiFi, dům

Celkově měření signálu dopadlo uspokojivě. WiFi modul i přes své malé rozměry by měl být schopen dostatečné síly signálu pro menší skupinu lidí.

3.17.5. Měření výdrže baterie

Pro Raspberry Pi jsem vybral baterii XIAOMI PowerBank 10400 mAh. Kapacita baterie při 3,6 V je 10400 mAh, tj. 37,44 Wh. Testování probíhalo na první verzi Raspberry Pi.

Testování proběhlo dvakrát. Při prvním pokusu zůstalo CPU nevytížené a byla zapnuta pouze WiFi. Při druhém pokusu jsem otestoval vytížení CPU programem Stress.

Pomocí tohoto programu na CPU běželo 15 vláken.

```
stress -cpu 5 -io 5 -vm 5 -vm-bytes 10M
```

U naší aplikace je velká výhoda, že Raspberry nemusí mít zapnutý grafický režim, tudíž je úspora energie velká. Výsledky jsou uvedeny v následující tabulce 3.6.

Typ testu	Doba výdrže [hodiny : minuty]
Bez zátěže (zapnutá WiFi)	23 : 50
Se zátěží (100% CPU, WiFi)	19 : 45

Tab. 3.6 - Test výdrže

Testování dopadlo nad míru uspokojivě, baterie má dostatečnou kapacitu pro provozování serveru téměř na celý den. Průvodce by tedy neměl narazit na problém.

3.17.6. Testování v reálném provozu

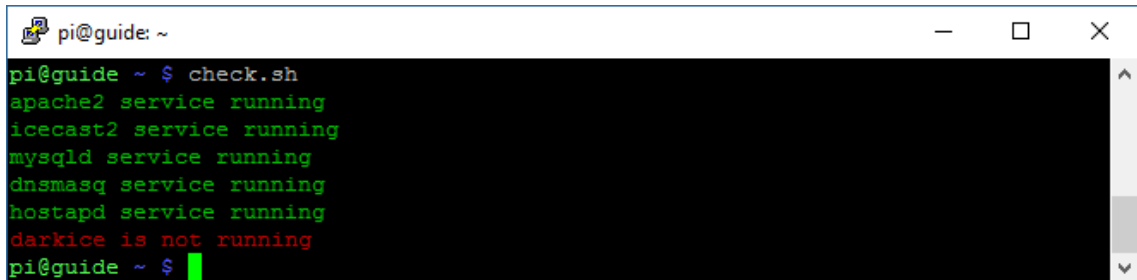
Server a aplikace byly v průběhu vývoje otestovány během 23rd International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision 2015. S pomocí prof. Ing. Václava Skaly, CSc., který dodal podklady pro tvorbu trasy, jsem připravil průvodce Plzní, který kopíroval trasu průvodce. Během testování bylo nalezeno několik nedostatků. Server se při připojení více jak 15 lidí restartoval. WiFi modul by měl tuto zátěž zvládnout. Rozhodli jsme se tedy pro vylepšení zbytku hardware. Byla vyměněna microSD karta za novější, podporující standard class 10. Také bylo pořízeno novější Raspberry Pi 2 model B, které by mělo být s 4 jádrovým procesorem a 1 GB RAM schopnější utáhnout vyšší zátěž.

3.17.7. Testování funkčnosti systémů

Na serveru je vytvořen skript pro kontrolu správného spuštění všech potřebných programů pro běh průvodce. Tento skript je vytvořen pro případ, že by server nefungoval podle očekávání a mohl být snadněji identifikován problém. Skript lze snadno spustit v terminálu následujícím příkazem.

```
check.sh
```

Skript v terminálu poté vypíše, zda jsou všechny vybrané procesy spuštěné, či nikoliv. Výjimkou je program DarkIce, který není spuštěn, pokud při startu serveru není připojena zvuková karta. Tuto situaci můžeme vidět na obrázku obr. 3.9, stejně jako seznam kontrolovaných programů.

A screenshot of a terminal window titled 'pi@guide: ~'. The terminal shows the execution of a script named 'check.sh'. The output lists several services: 'apache2 service running', 'icecast2 service running', 'mysqld service running', 'dnsmasq service running', and 'hostapd service running'. The last line of output is 'darkice is not running', where 'darkice' is in red and 'is not running' is in green. The prompt 'pi@guide ~ \$' is visible at the end of the output.

```
pi@guide ~ $ check.sh
apache2 service running
icecast2 service running
mysqld service running
dnsmasq service running
hostapd service running
darkice is not running
pi@guide ~ $
```

Obr. 3.9 - Testování programů

4. Závěr

V této práci byl vytvořen celkový systém pro průvodcovskou činnost. Nejprve byl vybrán hardware a operační systém. Následně proběhla konfigurace serveru a instalace potřebných aplikací. Byl nastaven přístupový bod pro připojení klientů a vytvořen server pro streamování hlasu z mikrofonu. Součástí práce je také rozbor různých vrstvených architektur a následné použití MVC v aplikační části této práce. Také bylo ukázáno použití databáze MySQL. Ve webové aplikaci byl vytvořen systém pro tvorbu oblastí spolu s mapou, kterou využívají památky i trasa. Podařilo se realizovat multimediální část práce pomocí HTML5 a je tedy možnost vložit obrázky, audio a video soubory a zvolit pozici památky na mapě. Díky tomu lze snadno poskládat trasu pouhým seřazením památek. Pomocí Bootstrapu bylo zajištěno správné zobrazení na všech platformách. Povedlo se realizovat jednoduchou mapu, která uživatelům zobrazuje pozici památky a obsahuje funkce pro zoom. Kompletní uživatelská dokumentace je k nahlédnutí u vedoucího práce.

Práce by se dala dále rozšířit o propracovanější mapu, která by obsahovala celkovou trasu a ukazovala uživateli, v jaké části trasy se nachází. Také by bylo možné vytvořit propracovanější systém pro vkládání multimédií například s jejich náhledy. Nicméně i bez těchto nápadů na rozšíření lze pomocí této aplikace provádět plnohodnotnou průvodcovskou činnost. Tato práce byla poutavá nejen díky poměrně rozsáhlému problému, ale také tím, že jsem měl možnost se dozvědět informace o průvodcovské činnosti a nahlédnout do historie Plzně.

Přehled zkratek

- APT - Advanced Packaging Tool
- ARM - Advanced RISC Machine
- DHCP - Dynamic Host Configuration Protocol
- DNS - Domain Name System
- FAT - File Allocation Table
- GPS - Global Positioning System
- HDMI - High-Definition Multi-media Interface
- HTML - HyperText Markup Language
- IOT - Internet of Things
- IP - Internet Protocol
- JPEG - Joint Photographic Experts Group
- LAN - Local Area Network
- MP3 - MPEG-2 Audio Layer III
- MP4 - MPEG-4 Part 14
- OGG - Ogg Vorbis
- PHP - PHP: Hypertext Preprocessor
- PNG - Portable Network Graphics
- SSH - Secure Shell
- URL - Uniform Resource Locator
- WAV - Waveform Audio File Format
- WSCG - Winter School of Computer Graphics

Literatura

- [1] Rouse, Margaret. techtarget.com. [Online] Duben 2012. [Citace: 12. Listopad 2015.] <http://whatis.techtarget.com/definition/Raspberry-Pi-35-computer>.
- [2] Hayward, David. cnet.com. [Online] 14. Březen 2014. [Citace: 10. Listopad 2015.] <http://www.cnet.com/how-to/25-fun-things-to-do-with-a-raspberry-pi/>.
- [3] Segers, Jens. jenssegers.be. [Online] 15. Únor 2013. [Citace: 15. Listopad 2015.] <http://www.jenssegers.be/43/Realtek-RTL8188-based-access-point-on-Raspberry-Pi>.
- [4] Pecinovský, Rudolf. pecinovsky.cz. [Online] 26. Listopad 2012. [Citace: 14. Listopad 2015.] http://vyuka.pecinovsky.cz/fit/Architektonicke_vzory.pdf.
- [5] Tonar, Jiří. web-integration.info. [Online] 15. Říjen 2013. [Citace: 20. Listopad 2015.] <http://www.web-integration.info/cs/blog/jak-uplatnit-principy-trivrstve-architektury-v-ramci-web-integracniho-projektu/>.
- [6] Turoň, Jan. janturon.cz. [Online] [Citace: 22. Listopad 2015.] <http://janturon.cz/kniha/architektonicke-vzory>.
- [7] Bernard, Borek. zdrojak.cz. [Online] 2009. Května 7. [Citace: 23. Listopad 2015.] <https://www.zdrojak.cz/clanky/uvod-do-architektury-mvc/>.
- [8] Čápka, David. itnetwork.cz. [Online] 20. Říjen 2012. [Citace: 25. Listopad 2015.] <http://www.itnetwork.cz/php/mvc>.
- [9] Dajbych, Václav. dotnetportal.cz. [Online] 21. Duben 2009. [Citace: 2015. Listopad 28.] <http://www.dotnetportal.cz/clanek/4994/MVVM-Model-View-ViewModel>.
- [10] Agius, Anthony. decryption.net.au. [Online] 2014. Květen 2014. [Citace: 12. Prosinec 2015.] <http://decryption.net.au/index.php/2014/05/24/a-very-simple-way-to-use-a-raspberry-pi-as-a-wireless-access-point-not-a-router-with-an-rtl8192cu-based-wi-fi-chipset/>.
- [11] Wright, Stephen. stewright.me. [Online] 22. Září 2012. [Citace: 15. Prosinec 2015.] <https://www.stewright.me/2012/09/tutorial-install-apache-php-and-mysql-on-raspberry-pi/>.
- [12] Müller, Steffen. stmlr.net. [Online] 10. Říjen 2012. [Citace: 15. Únor 2016.] <https://stmlr.net/blog/live-mp3-streaming-from-audio-in-with-darkice-and-icecast2-on-raspberry-pi/>.
- [13] Saffary, Cina. <https://github.com/1000hz>. [Online] 29. Leden 2015. [Citace: 20. Únor 2016.] <https://github.com/1000hz/bootstrap-validator>.

- [14] Treworgy, James. outsharked.com. [Online] 25. Únor 2013. [Citace: 24. Únor 2016.] <http://www.outsharked.com/imagemapster/>.

Přílohy

Příloha 1. konfigurace hostapd

```
# Basic configuration
    interface=wlan0
    ssid=GUIDE
    channel=1
    #bridge=br0
# WPA and WPA2 configuration
    macaddr_acl=0
    auth_algs=1
    ignore_broadcast_ssid=0
    wpa=3
    wpa_passphrase=leleazta
    wpa_key_mgmt=WPA-PSK
    wpa_pairwise=CCMP
    rsn_pairwise=CCMP
# Hardware configuration
    driver=rtl871xdrv
    ieee80211n=1
    hw_mode=g
    device_name=RTL8192CU
    manufacturer=Realtek
```

Příloha 2. konfigurace .htaccess

```
# .htaccess pro nastaveni apache MVTG

php_value upload_max_filesize 40M
php_value post_max_size 200M

Options -Indexes

RewriteEngine On
#RewriteBase /

# front controller
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule !\.(pdf|js|ico|gif|jpg|png|css|rar|zip|tar|.gz)$ index.php [L]

AddType application/x-httpd-php .php .phtml
```

Příloha 3. konfigurace DarkIce

```
# this section describes general aspects of the live streaming session
[general]
duration    = 0          # duration of encoding, in seconds. 0 means forever
bufferSecs  = 8          # size of internal slip buffer, in seconds
reconnect   = yes        # reconnect to the server(s) if disconnected

# this section describes the audio input that will be streamed
[input]
device      = hw:1,0     # Alsa soundcard device for the audio input
sampleRate  = 44100      # sample rate in Hz. try 11025, 22050 or 44100
bitsPerSample = 16       # bits per sample. try 16
channel     = 2          # channels. 1 = mono, 2 = stereo

# this section describes a streaming connection to an IceCast2 server
# there may be up to 8 of these sections, named [icecast2-0] ... [icecast2-7]
# these can be mixed with [icecast-x] and [shoutcast-x] sections
[icecast2-0]
bitrateMode = vbr        # variable bit rate
format      = mp3        # format of the stream: mp3
quality     = 0.4        # quality of the stream sent to the server
server      = localhost  # host name of the server
port        = 8000       # port of the IceCast2 server, usually 8000
password    = leleazta   # source password to the IceCast2 server
mountPoint  = raspi      # mount point of this stream on the IceCast2 server
name        = GUIDE      # name of the stream
description = Mobile virtual tourist guide # description of the stream
url         = http://localhost # URL related to the stream
genre       = my genre    # genre of the stream
public      = no         # advertise this stream?
```


Příloha 4. uživatelská příručka

Spuštění serveru

Spuštění serveru je velmi jednoduché, stačí jej pouze připojit na zdroj elektrické energie. Pokud spustíme server zároveň s připojenou zvukovou kartou, je automaticky zaveden program pro streamování záznamu z mikrofonu.

Připojení k serveru

K serveru je možno se připojit dvěma způsoby. Prvním z nich je použití WiFi, která má nakonfigurovaný název GUIDE a heslo leleazta. Tento způsob je hlavním pro připojení klientů při průvodcovské činnosti.

Druhým způsobem je připojení do sítě pomocí síťového kabelu. Síť musí obsahovat DHCP server pro přidělování adres. Při tomto připojení se počítá s připojením do sítě Internet pro stahování aktualizací a nových programů.

Webová aplikace

Do webové aplikace je možno vstoupit po zadání URL `guide.raspi/`. Zde lze nalézt část webové aplikace pro uživatele, konkrétně vytvořené trasy. Po stisknutí odkazu login se dostaneme na přihlašovací obrazovku. K přihlášení můžeme využít administrátorský účet admin s heslem testovací.

Po přihlášení máme možnost spravovat uživatelské účty, oblasti, památky a trasy. Pod odkazem Users lze uživatele smazat nebo změnit jejich detaily, nikdy ovšem nelze změnit heslo uživatele. V následující stránce Areas určené pro oblasti lze provádět jejich kompletní obsluhu. Při tvorbě, která se nalézá pod tlačítkem Create new area, je nutné zadat mapu. Mapa by měla být ve formátu JPEG či PNG a šířka by měla být uměřená, tj. ideálně kolem 1000 pixelů. Oblast je možné samozřejmě upravovat pomocí tlačítka Edit area. Je nutno zmínit, že smazání oblasti vede k smazání vše památek a tras, které se v dané oblasti nalézají.

Část starající se o památky se ukrývá pod odkazem Monuments. Je podobná částí s oblastmi a opět zde nalezneme kompletní obsluhu. Při tvorbě památek je možno vložit multimédia a to obrázky, video a audio. Posledním krokem při tvorbě památky je vybrání polohy, které se provádí kliknutím do mapy. U památek lze odděleně editovat detaily a multimédia přes tlačítka Edit Details a Edit Multimedia. Při smazání památky dojde k odstranění ve všech trasách.

Poslední webová část má odkaz Guides. Tato část se stará o správu tras. Možnosti jsou stejné jako u předchozích částí, ovšem tvorba trasy je složitější. Při tvorbě (tlačítko Create new guide) jsme vyzváni k vybrání oblasti a následně se vypíše všechny dostupné památky v dané oblasti. Poté lze památky seřadit pomocí táhni a pusť, přičemž můžeme pozorovat změnu pořadí památek v poli, které je umístěno pod seznamem památek. Až nakonec lze zadat jméno a popis trasy. Také se musí vložit doba, kterou zabere přesun od jedné památky k druhé. Pokud je zvolena nula, žádná čekací doba nebude zobrazena. Editace tras je opět rozdělena na dvě části, přičemž v edit guide lze zvolit viditelnost trasy a v edit monumets upravovat rozložení památek či přidávat nové.

Prohlížení trasy je jednoduché, uživatel vybere zvolenou trasu a zobrazí se mu první památka či informace o přesunu k památce. Uživatel může mezi památkami přepínat pomocí tlačítek v menu. Pokud památka obsahuje multimédia, zobrazí se galerie s možností si dané soubory stáhnout. Pod galerií se nalézá mapa s červeným kolečkem, znázorňující kde se daná památka nachází. K možnostem mapy patří i jednoduchý zoom pomocí tlačítek umístěných pod mapou.