



## Zadání bakalářské práce

|                             |  |
|-----------------------------|--|
| <b>Název:</b>               | Vývoj nové správy rádiových stanic pro Czech-American TV     |
| <b>Student:</b>             | Karel Vrabec   |
| <b>Vedoucí:</b>             | Ing. Tomáš Vondra, Ph.D.                                     |
| <b>Studijní program:</b>    | Informatika  |
| <b>Obor / specializace:</b> | Webové a softwarové inženýrství, zaměření Počítačová grafika |
| <b>Katedra:</b>             | Katedra softwarového inženýrství                             |
| <b>Platnost zadání:</b>     | do konce letního semestru 2021/2022                          |

### Pokyny pro vypracování

Czech-American TV je nekomerční televizní stanice, která rozšiřuje povědomí o České republice nejen v USA. Na stránkách [www.catvusa.com](http://www.catvusa.com) nabízí dvě online rádia, Classic a Folk. Správa těchto dvou rádií je však složitá, obtížná a zastaralá.

Ve spolupráci s televizním producentem Johnem Honnerem analyzujte požadavky pro tvorbu nové interaktivní správy. Pomocí moderních webových technologií navrhněte a implementujte řešení v podobě nového pluginu do WordPressu.

Správa umožní upravovat jednotlivá rádia, jejich playlisty a audio soubory. Výstupem je přehrávač, který je možné vložit na webové stránky a který mimo přehrávání písniček navíc zobrazuje doprovodné texty, fotografie, popř. videa z existující databáze. Přehrávač musí pracovat v různých prohlížečích a na různých zařízeních.

Zaměřte se na UI nové správy a přehrávače. Otestujte funkčnost pluginu a provedte test použitelnosti na přispěvatelích Czech-American TV.





**FAKULTA  
INFORMAČNÍCH  
TECHNOLOGIÍ  
ČVUT V PRAZE**

Bakalářská práce

## **Vývoj nové správy rádiových stanic pro Czech-American TV**

***Karel Vrabec***

Katedra softwarového inženýrství  
Vedoucí práce: Ing. Tomáš Vondra, Ph.D.

1. května 2021



---

## **Poděkování**

Chtěl bych poděkovat svému vedoucímu Ing. Tomáši Vondrovi, Ph.D. za jeho cenné rady, čas a odbornou pomoc, kterou mi poskytoval během psaní této bakalářské práce. Poděkování patří i zadavateli Johnu Honnerovi za jeho trpělivost, vstřícnost a zkušenosti, které se mnou sdílel při tvorbě nové správy rádiových stanic. Dále bych chtěl poděkovat své rodině za podporu a následujícím osobám za testování, korekturu či technickou pomoc – Aleš Kocián, Dana Vynikarová, Daniel Pogorzelski, Iva Floyd, Joe Mraz, Katie Lorence Janette, Katie Schon, Matthew Slaboch, Michal Ozogán, Ondřej Guth a další.



---

## **Prohlášení**

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 2373 odst. 2 zákona č. 89/2012 Sb., občanský zákoník, ve znění pozdějších předpisů, tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené.

V Lánech dne 1. května 2021

.....

České vysoké učení technické v Praze

Fakulta informačních technologií

© 2021 Karel Vrabec. Všechna práva vyhrazena.

*Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí a nad rámec oprávnění uvedených v Prohlášení na předchozí straně, je nezbytný souhlas autora.*

### **Odkaz na tuto práci**

Vrabec, Karel. *Vývoj nové správy rádiových stanic pro Czech-American TV.* Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2021.

---

# Abstrakt

Tato bakalářská práce se zabývá tvorbou pluginu do redakčního systému WordPress. Zadavatelem je nezisková organizace Czech-American TV, která propaguje Českou republiku v USA. Webové stránky této organizace svým návštěvníkům nabízí dvě on-line rádia s českou vážnou a folklorní hudbou. Údržba těchto rádií je však složitá a velmi obtížná. Pomocí moderních webových technologií je navrhнуто a vytvořeno zcela nové řešení. Výsledkem práce je plugin do systému WordPress, který dodává novou jednoduchou správu pro řízení zmíněných rádiových stanic a jejich playlistů dle požadavků zadavatele.

**Klíčová slova** plugin, WordPress, Czech-American TV, UI, uživatelská přívětivost, JavaScript, PHP

---

# Abstract

This bachelor's thesis deals with the development of a brand new plugin for the WordPress content management system. The plugin is demanded by the Czech-American TV (a client), a non-profit organization which promotes the Czech Republic in the USA. The organization's website provides two e-radios with the Czech classical and folk music. But the maintenance of these radios is too hard and complicated. Therefore, a new solution is designed and developed with the help of modern web technologies. The result of this work is the WordPress plugin that adds a new interactive administration to manage and control each of the radio stations and their respective playlists very easily.

**Keywords** plugin, WordPress, Czech-American TV, UI, user friendliness, JavaScript, PHP

---

# Obsah

|   |          |
|---|----------|
| <b>Úvod</b>                                 | <b>1</b> |
| <b>1 Cíl</b>                                | <b>3</b> |
| <b>2 Teorie</b>                             | <b>5</b> |
| 2.1 WordPress . . . . .                     | 5        |
| 2.1.1 Knihovna médií . . . . .              | 5        |
| 2.1.2 Uživateli . . . . .                   | 6        |
| 2.1.3 Šablony . . . . .                     | 6        |
| 2.1.4 Plugins . . . . .                     | 6        |
| 2.2 Tvorba pluginů . . . . .                | 7        |
| 2.2.1 Hook . . . . .                        | 8        |
| 2.2.2 Příspěvkový typ . . . . .             | 9        |
| 2.2.3 Taxonomie . . . . .                   | 10       |
| 2.2.4 Navigační menu . . . . .              | 11       |
| 2.2.5 Metadata . . . . .                    | 12       |
| 2.2.6 Meta box . . . . .                    | 12       |
| 2.2.7 Shortcode . . . . .                   | 13       |
| 2.2.8 Best Practices . . . . .              | 14       |
| 2.3 Přehled použitých technologií . . . . . | 15       |
| 2.3.1 Advanced Custom Fields . . . . .      | 15       |
| 2.3.2 Babel . . . . .                       | 16       |
| 2.3.3 Bootstrap . . . . .                   | 16       |
| 2.3.4 Composer . . . . .                    | 16       |
| 2.3.5 Gulp . . . . .                        | 17       |
| 2.3.6 Node Package Manager . . . . .        | 17       |
| 2.3.7 React . . . . .                       | 17       |
| 2.3.8 Sass . . . . .                        | 18       |
| 2.3.9 Swiper . . . . .                      | 18       |

|  |           |
|--|-----------|
| <b>3 Analýza</b>                                       | <b>19</b> |
| 3.1 Zadavatel . . . . .                                | 19        |
| 3.2 Původní stav . . . . .                             | 21        |
| 3.2.1 Rádio Folk . . . . .                             | 21        |
| 3.2.2 Rádio Classic . . . . .                          | 22        |
| 3.2.3 Výstražná hlášení . . . . .                      | 23        |
| 3.2.4 Zjištěné chyby . . . . .                         | 23        |
| 3.3 Požadovaný stav . . . . .                          | 25        |
| 3.3.1 Funkční požadavky . . . . .                      | 25        |
| 3.3.2 Nefunkční požadavky . . . . .                    | 27        |
| 3.3.3 Aktéři . . . . .                                 | 28        |
| 3.3.4 Případy užití . . . . .                          | 28        |
| 3.3.5 Diagram případů užití . . . . .                  | 31        |
| 3.3.6 Doménový model . . . . .                         | 32        |
| 3.3.7 Podobná řešení . . . . .                         | 33        |
| <b>4 Návrh</b>   | <b>35</b> |
| 4.1 Radio Manager . . . . .                            | 35        |
| 4.2 Frontend . . . . .                                 | 36        |
| 4.2.1 Přehrávač . . . . .                              | 36        |
| 4.2.1.1 Název rádiové stanice . . . . .                | 37        |
| 4.2.1.2 Multimédia . . . . .                           | 37        |
| 4.2.1.3 Ovládání . . . . .                             | 38        |
| 4.2.1.4 Výstražná hlášení . . . . .                    | 39        |
| 4.2.2 Kontext . . . . .                                | 40        |
| 4.3 Backend . . . . .                                  | 40        |
| 4.3.1 Rádiové stanice . . . . .                        | 41        |
| 4.3.1.1 Nastavení . . . . .                            | 42        |
| 4.3.1.2 Playlist . . . . .                             | 43        |
| 4.3.1.3 Výstražná hlášení . . . . .                    | 44        |
| 4.3.2 Hudebníci . . . . .                              | 45        |
| 4.3.3 Žánry . . . . .                                  | 46        |
| 4.3.4 Pomoc . . . . .                                  | 47        |
| 4.4 Propojení frontendové a backendové části . . . . . | 47        |
| <b>5 Implementace</b>                                  | <b>49</b> |
| 5.1 Vývoj . . . . .                                    | 49        |
| 5.2 Frontend . . . . .                                 | 49        |
| 5.2.1 Automatizace procesů . . . . .                   | 50        |
| 5.2.1.1 Skripty . . . . .                              | 51        |
| 5.2.1.2 Kaskádové styly . . . . .                      | 51        |
| 5.2.1.3 Sledování změn . . . . .                       | 51        |
| 5.2.2 Komponenty UI . . . . .                          | 52        |
| 5.2.3 Reprezentace dat . . . . .                       | 54        |

|                                   |  |           |
|-----------------------------------|--|-----------|
| 5.2.3.1                           | Použití návrhového vzoru Factory . . . . .         | 57        |
| 5.2.3.2                           | Použití návrhového vzoru Flyweight . . . . .       | 58        |
| 5.3                               | Backend . . . . .                                  | 58        |
| 5.3.1                             | Architektura . . . . .                             | 59        |
| 5.3.1.1                           | Použití návrhového vzoru Facade . . . . .          | 61        |
| 5.3.1.2                           | Použití návrhového vzoru Singleton . . . . .       | 62        |
| 5.4                               | Propojení frontendové a backendové části . . . . . | 63        |
| 5.5                               | Nasazení . . . . .                                 | 65        |
| <b>6</b>                          | <b>Testování</b>                                   | <b>67</b> |
| 6.1                               | Funkčnost . . . . .                                | 67        |
| 6.2                               | Kompatibilita prohlížečů a zařízení . . . . .      | 68        |
| 6.2.1                             | Testované prohlížeče . . . . .                     | 68        |
| 6.2.2                             | Testovaná zařízení . . . . .                       | 69        |
| 6.2.3                             | Vyskytlé problémy . . . . .                        | 69        |
| 6.3                               | Prototypy . . . . .                                | 70        |
| 6.4                               | Použitelnost . . . . .                             | 71        |
| 6.4.1                             | Příprava . . . . .                                 | 71        |
| 6.4.2                             | Průběh . . . . .                                   | 72        |
| 6.4.3                             | Výsledky . . . . .                                 | 72        |
| 6.5                               | Překlady . . . . .                                 | 73        |
| 6.6                               | Shrnutí výsledků . . . . .                         | 74        |
| <b>Závěr</b>                      |  | <b>77</b> |
| <b>Literatura</b>                 |  | <b>79</b> |
| <b>A Seznam použitých zkratek</b> |  | <b>83</b> |
| <b>B Uživatelská příručka</b>     |  | <b>85</b> |
| B.1                               | Dependencies . . . . .                             | 85        |
| B.2                               | Installation . . . . .                             | 85        |
| B.3                               | Uninstallation . . . . .                           | 87        |
| B.4                               | PART 1: Basics . . . . .                           | 88        |
| B.4.1                             | Features . . . . .                                 | 88        |
| B.4.2                             | Different playlist . . . . .                       | 88        |
| B.4.3                             | Shuffle mode . . . . .                             | 89        |
| B.5                               | PART 2: Website . . . . .                          | 90        |
| B.5.1                             | Videos . . . . .                                   | 90        |
| B.5.2                             | Images . . . . .                                   | 91        |
| B.5.3                             | Description . . . . .                              | 91        |
| B.5.4                             | Description (no images) . . . . .                  | 92        |
| B.5.5                             | Logo . . . . .                                     | 93        |
| B.5.6                             | Website posts . . . . .                            | 93        |

|            |  |            |
|------------|--|------------|
| B.5.7      | Warnings . . . . .                       | 94         |
| B.5.8      | Preloader . . . . .                      | 95         |
| B.5.9      | Understand the multimedia rule . . . . . | 95         |
| <b>B.6</b> | <b>PART 3: Administration . . . . .</b>  | <b>96</b>  |
| B.6.1      | Radio stations . . . . .                 | 97         |
| B.6.2      | Editing a radio station . . . . .        | 97         |
| B.6.2.1    | Name . . . . .                           | 97         |
| B.6.2.2    | Logo . . . . .                           | 97         |
| B.6.2.3    | Shortcode . . . . .                      | 98         |
| B.6.2.4    | Publish . . . . .                        | 98         |
| B.6.2.5    | Settings . . . . .                       | 99         |
| B.6.2.6    | Playlist . . . . .                       | 100        |
| B.6.2.7    | Warnings . . . . .                       | 101        |
| B.6.3      | Musicians . . . . .                      | 102        |
| B.6.4      | Editing a musician . . . . .             | 103        |
| B.6.4.1    | Name . . . . .                           | 103        |
| B.6.4.2    | Description . . . . .                    | 103        |
| B.6.4.3    | Featured image . . . . .                 | 103        |
| B.6.4.4    | Genres . . . . .                         | 104        |
| B.6.4.5    | Publish . . . . .                        | 104        |
| B.6.4.6    | Images . . . . .                         | 105        |
| B.6.4.7    | Introductions . . . . .                  | 106        |
| B.6.4.8    | Recordings . . . . .                     | 107        |
| B.6.5      | Genres . . . . .                         | 108        |
| <b>B.7</b> | <b>PART 4: Bonus features . . . . .</b>  | <b>110</b> |
| B.7.1      | Skipping . . . . .                       | 110        |
| B.7.2      | Clever images . . . . .                  | 110        |
| B.7.3      | Extended logo . . . . .                  | 110        |
| <b>B.8</b> | <b>Contact . . . . .</b>                 | <b>111</b> |
| <b>C</b>   | <b>Testovací scénáře</b>                 | <b>113</b> |
| C.1        | Tvorba hudebních žánrů . . . . .         | 113        |
| C.2        | Tvorba hudebníků . . . . .               | 114        |
| C.3        | Tvorba rádiových stanic . . . . .        | 116        |
| <b>D</b>   | <b>Obsah přiloženého CD</b>              | <b>121</b> |

---

# Seznam obrázků

|   |    |
|---|----|
| 2.1 Jednoduchý wireframe prvku Repeater Field . . . . .   | 15 |
| 3.1 Logo CATV USA [24] . . . . .                          | 19 |
| 3.2 Webové stránky CATV USA . . . . .                     | 20 |
| 3.3 Ukázka rádia Folk . . . . .                           | 21 |
| 3.4 Ukázka rádia Classic . . . . .                        | 22 |
| 3.5 Přehled funkčních požadavků . . . . .                 | 25 |
| 3.6 Přehled nefunkčních požadavků . . . . .               | 27 |
| 3.7 Mapování požadavků na případy užití . . . . .         | 28 |
| 3.8 Přehled případů užití . . . . .                       | 28 |
| 3.9 Diagram případů užití . . . . .                       | 32 |
| 3.10 Analytický doménový model . . . . .                  | 33 |
| 4.1 Ukázka rádia Classic . . . . .                        | 36 |
| 4.2 První část přehrávače . . . . .                       | 37 |
| 4.3 Druhá část přehrávače . . . . .                       | 38 |
| 4.4 Poslední část přehrávače . . . . .                    | 39 |
| 4.5 Výstražné hlášení . . . . .                           | 39 |
| 4.6 Ukázka přehrávačů na mobilních telefonech . . . . .   | 40 |
| 4.7 Rozhraní pluginu v administraci . . . . .             | 41 |
| 4.8 Rozložení prvků v příspěvku rádiové stanice . . . . . | 42 |
| 4.9 Nastavení rádiové stanice . . . . .                   | 43 |
| 4.10 Playlist rádiové stanice . . . . .                   | 44 |
| 4.11 Výstražná hlášení rádiové stanice . . . . .          | 45 |
| 4.12 Rozložení prvků v příspěvku hudebníka . . . . .      | 46 |
| 4.13 Shortcode v příspěvku rádiové stanice . . . . .      | 47 |
| 5.1 Diagram Gulp tasks . . . . .                          | 50 |
| 5.2 Diagram React komponent . . . . .                     | 53 |
| 5.3 Diagram tříd frontendové části . . . . .              | 55 |

|     |   |    |
|-----|---|----|
| 5.4 | Diagram tříd backendové části           | 60 |
| 6.1 | Zachovaná fotografie staršího prototypu | 70 |

---

# Seznam kódů

|     |  |    |
|-----|--|----|
| 2.1 | Hlavičkový komentář pluginu . . . . .                      | 7  |
| 2.2 | Registrace vlastní funkce k akci . . . . .                 | 8  |
| 2.3 | Registrace vlastního příspěvkového typu . . . . .          | 9  |
| 2.4 | Registrace vlastní taxonomie . . . . .                     | 10 |
| 2.5 | Tvorba vlastního menu . . . . .                            | 11 |
| 2.6 | Tvorba vlastních metadat . . . . .                         | 12 |
| 2.7 | Tvorba vlastního meta boxu . . . . .                       | 13 |
| 2.8 | Tvorba vlastního shortcode . . . . .                       | 14 |
| 5.1 | Statická tovární metoda pro tvorbu prvků seznamů . . . . . | 57 |
| 5.2 | Továrna musí váhy na hudebníky . . . . .                   | 58 |
| 5.3 | Fasáda pro snadné řízení subsystémů . . . . .              | 62 |
| 5.4 | Subsystém pro tvorbu příspěvkových typů . . . . .          | 63 |



---

# Úvod

Redakční systém WordPress je mezi tvůrci webových stránek a aplikací dobře známým a zavedeným pojmem. Díky jeho jednoduchosti, kompatibilitě a přizpůsobitelnosti na počítačích a mobilních zařízeních lze spravovat webové stránky mnohem jednodušeji než kdy dříve. Teprve s rozšiřitelností ve formě pluginů se ze systému WordPress stává velice sofistikovaný nástroj, který umožňuje vytvářet mnohem rozsáhlejší webové aplikace. Není proto překvapivé, že je mezi svými uživateli také oblíbený. Díky své promyšlenosti jednoznačně patří mezi nejpopulárnější systémy pro správu obsahu na světě.

Czech-American TV je televizní stanice, která prostřednictvím systému WordPress spravuje své webové stránky. Na těch se nachází např. archiv vzdělávacích pořadů o České republice, články o českých tradicích nebo videa s českými recepty. Kromě toho stránky nabízí i dvě rádiové stanice s českou folklorní a vážnou hudbou, které může návštěvník na webu poslouchat. Rádia jsou vestavěna v současné vzhledové šabloně a jejich fungování vykazuje velké množství chyb. Vzhledem k tomu, že Czech-American TV připravuje novou vzhledovou šablonu, představuje současný stav rádií velice závažnou komplikaci a brzdu vývoje webové stránky. Z tohoto důvodu byl vznesen požadavek na vytvoření nové správy rádiových stanic jako vyčleněného pluginu do systému WordPress.

V teoretické části práce je čtenář seznámen se systémem WordPress a tvorbou pluginů. V této části práce se nachází i definice všech potřebných pojmu a stručný popis moderních webových technologií, které byly použity při implementaci. Součástí je i analýza původního a požadovaného stavu rádiových stanic včetně analýzy zadavatele.

Praktická část práce se skládá ze tří kapitol. První kapitolou je návrh nového pluginu rozdělený na frontend (webová část), backend (administrační část) a propojení těchto dvou částí. Druhou kapitolou je implementace pluginu pomocí moderních webových technologií, ve které jsou popsány implementační detaily výsledku. Třetí a zároveň poslední kapitolou je testování nového plu-

## ÚVOD

---

ginu především z hlediska správného fungování, použitelnosti a kompatibility prohlížečů a zařízení. Kapitoly jsou doplněny o pomocné diagramy, náčrty a wireframes, aby si čtenář mohl snáze představit danou problematiku.

Výsledek práce zprístupní administrátorům Czech-American TV doposud nepřístupné a těžce ovladatelné části jejich webové stránky – rádiové stanice. Každou z nich plně automatizuje a oprostí od pravidelné údržby. Usnadní přidávání, úpravu a odebírání rádií pro další zamýšlené rádiové stanice. Ulehčí práci vývojářům, kteří se při tvorbě nové vzhledové šablony nebudou muset rádii tolik zabývat. Navíc přispěje k dlouhodobému cíli Czech-American TV plně automatizovat své webové stránky a zpřístupnit každou jejich část obyčejnému administrátorovi. Zejména však podpoří tuto neziskovou organizaci v propagaci České republiky (a její hudby) v zahraničí.

# KAPITOLA **1**

---

## Cíl

Hlavním cílem této bakalářské práce je vytvořit nový zásuvný modul do systému WordPress, který rozšíří webovou stránku Czech-American TV o nové funkcionality. Mezi ně patří chybějící správa pro řízení rádiových stanic, která umožní upravovat jednotlivá rádia, jejich nastavení, playlisty a nahrávky. Výstupem každé rádiové stanice je přehrávač, který je možné vložit do libovolného příspěvku v systému WordPress. Jeho vzhled a logika (tj. chování a fungování v různých situacích) patří mezi další a poslední funkcionality dodané pluginem. Přehrávač je kromě přehrávání nahrávek schopen zobrazit doprovodné texty, fotografie, příp. videa z existujícího úložiště. Tento hlavní cíl zahrnuje několik dílčích cílů.

Prvním cílem je analýza požadavků zadavatele na novou správu a logiku rádií. Do tohoto cíle spadá i seznámení se zadavatelem jako takovým, jeho krátkodobými i dlouhodobými cíli a prostředím jeho webové stránky.

Druhým cílem je návrh a implementace nového pluginu pomocí moderních webových technologií (se zaměřením na uživatelské rozhraní jak backendové, tak frontendové části).

Třetím cílem je automatizace fungování rádiových stanic s pohodlnou a nezáročnou údržbou. Důležité je, aby správa byla jednoduchá na používání a přehledná i pro netechnické uživatele.

Posledním cílem je otestování funkčnosti nového pluginu, kompatibility s různými prohlížeči a zařízeními včetně provedení testu použitelnosti na přispěvatelích Czech-American TV.



# KAPITOLA 2

---

## Teorie

Tato kapitola je věnována popisu systému WordPress se zaměřením na tvorbu pluginů. Součástí je také přehled všech technologií, které byly použity při vývoji kýženého pluginu. Kapitola slouží k uvedení čtenáře do zkoumané problematiky a všech následujících kapitol.

### 2.1 WordPress

WordPress je svobodný redakční systém pro správu webových stránek a aplikací, který je jedním z nejpopulárnějších CMS (tj. „Content Management System“) na světě. V současné době je zaveden přibližně na 40 % všech webových stránek a má tak vysoký náskok před svými konkurenty, jakými jsou např. Wix, Joomla nebo Squarespace.

Předností systému je zejména jednoduchá, přehledná a rozšířitelná webová správa, kterou je možné obsluhovat bez znalosti programování. Systém je napsán pomocí skriptovacího jazyka PHP (tj. „PHP: Hypertext Preprocessor“) v kombinaci se systémem řízení báze dat MySQL (tj. „My Structured Query Language“). Rozvíjet a udržovat jej pomáhá vcelku rozsáhlá komunita vývojářů, tvůrců a dobrovolníků, kteří pravidelně pořádají setkání v různých městech celého světa .

Mezi nejdůležitější funkcionality, kterými WordPress disponuje, patří především tvorba, editace a publikování webových článků. V administraci k tomu lze použít buď klasický WYSIWYG (tj. „What You See Is What You Get“) editor ve formě pluginu, nebo novější editor Gutenberg, který je v systému WordPress oficiálně od verze 5.0 .

#### 2.1.1 Knihovna médií

Psaní webových článků v systému WordPress lze doplnit o různé obrázky, hudbu, videa či ostatní soubory, jako např. PDF (tj. „Portable Document Format“). Zároveň je možné vytvářet variace těchto souborů v podobě galerií

## 2. TEORIE

---

či audio nebo video playlistů. Pro správu těchto typů souborů slouží v systému WordPress knihovna médií (tzv. „Media Library“).

Soubory lze v knihovně třídit podle času nahrání (tj. měsíce a roku) a typu (tj. obrázky, audio, video atd.). Každému souboru lze navíc nastavit různé meta informace (např. alternativní text pro obrázky či album pro audio soubory). Obrázky lze upravovat pomocí jednoduchého vestavěného editoru, který umožňuje ořezávat, převracet, rotovat či škálovat daný obrázek. Soubory lze do webových článků vkládat z knihovny médií nebo přímo z počítače. Další možností je vkládání souborů pomocí URL (tj. „Uniform Resource Locator“) adresy (např. u videí ze serverů třetích stran).

Výhodou knihovny médií je především snadný přenos souborů do objektových úložišť (pomocí k tomu přizpůsobených WordPress pluginů). Nevýhodou jsou pak omezené možnosti organizace a hierarchie souborů.

### 2.1.2 Uživatelé

Ke správě webových stránek v systému WordPress může přistupovat více uživatelů. Každý uživatel má svůj účet, ke kterému se přihlašuje pomocí svého uživatelského jména (nebo e-mailové adresy) a hesla. Tomuto účtu je přiřazena určitá role (tzv. „Role“), která představuje sadu oprávnění (tzv. „Capabilities“) daného uživatele [4].

WordPress nabízí 6 základních uživatelských rolí – Super Admin, Administrator, Editor, Author, Contributor a Subscriber. Mezi jejich oprávnění patří kupříkladu `edit_posts` (schopnost spravovat příspěvky a komentáře), `activate_plugins` (schopnost spravovat pluginy) nebo také `manage_options` (schopnost spravovat základní nastavení). [4]

V základní verzi systému WordPress lze pouze přidávat, upravovat a mazat jednotlivé uživatelské účty. Pomocí pluginů, jako je např. User Role Editor, však lze spravovat i vlastní role a oprávnění.

### 2.1.3 Šablony

Důležitou součástí každé webové stránky je její vzhledová šablona. Dle [5] by webový design nikdy neměl být podceňován, jelikož v závislosti na jeho kvalitě je v návštěvníkovi vzbuzovaná mnohem větší důvěra. WordPress svým uživatelům v současné době nabízí výběr z přibližně 8 tisíc responzivních šablon, které lze stáhnout a nainstalovat přímo v administraci [6]. Ty mohou zjednodušit a urychlit práci ohledně vzhledu celého webu, a to zejména vývojářům. Mezi nejznámější šablony patří např. Twenty Twenty nebo Astra. Lze si však vytvořit i šablonu vlastní.

### 2.1.4 Pluginsy

Základní funkce systému WordPress nemusí vždy postačovat. Z tohoto důvodu jsou k dispozici různé pluginy (neboli zásuvné moduly), které lze do systému

WordPress jednoduše doinstalovat. V současné době je k dispozici přes 58 tisíc pluginů pro WordPress [7]. Jedná se o samostatné moduly, které rozšiřují různé oblasti systému. Mezi nejznámější z nich patří např. Classic Editor, WPForms nebo Advanced Custom Fields. Lze si však vytvořit i plugin vlastní.

## 2.2 Tvorba pluginů

Základní modul do systému WordPress je tvořen převážně PHP kódem, ale může být doplněn i o HTML (tj. „HyperText Markup Language“), CSS (tj. „Cascading Style Sheets“) nebo JS (tj. „JavaScript“). Pro vytvoření nového pluginu se podle [8] stačí držet následujících kroků:

1. Vytvořit novou složku a pojmenovat ji podle názvu kýzeného pluginu (např. „muj-plugin“).
2. Otevřít tuto složku a v té vytvořit nový PHP soubor, jehož název se shoduje s názvem složky (tj. „muj-plugin.php“).
3. Na začátek nového PHP souboru přidat hlavičkový komentář, který systému WordPress předá všechny základní informace o pluginu.

```
<?php
```

```
/**  
 * Plugin Name: Název pluginu  
 * Description: Popis pluginu  
 * Version: Verze pluginu ve tvaru X.X.X  
 * Author: Jméno autora  
 * License: Název licence  
 * ...  
 */
```

```
?>
```

Kód 2.1: Hlavičkový komentář pluginu

4. Otevřít složku „wp-content“ a dále složku „plugins“ v základní instalaci systému WordPress.
5. Vytvořenou složku „muj-plugin“ s PHP souborem vložit do složky „plugins“, kde se nachází i ostatní WordPress pluginy.

Výsledkem tohoto postupu je jednoduchý WordPress plugin, který je nyní dostupný ve výpisu všech pluginů v tomto systému (po kliknutí na položku „Plugins“ v levém navigačním menu v administraci). Základní PHP soubor

## 2. TEORIE

---

s hlavičkou, který představuje samotný plugin, lze rozšířit o vlastní PHP funkce a další kód. Podobně i hlavičkový komentář lze rozšířit o další informace (např. požadovaná verze systému WordPress či PHP, webová stránka pluginu nebo autora atp.).

Složitost výsledného pluginu velice záleží na tom, k čemu byl určen. Příkladem budiž plugin Hello Dolly s pouhými 82 řádky [8]. Lze tak vytvářet jednoduché pluginy s jedním PHP souborem, nebo pluginy složité, jejichž kód musí být členěn do více souborů a složek, a navíc obsahuje i HTML, CSS a JS.

### 2.2.1 Hook

Hook je označení pro konkrétní událost, která nastává v určitém momentu běhu systému WordPress. Představuje způsob, jakým lze integrovat vlastní kód do systému bez nutnosti měnit systémové soubory [8]. Mezi hooky patří kupříkladu `wp_loaded` (nastává po kompletním načtení systému WordPress), `get_footer` (zavolán před načtením patičky webu) nebo `the_title` (vyvolán pro úpravy nadpisu článků) [9]. WordPress poskytuje rozhraní Plugin API, jehož součástí jsou funkce uvedené dále v této podkapitole. Ty umožňují přiřadit vlastní funkce k daným hookům. Hooky se rozlišují na akce a filtry [8].

Akce se používají pro provádění jednorázových činností (např. vložení dat do databáze, vytvoření nové sekce v navigačním menu atp.). Nejdříve je nutné vytvořit vlastní funkci (s vlastním kódem) bez návratové hodnoty. Tu lze registrovat k dané akci pomocí funkce `add_action()`. [8]

```
<?php  
  
add_action(  
    string $tag,  
    callable $function_to_add,  
    int $priority = 10,  
    int $accepted_args = 1  
);  
  
?>
```

Kód 2.2: Registrace vlastní funkce k akci

Parametr `$tag` představuje název hooku (akce), `$function_to_add` je název registrované funkce s vlastním kódem. Mezi zbývající parametry patří priorita `$priority`, která se používá v případě, kdy je na daný hook registrováno více funkcí. Dále počet argumentů `$accepted_args` přijímaných registrovanou funkcí. [10]

Filtry slouží pro úpravu existujících dat (např. nadpisu či obsahu článků). Nejdříve je nutné vytvořit vlastní funkci, která přijme, modifikuje a vrátí určitá data. Podobně jako u akcí je potřebné tuto funkci zaregistrovat k danému

filtru. To lze provést pomocí funkce `add_filter()`, která přijímá stejné parametry jako funkce `add_action()`. [8]

Po registraci vlastní funkce k hooku (akci či filtru) je tato funkce zavolána, jakmile nastane daný hook. Vlastní funkce může přijímat parametry specifické pro danou akci či filtr. Tyto parametry jsou definovány daným hookem a lze je dohledat v oficiální dokumentaci Plugin API.

Při tvorbě pluginu lze použít následující základní hooky:

- `activation_hook` – zavolán při aktivaci pluginu,
- `deactivation_hook` – zavolán při deaktivaci pluginu,
- `uninstall_hook` – zavolán při odinstalaci pluginu. [8]

Ve vlastních funkcích připojených k těmto hookům lze řešit např. inicializaci pluginu nebo smazání vytvořených dat z databáze. Registrace vlastních funkcí se provádí pomocí funkce `register_<název_hooku>()`. Této funkci je nutné předat nejen samotný callback, ale i cestu k hlavnímu PHP souboru (daného pluginu), který obsahuje hlavičkový komentář. [10]

### 2.2.2 Příspěvkový typ

Příspěvkový typ (v angličtině „Post Type“) je způsob, jakým lze v systému WordPress třídit obsah. Používá se pro vytváření archivů s tematicky podobnými příspěvky (např. knihy, auta nebo recepty). WordPress ve výchozím nastavení nabízí několik těchto příspěvkových typů, jako např. `post`, `page`, `attachment` nebo `revision`. Pomocí funkce `register_post_type()` je však možné si zaregistrovat i vlastní příspěvkový typ. [11]

<?php

```
register_post_type(  
    string $post_type,  
    array|string $args = array(),  
);  
  
?>
```

Kód 2.3: Registrace vlastního příspěvkového typu

Parametr `$post_type` značí identifikátor příspěvkového typu (kupříkladu „book“, „car“ nebo „recipe“). Tento identifikátor zároveň slouží jako slug (část URL adresy, pod kterou je daný příspěvkový typ přístupný). `$args` je asociační pole, které obsahuje různé parametry specifikující tento příspěvkový typ. Mezi ně patří např. název (`label`), popis (`description`), přístupnost (`public`) nebo možnost vytvářet hierarchie z příspěvků (`hierarchical`). [10]

## 2. TEORIE

---

Registraci nového příspěvkového typu je vhodné provést ve vlastní funkci, která je pomocí funkce `add_action()` přiřazena hooku `init` [8]. Po zaregistrování příspěvkového typu se v levém navigačním menu v administraci objeví nová položka s jeho názvem. Jedná se o administrační rozhraní, kde lze spravovat veškeré příspěvky tohoto typu. Tvorbu příspěvkových typů lze výrazně zjednodušit pomocí WordPress pluginů, jako např. Custom Post Type UI.

### 2.2.3 Taxonomie

Taxonomie je dalším způsobem, jakým lze v systému WordPress seskupovat obsah. Skládá se z prvků, které se nazývají termíny. Mezi taxonomií a příspěvkovým typem je ovšem rozdíl. Příspěvkový typ zastupuje nějaký obsah (např. množinu příspěvků), zatímco pomocí taxonomie (resp. jejích termínů) lze dále trádit obsah v rámci jednoho či několika příspěvkových typů. [11]

Jako příklad poslouží dva příspěvkové typy – česká města a české recepty. Řekněme, že na webové stránce existuje několik příspěvků o českých městech a receptech. Taxonomií jsou české kraje a mezi její termíny konkrétně patří „Středočeský kraj“, „Karlovarský kraj“ atd. Pokud je uvedená taxonomie aktivovaná pro oba zmíněné příspěvkové typy, pak lze české recepty a města přiřazovat daným krajům během úprav těchto příspěvků.

Systém WordPress ve výchozím nastavení nabízí dvě taxonomie – kategorie a tagy [12]. Pomocí funkce `register_taxonomy()` je však možné si zaregistrovat i vlastní taxonomii.

<?php

```
register_taxonomy(  
    string $taxonomy,  
    array|string $object_type,  
    array|string $args = array()  
);  
  
?>
```

Kód 2.4: Registrace vlastní taxonomie

Parametr `$taxonomy` značí identifikátor taxonomie (kupříkladu „region“). `$object_type` reprezentuje identifikátory příspěvkových typů, pro které má být daná taxonomie aktivována. `$args` je asociativní pole, které obsahuje různé parametry specifikující tuto taxonomii. Mezi ně patří např. přístupnost (`public`) či možnost vytvářet hierarchie z termínů (`hierarchical`). [10]

Podobně jako u příspěvkového typu je vhodné provést registraci nové taxonomie ve vlastní funkci, a tu následně pomocí funkce `add_action()` přiřadit hooku `init` [8]. Po registraci taxonomie se u položky daného příspěvkového

typu v levém navigačním menu objeví položka dané taxonomie. Jedná se o administrativní rozhraní, kde lze spravovat jednotlivé termíny (příp. sledovat, kolik příspěvků je k daným termínům momentálně přiřazeno).

#### 2.2.4 Navigační menu

WordPress umožňuje přidávat položky do navigačního menu, které se nachází v levé části administrace tohoto systému. Toto navigační menu slouží jako hlavní rozcestník a obsahuje např. položky jednotlivých příspěvkových typů, pluginů, vzhledu, nastavení nebo seznamu uživatelů. Každá taková položka (tzv. menu) může obsahovat další podpoložky (tzv. submenu) [8]. V kontextu tvorby pluginů tak lze vytvářet vcelku přehledná rozhraní, pomocí kterých se administrátor snadněji přemístí do nastavení daného pluginu. Novou položku lze přidat pomocí funkce `add_menu_page()` [8].

```
<?php  
  
add_menu_page(  
    string $page_title,  
    string $menu_title,  
    string $capability,  
    string $menu_slug,  
    callable $function = '',  
    string $icon_url = '',  
    int $position = null  
);  
  
?>
```

Kód 2.5: Tvorba vlastního menu

Parametr `$page_title` je hlavní titulek, který se objeví v záložce prohlížeče po kliknutí na tuto položku. `$menu_title` je název položky v navigačním menu. `$capability` je minimální oprávnění uživatelů, kteří tuto položku uvidí (např. `edit_posts` pro editory). `$menu_slug` je identifikátor dané položky. `$function` je jméno vlastní funkce, která obsahuje HTML obsah této položky. `$icon_url` je identifikátor ikonky, která se u položky objeví. `$position` je pořadí položky v navigačním menu. [10]

Registraci nové položky je vhodné provést ve vlastní funkci, a tu následně zaregistrovat pomocí funkce `add_action()` pro hook `admin_menu` [8]. Podpoložky lze vytvářet pomocí funkce `add_submenu_page()` s obdobnými parametry jako při tvorbě vlastního menu. Navíc je však nutné předat identifikátor rodičovské položky [10].

## 2. TEORIE

---

### 2.2.5 Metadata

Metadata jsou libovolné informace přiřazené k příspěvkům, uživatelům, komentářům či termínům taxonomií [8]. Jedná se o dvojice klíč/hodnota, kde klíč představuje název metadat a hodnota zastupuje informaci přiřazenou tomuto klíči [13]. Metadata lze přidávat pomocí funkce `add_post_meta()` [8].

```
<?php
```

```
add_post_meta(  
    int $post_id,  
    string $meta_key,  
    mixed $meta_value,  
    bool $unique = false  
) ;
```

```
?>
```

Kód 2.6: Tvorba vlastních metadat

Parametr `$post_id` značí identifikátor příspěvku, kterému jsou nová metadata přiřazena. `$meta_key` je klíč metadat. `$meta_value` je hodnota metadat. Pokud je `$unique` nastavený na hodnotu `false`, pak lze přidávat metadata se stejným klíčem. [10]

V systému WordPress existuje rozhraní pro správu metadat u každého příspěvku. Jedná se o box s názvem Custom Fields (v překladu „uživatelská pole“), který je nutné nejdříve aktivovat v nastavení „Screen Options“. V tomto boxu lze jednoduše a bez použití kódu přidávat, upravovat a mazat jednotlivá metadata. Ke každému příspěvku lze přiřadit téměř neomezené množství metadat [8]. Pro uživatelská pole existují různé nadstavby ve formě WordPress pluginů, jako např. Advanced Custom Fields.

### 2.2.6 Meta box

Při úpravách příspěvků jsou k dispozici různé administrační prvky, jako např. box s titulní fotografií (tj. „Featured image“), box s termíny určité taxonomie (pokud je pro daný typ příspěvků aktivovaná) nebo box s možností publikace či uložení daného příspěvku (tj. „Publish“). Tyto boxy se označují jako tzv. „Meta Boxes“. Meta boxy se používají pro seskupování informací k danému příspěvku. Pomocí funkce `add_meta_box()` lze vytvořit nový meta box s vlastním HTML obsahem. Tvorbu meta boxu je vhodné provést ve vlastní funkci zaregistrované pomocí funkce `add_action()` pro hook `add_meta_boxes`. [8]

```
<?php  
  
add_meta_box(  
    string $id,  
    string $title,  
    callable $callback,  
    string|array|WP_Screen $screen = null,  
    string $context = 'advanced',  
    string $priority = 'default',  
    array $callback_args = null  
);  
  
?>
```

Kód 2.7: Tvorba vlastního meta boxu

Parametr `$id` je identifikátor meta boxu. `$title` je jeho titulek (kupříkladu „Featured image“). `$callback` je vlastní funkce s HTML obsahem boxu. `$screen` je typ příspěvků, u kterých se daný box zobrazí. `$context` je pozice boxu (např. v postranním sloupci). `$priority` je priorita daného meta boxu (čím větší, tím je box umístěný výše). `$callback_args` představují data odeslaná vlastní funkci v parametru `$callback`. [10]

### 2.2.7 Shortcode

Shortcode (v překladu „krátký kód“) je makro, které lze vložit do libovolného příspěvku v systému WordPress. Toto makro zastupuje vlastní HTML kód, pomocí kterého je možné interagovat s obsahem daného příspěvku. Po zobrazení příspěvku na webové stránce je místo shortcode vykreslen zastupovaný HTML kód. Systém WordPress ve výchozím nastavení nabízí několik shortcodes, mezi nimi např. `[gallery]` (vykreslí galerii z připojených obrázků). Shortcode může být parametrický, pokud je nutné zastupovanému kódu předat jistá data (např. ID obrázků v případě galerie). Zároveň může být i párový (s počátečním a koncovým tagem), pokud je nutné jím zapouzdřit určitý obsah. Vlastní shortcode lze vytvořit pomocí funkce `add_shortcode()`. [8], [11]

## 2. TEORIE

---

```
<?php  
  
add_shortcode(  
    string $tag,  
    callable $callback  
)  
  
?>
```

Kód 2.8: Tvorba vlastního shortcode

Parametr `$tag` je identifikátorem shortcode. `$callback` je vlastní funkce, která je zavolána v případě, že byl daný shortcode nalezen v textu příspěvku. Tato funkce přijímá 3 parametry: `$atts` s atributy shortcode, `$content` s obsahem shortcode (pokud je párový) a `$shortcode_tag` (totožný s parametrem `$tag`). Návratovou hodnotou této funkce je zastupovaný HTML kód, který se zobrazí místo shortcode. Po registraci shortcode tak stačí do obsahu některého z příspěvků vložit: [<identifikator\_shortcode>]. [8], [10]

### 2.2.8 Best Practices

Při vývoji WordPress pluginů je podle [8] vhodné dodržovat různá architektonická, bezpečnostní a organizační doporučení, která přispívají ke správnému chodu celého systému. V následujícím seznamu jsou uvedené příklady některých z nich:

- Pluginsy sdílejí stejný jmenný prostor pro názvy svých tříd, funkcí a proměnných. Z tohoto důvodu může docházet ke jmenným kolizím. Všechny názvy by měly obsahovat unikátní prefix. To platí i pro identifikátory vlastních příspěvkových typů, taxonomií, meta boxů nebo shortcodes.
- Uživatelé mají v systému WordPress různé role. Těmto rolím jsou přiřazena různá oprávnění. Pomocí funkce `current_user_can()` lze ošetřovat práva daného uživatele a zpřístupňovat mu jen určitý obsah. Důležité části by měly být přístupné pouze oprávněnějším uživatelům.
- Potenciálně nebezpečná vstupní či výstupní data by měla být ošetřována vůči nevhodným znakům (např. HTML, tabulátory či odrádkování). Lze tak předejít škodlivým XSS (tj. Cross-Site Scripting) útokům.
- Hlavní složka s pluginem by měla obsahovat pouze nejdůležitější soubory, jako např. hlavní PHP soubor s hlavičkovým komentářem. Všechny ostatní soubory by měly být uloženy v podsložkách.

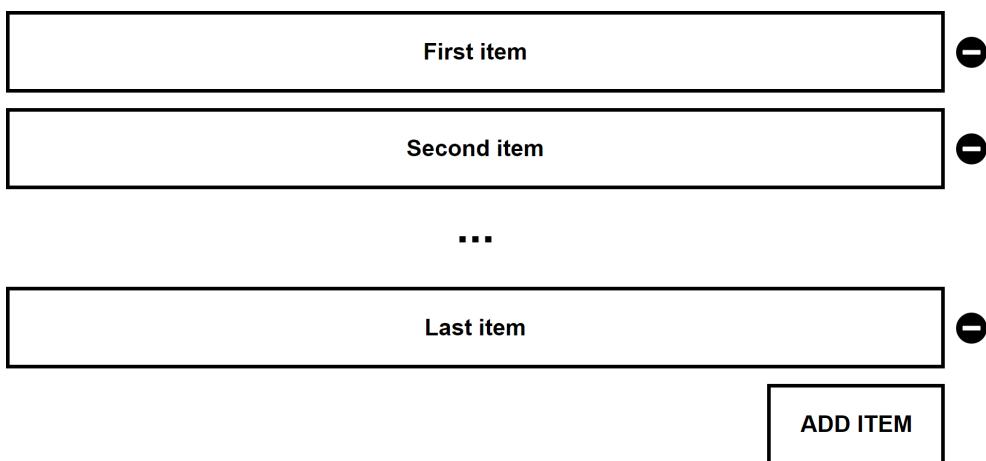
## 2.3 Přehled použitých technologií

Tento přehled slouží pro seznámení čtenáře s veškerými technologiemi, které byly použity při vývoji nového pluginu. Jednotlivé technologie a pojmy se používají v pozdějších kapitolách, a proto je vyžadována alespoň jejich základní znalost v uvedeném rozsahu. Technologie jsou seřazeny podle abecedy.

### 2.3.1 Advanced Custom Fields

Advanced Custom Fields je plugin do systému WordPress, který rozšiřuje tvorbu webového obsahu o nové funkční prvky. Mezi ně patří např. „Text“ (textové pole), „Range“ (rozsah hodnot), „Checkbox“ (výběr položek), „Color Picker“ (výběr barvy), „File“ (soubor), „Gallery“ (galerie), „Link“ (URL odkaz) a další z celkem 30 prvků [14]. Tyto prvky lze seskupovat do tzv. „Field Groups“ a zobrazovat v administraci např. u příspěvků daného typu. Během úprav těchto příspěvků je pak možné vyplnit požadované informace. Jedná se o nadstavbu tzv. „Custom Fields“.

Nejdůležitějším prvkem (pro účely této práce) je tzv. „Repeater Field“ (kolekce položek). Toto uživatelské pole umožňuje uchovávat kolekci libovolného počtu položek, z nichž každá je kombinací uvedených funkčních prvků. Jednotlivé položky lze libovolně přidávat, upravovat, mazat nebo přemisťovat ve stylu „Drag and Drop“. Repeater Field je možné využít např. pro ukládání informací o členech týmu, kdy každá položka kolekce reprezentuje daného člena. U něj lze ukládat např. fotografiu (prvek „Image“), jméno, e-mail nebo telefonní číslo (prvky „Text“).



Obrázek 2.1: Jednoduchý wireframe prvku Repeater Field

## 2. TEORIE

---

### 2.3.2 Babel

Babel je kompilátor pro skriptovací jazyk JavaScript, který umožňuje převádět moderní prvky tohoto jazyka (verze ECMAScript 2015 a vyšších) do podoby srozumitelné současným a starším webovým prohlížečům. Jedná se o nástroj, který posílá daný kód ze vstupu na výstup. Skutečná komplikace je prováděna pomocí jeho pluginů, které dokáží transformovat novější syntaxi jazyka JavaScript. Místo ruční konfigurace je možné volit různé předvolby (tzv. „Presets“). Preset zastupuje sadu pluginů, které jsou použity ve finální komplikaci (např. `@babel/preset-react`, který umožňuje zkompilovat React a syntaxi JavaScript XML). Díky technologii Babel je možné naprogramovat aplikaci pomocí nejmodernějších výdobytků jazyka se zajištěnou kompatibilitou webových prohlížečů. [15]

### 2.3.3 Bootstrap

Bootstrap je frontendový framework pro tvorbu responzivních webových stránek a aplikací. Nabízí šablony pro různé prvky UI (tj. „User Interface“), jako např. „Forms“ (formuláře), „Buttons“ (tlačítka), „Navbar“ (navigační menu), „Modal“ (výstražná okna), „Tooltips“ (nápovedy) a další. Tyto šablony je možné použít ve vlastních projektech, čímž lze zjednodušit a urychlit proces vývoje. Jednou z nejdůležitějších součástí knihovny Bootstrap je tzv. „Grid System“ (mřížka). Mřížka rozděluje prostor na 12 stejně širokých sloupců. Pomocí třídy `col-*` lze libovolnému elementu přiřadit počet sloupců, který udává jeho celkovou šířku. Jako příklad lze uvést tři rovnoměrné elementy v řadě, z nichž každý má nastavenou třídu `col-4`. Třídě `col-*` lze přidat i další hodnotu, která udává hraniční úroveň (tzv. „Breakpoint“). Mezi tyto hodnoty patří např. `sm` (small,  $\geq 576\text{px}$ ), `md` (medium,  $\geq 768\text{px}$ ) nebo `lg` (large,  $\geq 992\text{px}$ ). Pokud je šířka obrazovky např. v rozsahu úrovně `md` a všechny elementy mají třídu `col-md-*`, pak jsou tyto elementy umístěny horizontálně (s definovanou šířkou). Na menších obrazovkách jsou umístěny naopak vertikálně pod sebou a zabírají celou šířku. [16]

### 2.3.4 Composer

Composer je správce závislostí pro jazyk PHP. Umožňuje deklarovat a spravovat závislosti (knihovny nebo balíčky ostatních autorů), na kterých je projekt závislý. Součástí tohoto nástroje je automatické nahrávání tříd spravovaných knihoven (tzv. „Autoloading“). Je však možné nahrávat i třídy vlastní a předejít tak početným `include` či `require` deklaracím na začátcích souborů. Instalace a údržba jednotlivých knihoven probíhá pomocí CLI (tj. Command Line Interface). [17]

### 2.3.5 Gulp

Gulp je nástroj, který umožňuje zjednodušit, automatizovat a urychlit různé procesy při vývoji software. Ať už se jedná o komplikaci JS, minifikaci CSS nebo okamžité promítání změn souborů do prohlížeče. Každý takový proces lze vytvořit jako tzv. „task“. Task je asynchronní JS funkce, jejíž návratovou hodnotou je např. stream. Stream je interpretován jako pipeline – sada samostatně pracujících jednotek (pluginů), které zpracovávají procházející data. Jednotlivé pluginy představují odpovídající NPM (tj. „Node Package Manager“) balíčky ostatních autorů, které je potřeba stáhnout a začlenit do projektu (např. `gulp-cssnano` pro minifikaci CSS souborů nebo `gulp-rename` pro přejmenování souborů). Tímto způsobem je možné vytvářet např. tasks pro komplikaci, sestavení a minifikaci JS nebo SCSS (tj. „Sassy CSS“) souborů. Vytvořené tasks lze jednoduše spouštět pomocí CLI. [18]

### 2.3.6 Node Package Manager

Node Package Manager (zkráceně „NPM“) je správce balíčků a závislostí pro jazyk JavaScript. Jedná se o největší registr balíčků kódu na světě. Používá se pro sdílení programů a užitečných nástrojů mezi vývojáři, kteří je mohou použít ve svých projektech (tím lze potenciálně urychlit a zjednodušit vývoj). Instalace NPM a jednotlivých balíčků (příp. jejich údržba) probíhá pomocí CLI. V současné době NPM nabízí více než 1 milion balíčků. [19]

### 2.3.7 React

React je frontendová JavaScript knihovna pro tvorbu uživatelského rozhraní, která se používá při vývoji webových stránek, aplikací a SPA (tj. „Single-Page Application“). Umožňuje vytvářet nezávislé, uzavřené a znovupoužitelné části kódu – komponenty – ze kterých se skládá celé UI.

Vstupem komponenty mohou být libovolná data. Ta do komponenty přichází ve formě JS objektu (tzv. „Props“). Komponenta může mít také svůj vlastní stav (tzv. „State“). Jedná se o privátní JS objekt s proměnnými daty. S každou změnou stavu se komponenta znova vykreslí. Pro vykreslení slouží metoda `render()`, jejíž návratovou hodnotou je UI dané komponenty. Při definování této metody lze využít potenciálu JSX (tj. „JavaScript XML“). [20]

JSX je syntaktické rozšíření jazyka JavaScript, které slouží ke snazšímu popisu UI. Kompilací JSX jsou vytvořeny React elementy (tj. JS objekty, které reprezentují skutečné HTML elementy). Ty jsou následně použity při sestavení DOM (tj. „Document Object Model“). [20]

Pomocí komponent je popsán vzhled celého UI. React vytváří VDOM (tj. „Virtual Document Object Model“), pomocí kterého je kompletní reprezentace UI udržována v paměti. Při provádění změn je VDOM synchronizován s DOM. React aktualizuje jen takové elementy, u kterých nastaly změny. [20]

## 2. TEORIE

---

Komponenty mohou definovat i další metody, které se uplatní v průběhu jejich životního cyklu. Mezi ně patří např. `componentDidMount()` (zavolána po vložení komponenty do DOM), `componentDidUpdate()` (zavolána po aktualizaci komponenty v DOM) nebo `componentWillUnmount()` (zavolána před odstraněním komponenty z DOM). [20]

### 2.3.8 Sass

Kaskádové styly (zkráceně „CSS“) se používají k definování vzhledu HTML elementů. Používání CSS však vede k rozsáhlému, nepřehlednému a těžce udržovatelnému kódu. Řešením tohoto problému je použití CSS preprocesoru, který zápis kaskádových stylů výrazně zjednoduší a zkrátí. Sass (tj. „Syntactically Awesome Style Sheets“) je preprocesor kaskádových stylů, které rozšiřuje o nové funkcionality, jako např. proměnné, vnořování, dědění nebo operátory. Pomáhá udržovat princip DRY (tj. „Don’t Repeat Yourself“), který spočívá v odstraňování duplicitních částí kódu. Vylepšený zápis je překládán do obyčejných CSS, které jsou čitelné běžnými webovými prohlížeči. Soubory s CSS napsané pomocí jazyka Sass mají příponu SCSS. [21]

### 2.3.9 Swiper

Slider (také „Slideshow“ nebo „Carousel“) je označení pro prezentaci snímků (např. obrázků) na webové stránce, ve které se jednotlivé snímky střídají. Swiper je technologie, zaměřená především na mobilní zařízení, která umožňuje tvorbu těchto interaktivních prvků v různých podáních. S její pomocí lze vytvářet např. horizontální či vertikální prezentace, volit různé efekty přechodů, přidat posuvné šipky či navigaci, aktivovat automatické přehrávání (tzv. „Autoplay“) nebo efekt pomalého načítání obrázků (tzv. „Lazy Load“). Díky rozsáhlému API (tj. „Application Programming Interface“) lze vytvořit vlastní prezentaci a upravit ji dle vlastních potřeb. Swiper je možné používat i v kombinaci s dalšími moderními technologiemi, jako např. React. [22]

# KAPITOLA 3

---

## Analýza

V této kapitole je provedena analýza zadavatele a původního stavu rádiových stanic na jeho webu (z vizuálního a implementačního hlediska). Následuje popis požadovaného stavu, ve kterém jsou uvedeny všechny funkční a nefunkční požadavky včetně případů užití a doprovodných diagramů. V následující kapitole je pak na základě analýzy navrhнуто vlastní řešení.

### 3.1 Zadavatel

Czech-American TV (dále jen „CATV USA“) je americká televizní stanice a nezisková organizace, která byla založena v roce 2003. Její sídlo se nachází v Cape Coral, Florida, USA. Zakladatelem a současným ředitelem CATV USA je John Honner, americký producent českého původu. [23]



Obrázek 3.1: Logo CATV USA [24]

### 3. ANALÝZA

Hlavním cílem této organizace je rozšířit povědomí o České republice a její kultuře, tradicích, regionech, historii, památkách, hudbě a tanci nejen ve Spojených státech amerických, ale i v dalších zemích celého světa. Cílí na všechny diváky, kteří rádi cestují, poznávají nové věci a chtějí se dozvědět o České republice něco více. [23]

CATV USA pravidelně vydává pořad „Czech-American TV Show (produced by John Honner)“ s naučnou a vzdělávací tematikou, který přibližuje Českou republiku divákům. Tento pořad je následně dodán kabelové televizi, která ho odvysílá v průběhu hlavního vysílacího času. Pořad se zaměřuje na české regiony, významné turistické destinace, hrady, zámky, muzea, českou kuchyni a recepty, tradice, folklorní hudbu a také na lekce češtiny. Jednotlivé díly jsou po odvysílání umístěny na webové stránky CATV USA, kde je mohou zhlédnout diváci z celého světa. [23]

Správa vlastních webových stránek patří mezi další činnosti této organizace. Webové stránky se nachází na adrese [www.catvusa.com](http://www.catvusa.com) a kromě již zmíněných dílů nabízí i recepty české kuchyně (v psané či video formě), výuku českého jazyka, články o českých tradicích, vyhledávání předků pro Američany s českými kořeny, fakta o České republice, dvě internetová rádia s českou folklorní a vážnou hudbou a mnoho dalších zajímavých věcí. [23]



Obrázek 3.2: Webové stránky CATV USA

Na fungování televizní stanice a jejím vylepšování se dobrovolně podílí studenti a univerzitní pracovníci z několika předních českých univerzit (v rámci svých závěrečných prací). Dále střední školy, orgány české státní správy, kraje,

## 3.2. Původní stav

města, destinační společnosti, česká média a další. Mezi obory dobrovolníků patří nejen informatika, ale také např. marketing, design, hudba či jazyky. [25]

CATV USA patří mezi první organizace, které se začaly věnovat internetovému vysílání a streamingu. Vzhledem k tomu, že se jedná o první a jedinečný projekt svého druhu, má obrovský potenciál i do budoucnosti. [25]

### 3.2 Původní stav

CATV USA měla na svých webových stránkách dvě internetová rádia s českou folklorní a vážnou hudební – „Radio Folk“ a „Radio Classic“. Tato rádia automaticky přehrávala předem připravené playlisty za pomoci přehrávače Plyr. Playlisty byly složené z různých typů nahrávek – např. vstupy moderátorů, písň nebo znělky televize CATV USA. Všechny byly uloženy a manuálně spravovány na FTP (tj. „File Transfer Protocol“) serveru. Přehrávač Plyr byl již zabudovaný na webu. Záměrem bylo, aby posluchač vůbec nepoznal, že poslouchá automatizovaná rádia. Veškeré nahrávky byly předem připravené. Rádia demonstrovala a napodobovala živé vysílání. [26]

#### 3.2.1 Rádio Folk



Obrázek 3.3: Ukázka rádia Folk

### 3. ANALÝZA

---

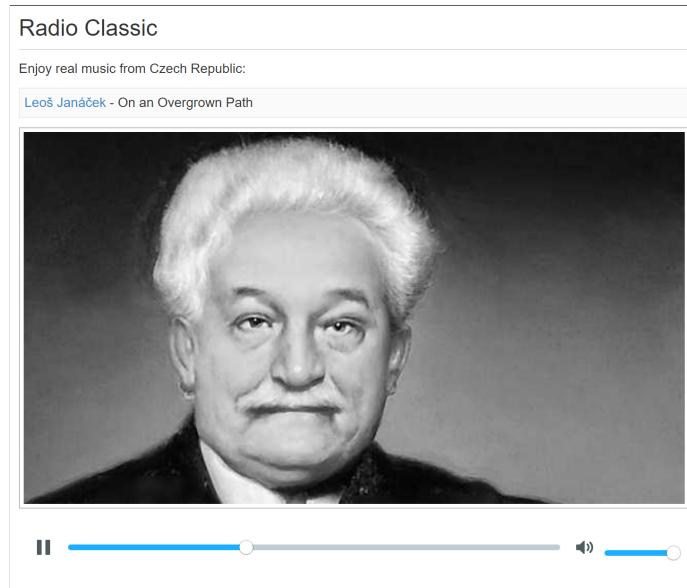
Rádio Folk přehrávalo české folklorní písni z playlistu, který obsahoval následující soubory:

1. přibližně deset krátkých nahrávek, na kterých bylo posluchači představeno rádio (tzv. znělky CATV USA se vstupem moderátora, např. „You are listening to the Czech-American TV radio programme!“),
2. přibližně dvacet krátkých nahrávek, na kterých byla posluchači představena hudební skupina (např. „You will now listen to the Czech brass band Sobuláci!“),
3. přibližně tisíc písni od přibližně dvaceti hudebních skupin. [26]

Soubory byly v playlistu uloženy následovně: jedna nahrávka z bodu 1, jedna nahrávka z bodu 2 a tři písni (od stejné hudební skupiny) z bodu 3. Tento vzor se neustále opakoval. Výběr znělek, hudebních skupin a písni byl při každém spuštění přehrávače náhodný a výlučný (tzn. že náhodně vybíral znělky, skupiny a písni, dokud nebyly přehrány úplně všechny). Tím bylo vyloučeno opakování (tzn. že posluchač nikdy neslyšel dvě stejné znělky, skupiny či písni po sobě). [26]

Nad přehrávačem bylo zobrazeno logo rádiové stanice a textové informace o hudební skupině, jejíž písni se právě přehrávaly. Uváděl se název skupiny, její webové stránky a telefonní číslo. [26]

#### 3.2.2 Rádio Classic



Obrázek 3.4: Ukázka rádia Classic

Rádio Classic přehrávalo českou vážnou hudbu z playlistu, který fungoval podobně jako playlist rádia Folk. Posluchači bylo nejdříve představeno rádio (formou znělky CATV USA), následovaly tři skladby od různých hudebních skladatelů (ti se již nepředstavovali) a takto stále dokola. Výběr znělek, hudebních skladatelů a jejich skladeb byl opět náhodný a výlučný (tzv. shuffle mód přehrávání). Narozdíl od rádia Folk se s každým přehrávaným skladatelem zobrazovala i jeho fotografie. Mezi textovými informacemi bylo jméno hudebního skladatele a název právě přehrávané skladby (v podobě odkazů vedoucích dále na server Wikipedia). [26]

#### 3.2.3 Výstražná hlášení

Obě rádia používala WordPress plugin s2Member Framework. Tento plugin primárně zpřístupňoval vybrané webové příspěvky pouze těm uživatelům, kteří si za malý poplatek pořídili členství. Tito uživatelé následně získali vlastní účet, přes který měli přístup k celému webovému obsahu. Ostatním uživatelům se vybraný obsah nezobrazoval. [26]

Neregistrovanému či nepřihlášenému posluchači se po přehrání několika písni (konkrétně tří u obou rádií, 15–20 minut) přehrávání zastavilo. Zobrazila se první výstraha, která požadovala, aby si posluchač pořídil zmíněné členství. K dispozici byl i odkaz na formulář, kde se členství dalo vyřídit. Výstrahu šlo snadno obejít, ale poslech byl i nadále pravidelně přerušován tímto výstražným hlášením. Registrovanému a přihlášenému posluchači se první výstraha nezobrazovala, a ten tak mohl poslouchat rádio bez přerušení. [26]

Po delší době (1–2 hodiny) se zobrazovala druhá výstraha, která pouze kontrolovala, zdali posluchač stále poslouchal. Jednalo se o prevenci proti zbytečnému streamingu a zatížení serveru. [26]

#### 3.2.4 Zjištěné chyby

V předešlých podkapitolách byly popsány jednotlivé rádiové stanice a výstražná hlášení. Jejich analýza proběhla v úzké spolupráci se zadavatelem. Po podrobném zkoumání se však ukázalo, že skutečný stav těchto rádií neodpovídá stavu, jak jej prezentoval zadavatel. Přehrávání rádiových stanic totiž doprovázela řada různých chyb:

1. Po přeskočení první výstrahy se celá webová stránka s rádiem aktualizovala a vytvořil se nový playlist. Z implementace později vyplynulo, že jednotliví hudebníci byli do playlistu vybíráni sice náhodně, ale bez ohledu na výlučnost. Například u rádia Folk se hudebníci vybírali podle složek na FTP serveru, které obsahovaly jejich písni (co složka, to hudebník). Složky však byly interpretovány jako CD/DVD jednotlivých hudebníků. Tím pádem mohlo existovat i více složek jednoho hudebníka. Navíc tyto složky občas obsahovaly i stejně písni. Hudebníci a písni se

### 3. ANALÝZA

---

tedy mohli opakovat. U rádia Classic po každých 5–10 skladbách nastala situace, kdy byl vybrán stejný hudební skladatel dvakrát po sobě.

2. U obou rádií se přibližně po hodině poslechu samo od sebe zastavilo přehrávání. Rádiová stanice následně nedokázala načítat ani přehrávat další písň, dokud se neaktualizovala celá webová stránka.
3. Druhá výstraha se vůbec nezobrazovala. V kódu nebyla nalezena její implementace. Pokud uživatel neinteragoval se stránkou s rádiovou stanicí, pak se nezobrazovala ani první výstraha. Jednalo se o případy, kdy v jedné záložce prohlížeče byla otevřena stránka s přehrávanou rádiovou stanicí a uživatel pracoval v jiné záložce (nebo mimo prohlížeč). Jinými slovy, pro korektní zobrazení prvního výstražného hlášení se uživatel musel na stránku s rádiovou stanicí dívat.
4. Nebylo chybou, že přehrávanou písň šlo jednoduše přeskočit. Stačilo přehrávání přesunout těsně před její konec na ukazateli průběhu. Písň následně dohrála až do konce. V ten moment se však přehrávání zastavilo. Po spuštění se začala přehrávat ta samá písň od začátku. Přítomnost ukazatele průběhu navíc kazila dojem posluchače, že doopravdy poslouchá živé rádio.
5. Po otevření webové stránky s rádiovou stanicí se přehrávání spustilo automaticky. Pokud se však stránka s daným rádiem aktualizovala (např. pomocí klávesy F5), rádio bylo pozastavené a žádné automatické spuštění nenastalo.
6. Daná webová stránka musela obsahovat pouze jednu rádiovou stanici. Pokud obsahovala např. rádio Classic a rádio Folk zároveň, pak se v obou rádiových stanicích přehrával playlist rádia Folk (vždy podle rádia vloženého naposledy).

Za způsobené chyby mohla především špatná implementace, která byla v minulosti prováděna již několikrát [26]. Z tohoto důvodu se v kódu nacházelo spoustu nepotřebných a zbytečných funkcí, proměnných a příkazů. Veškerá funkcionality ohledně rádií byla implementována v šabloně webu v následujících souborech:

- `radio.php` – hlavní PHP skript, který prováděl tvorbu playlistu v závislosti na vybrané rádiové stanici. Definoval UI a vlastní shortcode (pomocí něho se jednotlivá rádia zobrazovala na webu).
- `player.js` – novější zkompilovaný JS balíček, který zodpovídal za hlavní logiku přehrávače, jako např. procházení playlistu, přehrávání písni nebo promítání obrázků či informací o hudebnících).

### 3.3. Požadovaný stav

- **style.css** – hlavní kaskádové styly webu, jejichž součástí byla i pravidla stylů pro UI rádiových stanic.

Dalším problémem byl až příliš imperativní a nerozšířitelný kód (zejména v souboru **radio.php**). Playlisty rádiových stanic se rozlošovaly v **if() else{}** konstrukci dle názvu daného rádia. K jednotlivým souborům (písně, znělky, obrázky či informace o hudebnících) uloženým na FTP serveru se přistupovalo skrze jejich exaktní názvy. Tyto soubory musely být specificky pojmenované (např. symfonie č. 9 „Z Nového světa“ od Antonína Dvořáka se musela jmenovat **dvo\_new.mp3** a soubor s informacemi o skladateli **popis.txt**, protože to tak vyžadoval algoritmus). Přidávání dalších rádií, playlistů, hudebníků, písní nebo obrázků tedy bylo velice obtížné. Potenciálu objektově orientovaného programování (dále jen „OOP“) bylo využito pouze v souboru **player.js**. Z pohledu administrátora však bylo velice těžké upravovat výstražná hlášení, poněvadž už byla zkompilovaná právě v tomto souboru.

## 3.3 Požadovaný stav

Organizace CATV USA v současné době připravuje nový webový design. Chybě fungující rádiové stanice vestavěné v současně vzhledové šabloně tak představují velký problém. Proto je vyžadována změna aktuálního stavu. Optimálním řešením je realizace nového nezávislého WordPress pluginu, který umožní vytvářet automatizované rádiové stanice a přidá jednoduchou správu pro jejich řízení. Veškeré podrobnosti o požadovaných funkcionalitych jsou uvedeny v následujících podkapitolách.

### 3.3.1 Funkční požadavky

Požadavky se dělí na dvě kategorie: frontend (webová část) a backend (administrační část). Všechny požadavky jsou později namapovány na případy užití, které jejich obsah více specifikují.



Obrázek 3.5: Přehled funkčních požadavků

### **3. ANALÝZA**

---

**Zkratka:** F1

**Název:** Zobrazování rádiových stanic

**Kategorie:** Frontend

**Popis:** Uživatel může navštívit webovou stránku, na které se nachází předem připravená rádiová stanice. Rádio se skládá z názvu, multimédií a přehrávače. Mezi multimédia patří videa, obrázky či textový popis hudebníka, jehož píseň se přehrává. Pokud hudebníkovi uvedená multimédia chybí, pak se místo nich zobrazuje logo rádia, příp. vybrané příspěvky z webu. Přehrávač umožňuje spustit či zastavit přehrávání. Zobrazuje jméno hudebníka a název písni.

**Zkratka:** F2

**Název:** Zobrazování výstražných hlášení

**Kategorie:** Frontend

**Popis:** Uživateli se při poslechu pravidelně zobrazují výstražná hlášení. Tato hlášení zastavují přehrávání a vyžadují od uživatele provedení určité akce. První výstraha uživatele žádá, aby si za malý poplatek pořídil členství, které mu přinese spoustu výhod (např. zpřístupnění kompletního webového obsahu nebo nepřerušovaný poslech). Druhá výstraha pouze kontroluje, jestli uživatel stále poslouchá. Obě výstražná hlášení lze přeskočit.

**Zkratka:** F3

**Název:** Správa hudebníků

**Kategorie:** Backend

**Popis:** Uživatel může upravovat všechny hudebníky pomocí jednoduchého rozhraní v administraci systému WordPress. U každého hudebníka lze upravit jméno, textový popis, obrázky, vstupy moderátorů a písni. Hudebníkovi lze také nastavit odpovídající hudební žánr. Vstupy moderátorů se přehrávají vždy před písni daného hudebníka.

**Zkratka:** F4

**Název:** Správa rádiových stanic

**Kategorie:** Backend

**Popis:** Uživatel může upravovat všechny rádiové stanice pomocí jednoduchého rozhraní v administraci systému WordPress. U každého rádia lze upravit název, logo, webové příspěvky, playlist a výstražná hlášení. Playlist se skládá z vybraných hudebníků. Rádiová stanice následně tyto hudebníky prochází a přehrává jejich písni, příp. vstupy moderátorů.

### 3.3.2 Nefunkční požadavky



Obrázek 3.6: Přehled nefunkčních požadavků

**Zkratka:** N1

**Název:** Automatizace

**Popis:** Rádiové stanice pracují samostatně a v nepřetržitém provozu. Používají předem připravená data uložená ve správě (např. playlist rádia, hudebníky a jejich písničky, vstupy moderátorů atp.). K jejich běžnému provozu není potřebná aktivní účast osob (např. moderátorů).

**Zkratka:** N2

**Název:** Dostupnost

**Popis:** Plugin lze použít jen na webové stránce spravované v systému WordPress. Rádiové stanice fungují spolehlivě na různých zařízeních (stolní počítač, notebook, tablet, mobilní telefon) a jsou kompatibilní s různými prohlížeči (Chrome, Firefox, Opera, Safari, Edge).

**Zkratka:** N3

**Název:** Použitelnost

**Popis:** Úpravy hudebníků a rádiových stanic jsou jednoduché činnosti. Zvládnou je i netechničtí uživatelé, kteří nerozumí programování.

**Zkratka:** N4

**Název:** Jazyková lokalizace

**Popis:** Veškeré textové výstupy pluginu jsou přeložené do anglického jazyka a zkонтrolované rodilým mluvčím.

**Zkratka:** N5

**Název:** Architektura

**Popis:** Plugin má jednoduchou modulární architekturu, která je snadno udržovatelná a potenciálně rozšířitelná i do budoucnosti. Řídí se dle Best Practices (pro vývoj pluginů do systému WordPress), která zahrnují různá architektonická, bezpečnostní a organizační doporučení.

**Zkratka:** N6

**Název:** Migrace

**Popis:** Plugin je přizpůsobený pro budoucí potenciální migraci dat webu CATV USA na moderní cloudové úložiště.

### 3. ANALÝZA

---

#### 3.3.3 Aktéři

V této podkapitole jsou popsáni všichni aktéři, kteří dále vystupují v jednotlivých případech užití.

Posluchač je jakákoli osoba, která má přístup k webovým stránkám CATV USA. Prostřednictvím rádiových stanic poslouchá českou folklorní a vážnou hudbu. Může přistupovat pouze k frontendové části pluginu.

Editor je jakákoli osoba, která má přístup k administraci webu CATV USA. Vlastní potřebná práva pro úpravu jednotlivých hudebníků. Může přistupovat k backendové i frontendové části pluginu.

Administrátor je jakákoli osoba, která má přístup k administraci webu CATV USA. Vlastní potřebná práva pro úpravu jednotlivých rádiových stanic a hudebníků. Může přistupovat k backendové i frontendové části pluginu.

#### 3.3.4 Případy užití

Případy užití jsou namapovány na funkční požadavky následujícím způsobem:

| -         | Případy užití |     |     |     |     |     |     |     |     |      |      |      |
|-----------|---------------|-----|-----|-----|-----|-----|-----|-----|-----|------|------|------|
| Požadavky | UC1           | UC2 | UC3 | UC4 | UC5 | UC6 | UC7 | UC8 | UC9 | UC10 | UC11 | UC12 |
| F1        | +             | +   | +   |     |     |     |     |     |     |      |      |      |
| F2        |               |     |     | +   |     |     |     |     |     |      |      |      |
| F3        |               |     |     |     | +   | +   | +   | +   |     |      |      |      |
| F4        |               |     |     |     |     |     |     |     | +   | +    | +    | +    |

Obrázek 3.7: Mapování požadavků na případy užití

|   |                 |   |                 |
|---|-----------------|---|-----------------|
| <b>UC1</b><br>Získat informace o hudebníkovi    | <b>FRONTEND</b> | <b>UC2</b><br>Přehrávat rádiovou stanici              | <b>FRONTEND</b> |
| <b>UC3</b><br>Zobrazit stránku s rádiem         | <b>FRONTEND</b> | <b>UC4</b><br>Výřídit výstražné hlášení               | <b>FRONTEND</b> |
| <b>UC5</b><br>Spravovat informace o hudebníkovi | <b>BACKEND</b>  | <b>UC6</b><br>Spravovat obrázky hudebníka             | <b>BACKEND</b>  |
| <b>UC7</b><br>Spravovat nahrávky hudebníka      | <b>BACKEND</b>  | <b>UC8</b><br>Zobrazit stránku s nastavením hudebníka | <b>BACKEND</b>  |
| <b>UC9</b><br>Spravovat informace o rádiu       | <b>BACKEND</b>  | <b>UC10</b><br>Spravovat výstražná hlášení rádia      | <b>BACKEND</b>  |
| <b>UC11</b><br>Spravovat playlist rádia         | <b>BACKEND</b>  | <b>UC12</b><br>Zobrazit stránku s nastavením rádia    | <b>BACKEND</b>  |

Obrázek 3.8: Přehled případů užití

**Zkratka:** UC1**Název:** Získat informace o hudebníkovi**Kategorie:** Frontend

**Popis:** V přehrávači rádiové stanice se zobrazuje jméno hudebníka (např. „Bedrich Smetana“) a název skladby (např. „Vltava“), která se právě přehrává. Posluchač si během přehrávání může přečíst textový popis daného hudebníka, ve kterém se nachází např. informace o jeho životě nebo kariéře. Zároveň si může prohlížet obrázky, na kterých je daný hudebník vyobrazen. Při změně hudebníka nebo skladby se uvedené informace aktualizují.

**Zkratka:** UC2**Název:** Přehrát rádiovou stanici**Kategorie:** Frontend

**Popis:** Posluchač může spustit či zastavit přehrávání rádiové stanice pomocí k tomu určených ikonek (play/pause) v přehrávači. Rádiová stanice přehrává audio/video nahrávky hudebníků (např. písničky, reklamy, znělky). Během přehrávání zobrazuje obrázky a textové popisy hudebníků. Pokud obrázky a popisy hudebníkům chybí, může administrátor nastavit promítání vybraných webových příspěvků. Jedná se o jakékoli existující příspěvky z webu (např. české tradice). V takovém případě je zobrazena titulní fotografie a obsah daného příspěvku. Se změnou písničky se zobrazí další webový příspěvek.

**Zkratka:** UC3**Název:** Zobrazit stránku s rádiem**Kategorie:** Frontend

**Popis:** Pro poslech rádiové stanice musí posluchač nejprve přistoupit k předem připravené stránce s daným rádiem. Tato stránka obsahuje úvodní text, rádiovou stanici, příp. technické informace.

**Zkratka:** UC4**Název:** Vyřídit výstražné hlášení**Kategorie:** Frontend

**Popis:** Po přehrání určitého počtu hudebníků se přehrávání rádiové stanice zastaví. Zobrazí se výstraha, která požaduje, aby si posluchač pořídil členství. Ve výstraze je odkaz, který posluchače přesměruje na stránku s registračním formulářem. Výstrahu lze přeskočit a pokračovat v poslechu. Pokud se však posluchač nezaregistrouje nebo nepřihlásí, výstraha bude i nadále pravidelně přerušovat přehrávání. Registrovaní a přihlášení posluchači mohou rádio poslouchat bez přerušení. Při přehrávání se zobrazuje i další výstraha, která pouze kontroluje, jestli posluchač stále poslouchá.

### **3. ANALÝZA**

---

**Zkratka:** UC5

**Název:** Spravovat informace o hudebníkovi

**Kategorie:** Backend

**Popis:** Administrátor (příp. editor) může upravovat veškeré informace o daném hudebníkovi (tj. jméno a textový popis). Pro úpravy popisu má k dispozici vestavěný WYSIWYG editor. Hudebníka lze přiřadit k hudebnímu žánru, ke kterému muzikálně náleží. Žánry (např. klasická nebo lidová hudba) budou použity v budoucnosti pro tvorbu různých archivů na webovou stránku (např. archiv českých skladatelů).

**Zkratka:** UC6

**Název:** Spravovat obrázky hudebníka

**Kategorie:** Backend

**Popis:** Administrátor (příp. editor) může přidávat, upravovat či mazat obrázky daného hudebníka. Ty se postupně zobrazují v průběhu přehrávání jeho písni. Výběr konkrétního obrázku je náhodný a výlučný (shuffle mód).

**Zkratka:** UC7

**Název:** Spravovat nahrávky hudebníka

**Kategorie:** Backend

**Popis:** Administrátor (příp. editor) může přidávat, upravovat či mazat nahrávky daného hudebníka (např. písně, videa, znělky nebo reklamní oznámení). Taktéž může přidávat, upravovat či mazat úvodní nahrávky (např. „You will now listen to the Czech brass band Vacenovjáci!“). Když na hudebníka dojde řada v playlistu, přehraje se jedna úvodní nahrávka (např. vstup moderátora) následovaná libovolným počtem nahrávek (např. písni). Výběr konkrétní úvodní nahrávky a nahrávky je náhodný a výlučný (shuffle mód).

**Zkratka:** UC8

**Název:** Zobrazit stránku s nastavením hudebníka

**Kategorie:** Backend

**Popis:** Pro úpravu informací, obrázků a nahrávek daného hudebníka musí administrátor (příp. editor) nejprve přistoupit ke stránce s nastavením tohoto hudebníka. Ta se nachází v administraci systému WordPress společně se stránkami ostatních hudebníků. Na stejném místě lze jednotlivé hudebníky přidávat a mazat.

**Zkratka:** UC9

**Název:** Spravovat informace o rádiu

**Kategorie:** Backend

**Popis:** Administrátor může upravovat veškeré informace o dané rádiové stanici (tj. název a logo). Logo rádia se zobrazuje jen tehdy, když hudebník nemá žádné obrázky ani popis a zároveň nejsou promítány žádné webové příspěvky. Pro každou rádiovou stanici je možné nastavit dobu zobrazení obrázků přehrávaných hudebníků (v sekundách). Dále lze vybrat konkrétní webové příspěvky,

### 3.3. Požadovaný stav

které se v rádiu promítají namísto obrázků a popisů hudebníků. Promítání webových příspěvků je náhodné a výlučné (shuffle mód).

**Zkratka:** UC10

**Název:** Spravovat výstražná hlášení rádia

**Kategorie:** Backend

**Popis:** Administrátor může přidávat, upravovat či mazat výstražná hlášení dané rádiové stanice. U každého lze upravit nadpis, obsah, cílovou uživatelskou skupinu a počet hudebníků, po kterých se dané hlášení zobrazí. Součástí každé výstrahy je tlačítko pro její potvrzení. S tímto tlačítkem může být propojena nastavitelná URL adresa, na kterou je posluchač po kliknutí přesměrován.

**Zkratka:** UC11

**Název:** Spravovat playlist rádia

**Kategorie:** Backend

**Popis:** Administrátor může přidávat, upravovat či mazat položky v playlistu dané rádiové stanice. Položka playlistu reprezentuje skupinu hudebníků. Pro každou položku lze nastavit konkrétní hudebníky, počet přehrávaných hudebníků a počet přehrávaných nahrávek od každého z nich. Výběr hudebníků je náhodný a výlučný (shuffle mód). Rádiová stanice prochází playlist cyklicky ve směru shora dolů.

**Zkratka:** UC12

**Název:** Zobrazit stránku s nastavením rádia

**Kategorie:** Backend

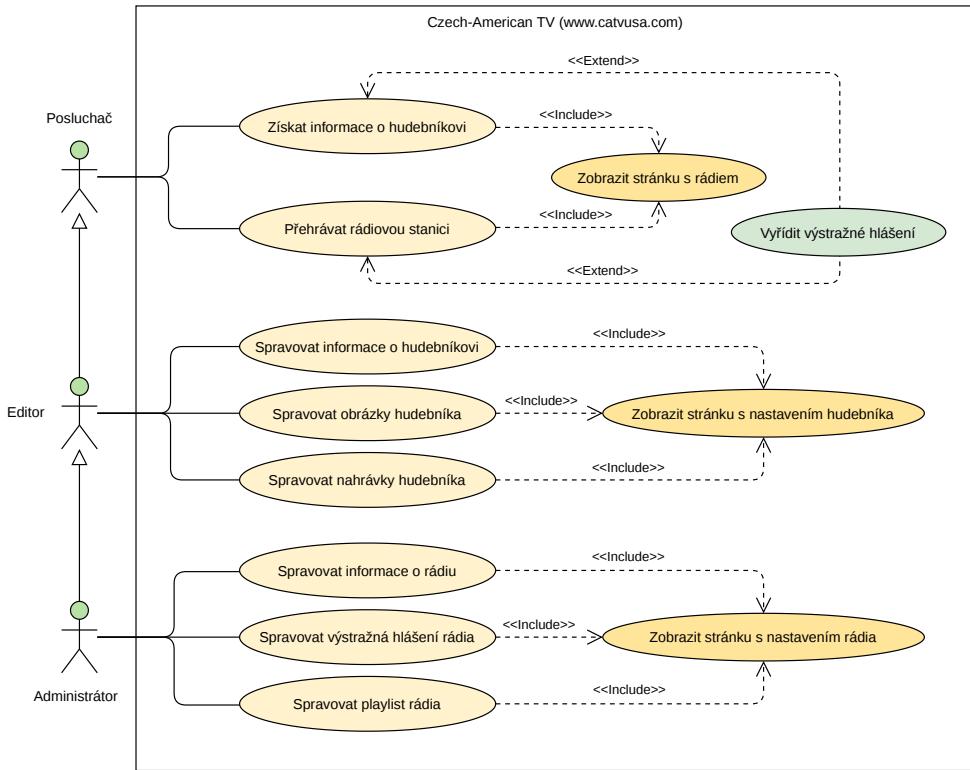
**Popis:** Pro úpravu informací, výstražných hlášení a playlistu dané rádiové stanice musí administrátor nejprve přistoupit ke stránce s nastavením tohoto rádia. Ta se nachází v administraci systému WordPress společně se stránkami ostatních rádií. Na stejném místě lze jednotlivá rádia přidávat a mazat.

#### **3.3.5 Diagram případů užití**

Následující diagram zobrazuje jednotlivé aktéry a jejich oprávnění v rámci webové stránky CATV USA. Každý z aktérů je asociován se specifickými případy užití. Oprávněnější aktéři dělí případy užití méně oprávnějších aktérů. U všech případů je nejdříve nutné zobrazit specifickou webovou stránku (s rádiovou stanicí nebo s nastavením hudebníka či rádia). Používání rádia navíc může být rozšířeno o případ vyřízení výstražných hlášení.

### 3. ANALÝZA

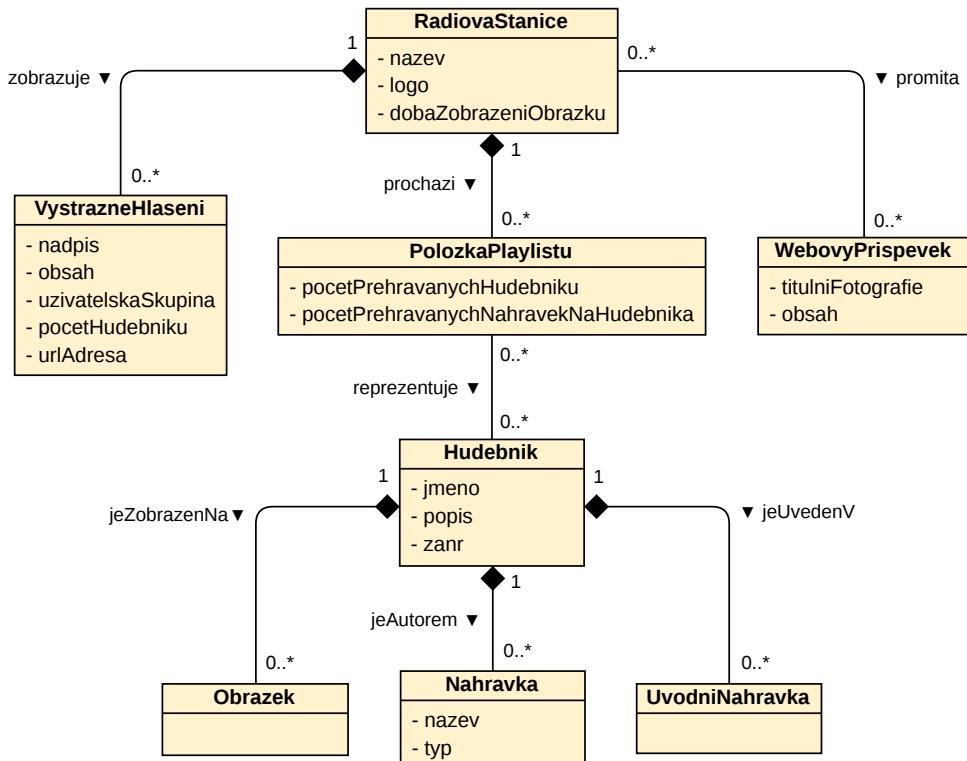
---



Obrázek 3.9: Diagram případů užití

#### 3.3.6 Doménový model

Následující diagram zachycuje entity problémové domény a vztahy mezi nimi. Slouží k jejich objasnění a lepšímu pochopení. Důležité je zmínit, že položky playlistu rádia jsou interpretovány jako soubory hudebníků, nikoliv jako nahrávky. Při přehrávání se z tohoto souboru vybere určitý počet hudebníků. Od každého z nich je následně přehrán určitý počet nahrávek. Z tohoto důvodu neexistuje přímá vazba mezi položkou playlistu a nahrávkou. Jedná se o nové pojetí playlistu, které bude více vysvětleno a rozšířeno v příští kapitole. V diagramu jsou v určitých případech použity kompoziční vztahy. Výstražná hlášení a položky playlistu jsou nedílné součásti rádiové stanice. Podobně obrázky, nahrávky ani úvodní nahrávky (s vazbou na konkrétního hudebníka) nemají bez hudebníka smysl.



Obrázek 3.10: Analytický doménový model

### 3.3.7 Podobná řešení

V této kapitole byly popsány a rozebrány veškeré požadavky na nový plugin do systému WordPress. Nastává otázka, zdali již nebyl vytvořen jiný WordPress plugin, který by uvedenou problematiku uspokojivě řešil.

Organizace CATV USA by si tak mohla tento plugin jednoduše dohledat v knihovně pluginů, která je přístupná přímo v administraci. Po instalaci pluginu a dalších administračních úkonech by byly zprovozněny nové rádiové stanice a dosažen požadovaný stav.

V současné době bohužel neexistuje žádný WordPress plugin, který by řešil potřeby a požadavky organizace CATV USA uvedené v této kapitole. Mezi potenciální pluginy patří *Radio Station*, *WP Radio Player* a *StreamCast*. Tyto pluginy se však zaměřují na vysílání již existujících živých rádií. Obsah vysílání nelze přímo vytvářet. Uvedené pluginy se tudíž neshodují s požadovaným stavem, a proto je v následující kapitole navrhnuto vlastní řešení.



# KAPITOLA 4

---

## Návrh

Na základě analýzy původního a požadovaného stavu bylo navrhnuto řešení v podobě nového pluginu do systému WordPress. V této kapitole je uveden popis a podrobnosti tohoto řešení. Výklad je doplněn o fotografie finálního prototypu a pomocné wireframes. V další kapitole je pak na základě návrhu popsána implementace nového pluginu.

### 4.1 Radio Manager

Nový plugin do systému WordPress se jmenuje *Radio Manager*. Představuje manažera, který spravuje a řídí všechna internetová rádia na webové stránce. Samotný plugin je rozdělen do tří částí – frontend (část viditelná z webové stránky), backend (část viditelná z administrace systému WordPress) a propojení těchto dvou částí (způsob, jakým lze rádiové stanice vkládat na web). Na webové stránce jsou rádiové stanice prezentovány koncovým posluchačům, zatímco v administraci systému WordPress jsou jednotlivá rádia spravována administrátory (příp. editory). Všechny uvedené části jsou podrobně popsány v následujících podkapitolách.

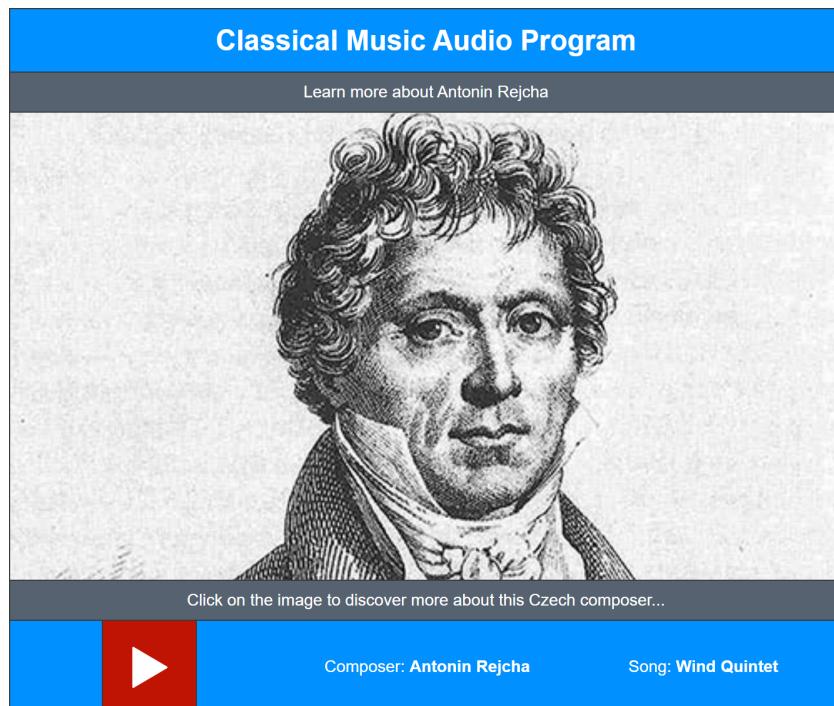
Návrh všech částí byl orientován na specifickou uživatelskou skupinu. Tato skupina zahrnuje anglicky mluvící uživatele z USA, kteří nerozumí technice ani programování. Někteří z nich znají systém WordPress jen na elementární úrovni (např. tvorba a editace příspěvků). Vyžadují jednoduchost, přehlednost, příp. adekvátní uživatelskou podporu. Důležité jsou pro ně zvyky, tradice a zavedené postupy. Z tohoto důvodu se neradi učí novým věcem, které potřebují pochopit a vysvětlit. Uživatelé z této skupiny byli vybráni pro testování, o kterém se lze dočíst více později v této práci (kap. 6).

## 4.2 Frontend

Výstupem pluginu na webové stránce je responzivní přehrávač, který lze vložit do jakéhokoli příspěvku vytvořeného v systému WordPress. Tento přehrávač používá data konkrétní rádiové stanice ze správy. V této podkapitole je popsán jeho vzhled a jednotlivé součásti. Tvorba rádiových stanic je popsána v podkapitole 4.3. Způsob, jakým lze přehrávač daného rádia vložit do konkrétního příspěvku, je popsán v podkapitole 4.4.

### 4.2.1 Přehrávač

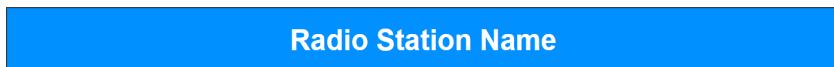
Přehrávač se skládá z názvu rádiové stanice, multimédií a ovládání. Použité barvy (světle modrá, tmavě červená, tmavě šedá a bílá) byly vybrány záměrně. Tyto barvy jsou součástí loga zadavatele a současné vzhledové šablony jeho webových stránek. Následující ukázky zobrazují finální prototyp přehrávače, který byl vytvořen a otestován v nástroji Axure RP.



Obrázek 4.1: Ukázka rádia Classic

#### 4.2.1.1 Název rádiové stanice

První částí přehrávače je hlavička, ve které se nachází název rádia.



Obrázek 4.2: První část přehrávače

#### 4.2.1.2 Multimédia

Ve druhé části přehrávače se zobrazují následující multimédia – videa, webové příspěvky, obrázky, textový popis a logo rádia. Videá, obrázky a textový popis jsou vázány na přehrávaného hudebníka, zatímco webové příspěvky a logo na danou rádiovou stanici. Hudebníci a rádiové stanice jsou v systému reprezentováni jako příspěvky nových typů (více později v této kapitole). V následujícím výčtu jsou podrobněji popsána jednotlivá multimédia:

- *Videa* jsou nahrávky (příp. úvodní nahrávky) ve formátu MP4, které jsou přiřazeny danému hudebníkovi.
- *Webové příspěvky* jsou vybrané příspěvky z webu, které mají titulní fotografií (tzv. „Featured image“) a textový obsah. S každou přehrávanou písni je zobrazen vždy jiný webový příspěvek. V přehrávači je zobrazena jeho titulní fotografie (včetně doprovodných titulků). Po kliknutí na tuto fotografií se místo ní zobrazí obsah daného webového příspěvku (s možností návratu zpět).
- *Obrázky* jsou soubory ve formátech JPG/PNG/GIF, které jsou přiřazeny danému hudebníkovi. V přehrávači se zobrazují jako snímky slideshow (včetně doprovodných titulků). Každý snímek je zobrazen po určitý čas. Přechody mezi snímky jsou plynulé. Po kliknutí na jakýkoli snímek se místo něj zobrazí popis daného hudebníka (s možností návratu zpět).
- *Popis* je obsah příspěvku s hudebníkem. Může obsahovat kupříkladu informace o jeho životě či kariéře.
- *Logo rádia* je titulní fotografie (tzv. „Featured image“) příspěvku s rádiovou stanicí.

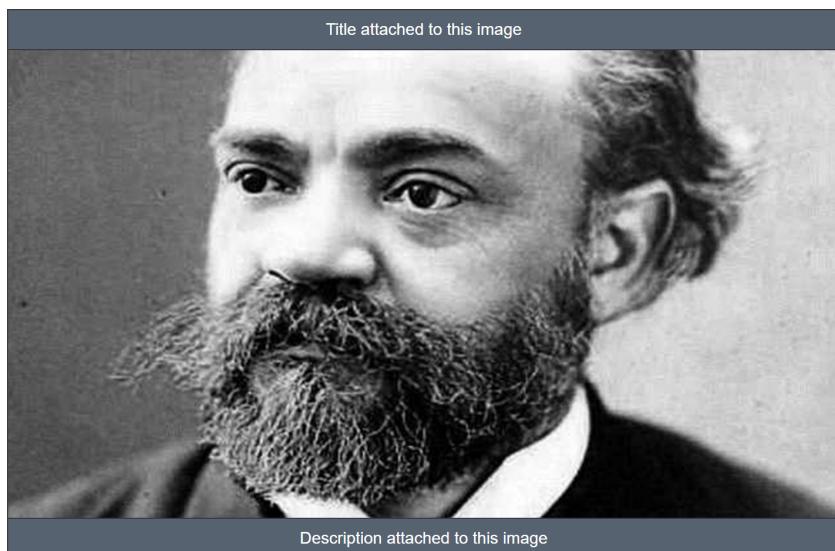
#### 4. NÁVRH

---

Multimédia se řídí následujícím pravidlem, které vyznačuje jejich priority:

**Videa > Webové příspěvky > Obrázky > Popis > Logo**

Pokud se přehrává videonahrávka, ostatní multimédia napravo od ní jsou skrytá. Pokud se přehrává audionahrávka a zároveň se promítají webové příspěvky, pak jsou v přehrávači zobrazeny právě tyto příspěvky (i přesto, že hudebník má obrázky i popis). V případě ostatních multimédií je chování analogické. Pokud hudebník nemá žádná videa, obrázky ani popis a webové příspěvky se nepromítají, pak je zobrazeno logo rádiové stanice.



Obrázek 4.3: Druhá část přehrávače

##### 4.2.1.3 Ovládání

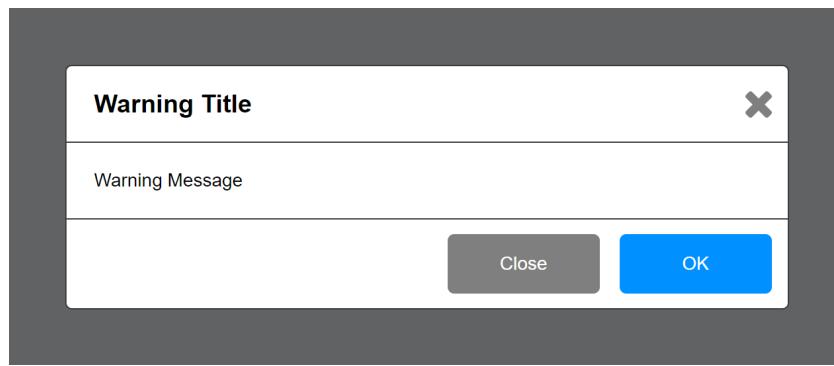
Poslední částí přehrávače je ovládání, které obsahuje tlačítko pro spuštění, příp. zastavení přehrávání. Tlačítko je záměrně červeně podbarvené, aby upoutalo pozornost posluchače [5]. Automatické spuštění přehrávání je (narozdíl od předešlého stavu rádií) vypnuté, jelikož představuje špatné UX (tj. „User eXperience“) [27]. V této části přehrávače se zobrazuje i jméno hudebníka a název skladby, která se v danou chvíli přehrává. Obě položky mají vlastní popisky (např. „Musician:“ nebo „Song:“). Ovládání používá mřížku knihovny Bootstrap a je tak responzivní na různých zařízeních.



Obrázek 4.4: Poslední část přehrávače

#### 4.2.1.4 Výstražná hlášení

Nepřímou součástí přehrávače jsou také výstražná hlášení. Jedná se o dialogová okna s tmavým pozadím, která se pravidelně zobrazují po přehrání určitého počtu hudebníků. Zatímco je výstraha zobrazená, přehrávání je po-zastavené. Každá výstraha obsahuje nadpis, obsah, ikonku křížku a tlačítka „Close“ a „OK“. Pomocí tlačítek a křížku lze výstrahu ukončit a pokračovat v poslechu. Tlačítko „OK“ však může být spojeno s libovolnou URL adresou, na kterou se posluchač po kliknutí přesměruje (v nové záložce prohlížeče).

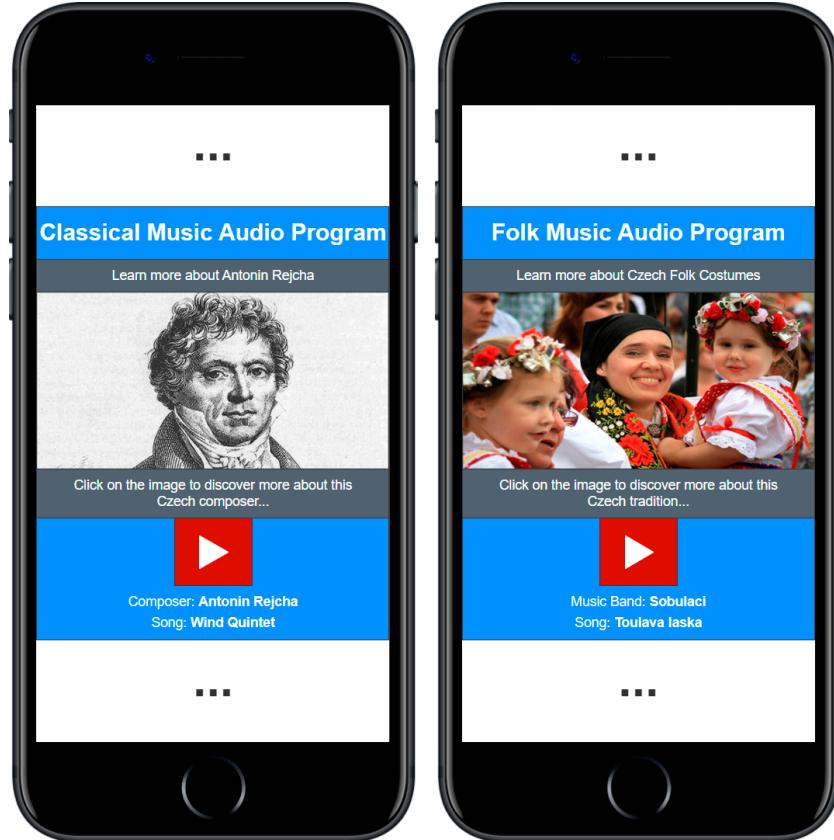


Obrázek 4.5: Výstražné hlášení

## 4. NÁVRH

### 4.2.2 Kontext

Vzhledem k tomu, že přehrávač rádia lze vložit do jakéhokoli příspěvku v systému WordPress, má administrátor (příp. editor) možnost vytvářet různé doprovodné prvky. Mezi ně patří např. úvodní poutavý text nad přehrávačem nebo technické informace pod ním. Takové prvky však nejsou součástí přehrávače ani pluginu *Radio Manager*.



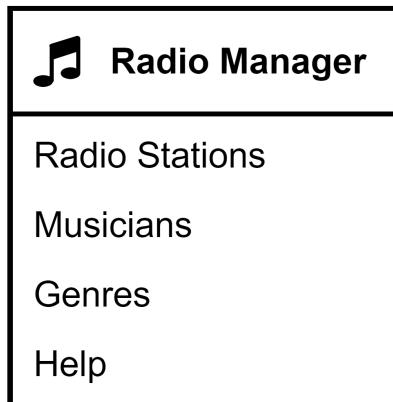
Obrázek 4.6: Ukázka přehrávačů na mobilních telefonech

### 4.3 Backend

Výstupem pluginu v administraci je sekce pojmenovaná jako *Radio Manager*, která se nachází v levém navigačním menu systému WordPress. Její název je vyznačen tematickou WordPress ikonou `dashicons-format-audio` (tj. dvoujice not). Mezi položkami uvedené sekce jsou:

- *Radio Stations* (stránka se správou rádiových stanic),
- *Musicians* (stránka se správou hudebníků),

- *Genres* (stránka se správou žánrů),
- *Help* (uživatelská příručka).



Obrázek 4.7: Rozhraní pluginu v administraci

Přístup k jednotlivým položkám je z bezpečnostních důvodů omezen. Uživatelé musí mít potřebná oprávnění. Administrátoři mají přístup ke všem položkám (díky oprávnění `manage_options`). Editoři mají přístup pouze k položkám *Musicians* a *Help* (díky oprávnění `edit_posts`). Ostatní uživatelé, kteří nemají oprávnění `manage_options` ani `edit_posts`, tuto sekci nevidí. V následujících podkapitolách jsou popsány všechny uvedené položky a jejich obsah. Doprovodné náčrty a wireframes byly vytvořeny v nástroji Axure RP.

#### 4.3.1 Rádiové stanice

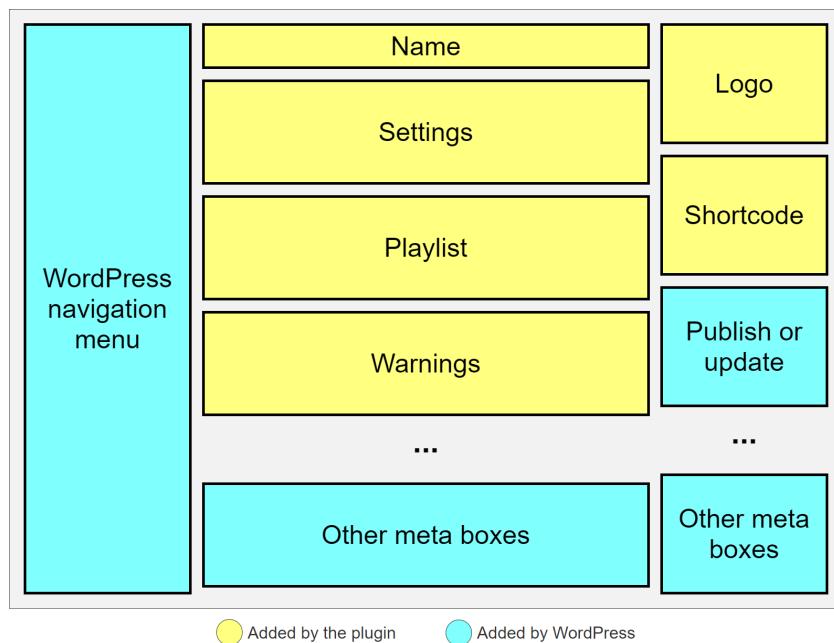
Obsahem položky *Radio Stations* je přehledný seznam, kde lze libovolně přidávat, upravovat, příp. mazat příspěvky s rádiovými stanicemi. Každé rádio je reprezentováno jako příspěvek nového typu s názvem `rm_radio_station`.

U každého příspěvku lze změnit název a titulní fotografii (tj. logo rádia), která zároveň slouží ke snadné orientaci v uvedeném seznamu. Dále je k dispozici unikátní shortcode, pomocí kterého lze rádio vložit na webovou stránku (více později v této kapitole). Následují tři uživatelská pole (vytvořená pomocí pluginu Advanced Custom Fields) pro nastavení, playlist a výstražná hlášení. Všechna jsou popsána v následujících podkapitolách.

Rozložení jednotlivých prvků je znázorněno na následujícím wireframe. Světle modré jsou označeny všechny prvky, které jsou automaticky přidané systémem WordPress. Zatímco světle žlutě jsou označeny prvky dodané pluginem. Rozložení většiny prvků může být ručně změněno.

## 4. NÁVRH

---



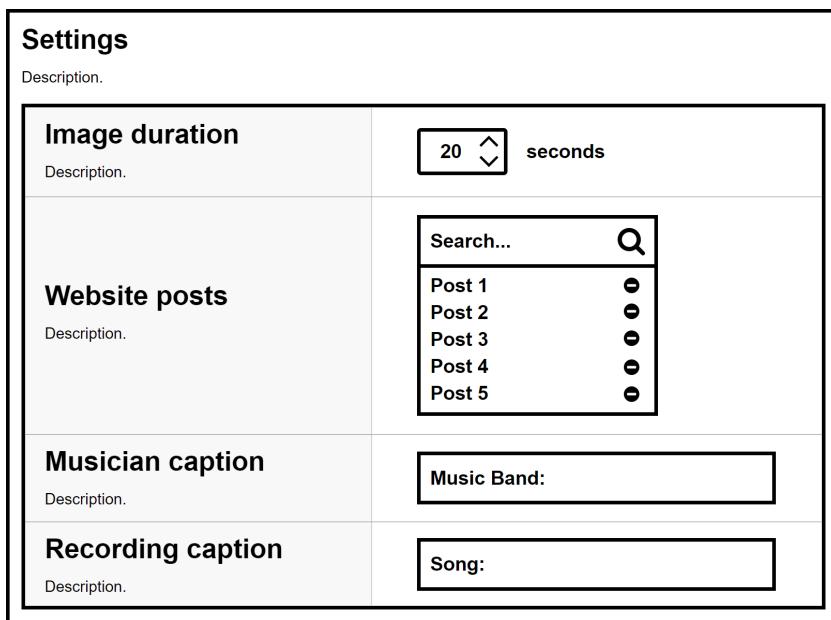
Obrázek 4.8: Rozložení prvků v příspěvku rádiové stanice

### 4.3.1.1 Nastavení

Prvním uživatelským polem (typu „Group“) v příspěvku s rádiovou stanicí je nastavení. Skládá se ze čtyř položek:

- *Image duration* – doba zobrazení jednoho obrázku (v sekundách), který se promítá v přehrávači jako snímek slideshow.
- *Website posts* – rozhraní pro výběr libovolných příspěvků z webu. Tyto příspěvky se následně promítají v přehrávači. Promítání webových příspěvků je však nutné aktivovat pro konkrétní položky playlistu (více níže). Webové příspěvky jsou promítány ve stylu shuffle.
- *Musician caption* – popisek hudebníka v přehrávači (např. „Composer:“ či „Music Band:“).
- *Recording caption* – popisek nahrávky v přehrávači (např. „Song:“).

Každá položka má textovou nápovědu (na místě „Description.“).



Obrázek 4.9: Nastavení rádiové stanice

#### 4.3.1.2 Playlist

Druhým uživatelským polem (typu „Repeater“) v příspěvku s rádiem je playlist. Playlist se skládá z libovolného počtu položek, které může administrátor přidávat, upravovat, příp. mazat. Každá položka obsahuje čtyři podpoložky:

- *Genres* – hudební žánry, které představují různé skupiny hudebníků (např. „Classical“ či „Folk“). Při přehrávání dané playlistové položky je zvolen jeden z těchto žánrů.
- *Number of musicians* – počet hudebníků vybíraných ze zvoleného hudebního žánru.
- *Number of recordings* – počet nahrávek vybíraných od zvolených hudebníků. Tyto nahrávky jsou následně přehrány.
- *Show website posts* – možnost aktivovat promítání webových příspěvků při přehrávání nahrávek vybraných v rámci dané playlistové položky.

Všechny podpoložky mají textové návodky (na místě „Description.“). Vzhledem k náročnosti zadavatelových požadavků a velkému množství písni nebylo možné playlist pojmut jako „obyčejný seznam nahrávek“. Z tohoto důvodu je zvolena vyšší abstrakce založená na hudebních žánrech (tj. skupinách hudebníků), ke kterým lze jednotlivé hudebníky libovolně přiřazovat.

#### 4. NÁVRH

Jako příklad poslouží následující wireframe. Řekněme, že obsahem daného playlistu je pouze první položka. Nejdříve je vybrán hudební žánr (v tomto případě je vždy zvolen „Classical“). Z tohoto žánru jsou vybráni tři skladatelé vážné hudby. Od každého z nich je vybrána jedna nahrávka. Rádiová stanice tedy přehrává tři skladby od různých skladatelů a takto stále dokola. Promítání webových příspěvků je vypnuté, tudíž se v průběhu přehrávání promítají obrázky skladatelů. Žánry a hudebníci jsou vybíráni ve stylu shuffle.

The wireframe shows a configuration interface for a radio station's playlist. At the top, there is a field labeled "Playlist\*" with a placeholder "Description.". Below it is a section titled "Genres\*" with a "CLASSICAL" button. The main configuration area contains four rows of settings:

- Number of musicians**: A slider set to 3, with a dropdown menu showing "3" and "musicians".
- Number of recordings**: A slider set to 1, with a dropdown menu showing "1" and "recordings".
- Show website posts**: A toggle switch set to "OFF".

Below these settings is a horizontal ellipsis (...). Further down is a section labeled "Last playlist item" with a delete icon. At the bottom right is a button labeled "ADD PLAYLIST ITEM".

Obrázek 4.10: Playlist rádiové stanice

##### 4.3.1.3 Výstražná hlášení

Posledním uživatelským polem (typu „Repeater“) v příspěvku s rádiem jsou výstražná hlášení. Administrátor může přidávat, upravovat, příp. mazat jednotlivé výstrahy. Každá výstraha obsahuje následující položky:

- *Active* – možnost aktivovat či deaktivovat výstražné hlášení.
- *Show to* – skupina uživatelů, kterým se výstraha zobrazuje. K dispozici jsou volby „All users“ (všichni uživatelé), „Registered in the s2Member“ (registrovaní) a „Not registered in the s2Member“ (neregistrovaní).
- *First* – počet hudebníků, po kterých se výstraha zobrazí poprvé. Pomocí této položky lze výstražné hlášení zobrazit např. ihned po spuštění přehrávače (*First: 0*).

- *Step* – počet hudebníků, po kterých se výstraha pravidelně zobrazuje (počítáno od prvního zobrazení ve *First*).
- *Title* – nadpis výstrahy.
- *Message* – obsah výstrahy.
- *Link* – URL adresa, na kterou je posluchač přesměrován po kliknutí na tlačítko „OK“. Pokud není vyplněna, pak toto tlačítko ukončuje výstražné hlášení.

Každá položka má textovou ná povědu (na místě „Description.“).

| Warnings           |  |
|--------------------|--|
| Description.       |  |
| <b>Active</b>      | <input checked="" type="checkbox"/> ON   |
| <b>Show to</b>     | NOT REGISTERED... ▾  |
| <b>First</b>       | 3 ^▼ musicians   |
| <b>Step</b>        | 3 ^▼ musicians   |
| <b>Title</b>       | MEMBERSHIP WARNING POPUP   |
| <b>Message*</b>    | Do you want to continue listening without interruptions? Then click the OK button and... |
| <b>Link</b>        | HTTPS:// ...   |
| ...                |  |
| Last warning       | -  |
| <b>ADD WARNING</b> |  |

Obrázek 4.11: Výstražná hlášení rádiové stanice

### 4.3.2 Hudebníci

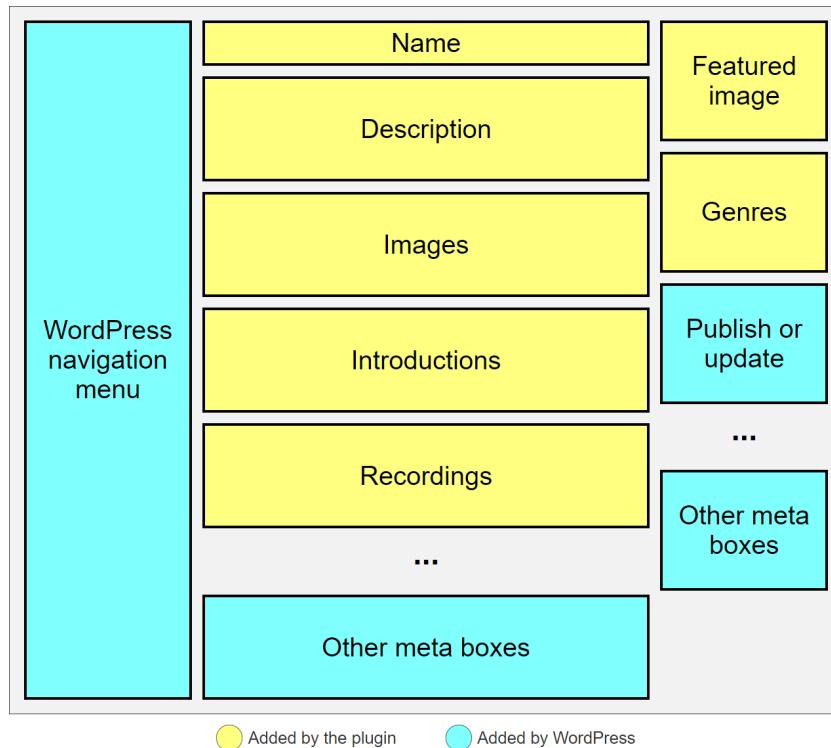
Podobně jako u rádií je obsahem položky *Musicians* přehledný seznam, kde lze libovolně přidávat, upravovat, příp. mazat příspěvky s hudebníky. Každý hudebník je reprezentován jako příspěvek nového typu s názvem `rm_musician`.

#### 4. NÁVRH

---

U každého příspěvku lze změnit název (tj. jméno hudebníka), obsah (tj. popis hudebníka) a titulní fotografiu, která slouží ke snadné orientaci v uvedeném seznamu. Pro úpravy obsahu slouží klasický WYSIWYG editor systému WordPress. K dispozici jsou také hudební žánry, ke kterým lze hudebníka přiřadit. Následují tři uživatelská pole (typu „Repeater“) pro obrázky, úvodní nahrávky a nahrávky daného hudebníka. Obrázky se promítají v průběhu přehrávání nahrávek (např. písni). Úvodní nahrávky (např. vstupy moderátora) se přehrávají před nahrávkami. U obrázků lze upravovat doprovodné titulky a u nahrávek jejich názvy. Uvedená multimédia se nahrávají do knihovny médií a pro přehrávání jsou vybírána ve stylu shuffle.

Rozložení jednotlivých prvků je znázorněno na následujícím wireframe. Světle modré jsou označeny všechny vestavěné prvky systému WordPress. Zatímco světle žluté jsou označeny prvky dodané pluginem. Rozložení většiny prvků může být ručně změněno.



Obrázek 4.12: Rozložení prvků v příspěvku hudebníka

##### 4.3.3 Žánry

Obsahem položky *Genres* je přehledný seznam, kde lze libovolně přidávat, upravovat, příp. mazat termíny nové taxonomie `rm_genre`. Mezi ně patří např. „Classical“, „Folk“, ba dokonce „Announcements“ (reklaumní oznámení) nebo

#### 4.4. Propojení frontendové a backendové části

---

„Jingles“ (znělky). Výběr termínů je k dispozici při tvorbě hudebníka. Administrátor (příp. editor) tak může jednotlivé hudebníky přiřazovat k různým žánrům, ze kterých je následně vytvářen playlist rádiové stanice.

##### 4.3.4 Pomoc

Obsahem položky *Help* je uživatelská příručka, která slouží jako podpora pro začínající uživatele. Může být použita i v případě různých nesnází či problémů. Uživatelská příručka je k dispozici v příloze B.

## 4.4 Propojení frontendové a backendové části

Každá rádiová stanice obsahuje shortcode. Jedná se o krátký kód, pomocí kterého lze dané rádio vložit do libovolného příspěvku v systému WordPress. Jeho parametrem je **id**, které představuje identifikační číslo příspěvku rádiové stanice. Administrátor může použít tlačítko *Copy* pro zkopirování shortcode, který následně vloží do jakéhokoli příspěvku. Po zobrazení příspěvku na webu se místo shortcode vykreslí přehrávač dané rádiové stanice. Na webové stránce může být vykresleno i více přehrávačů rádií, které fungují nezávisle na sobě. Implementační detaily se lze dočíst v následující kapitole.



Obrázek 4.13: Shortcode v příspěvku rádiové stanice



# KAPITOLA 5

## Implementace

V této kapitole je popsána implementace obou částí (tj. frontend a backend) nového WordPress pluginu *Radio Manager* včetně jejich propojení. Implementace byla zhotovena na základě návrhu a analýzy požadavků zadavatele. Výklad je doplněn o pomocné diagramy. V další kapitole je popsáno testování vytvořeného pluginu.

### 5.1 Vývoj

Plugin *Radio Manager* byl vyvíjen v prostředí Microsoft Visual Studio 2019 (s občasnou pomocí textového editoru Notepad++). Veškerý kód je uložen ve vzdáleném Git repozitáři, který se nachází na oficiálním GitHub účtu organizace CATV USA. Tento repozitář je privátní a mohou k němu přistupovat pouze oprávnění členové. Repozitář obsahuje dvě větve – *dev*, na které probíhal vývoj, a *main*, která obsahuje funkční a otestovaný kód z vývojářské větve. Od počátku vývoje bylo do repozitáře pravidelně přispíváno prostřednictvím vhodně pojmenovaných commitů. Významným commitům byly přiřazeny tagy ve formě čísel verzí (majoritních, minoritních i opravných). Veškerý kód byl komentován pomocí JSDoc a PHPDoc. Jeho testování probíhalo na testovací webové stránce CATV USA, která představuje kopii oficiální webové stránky [www.catvusa.com](http://www.catvusa.com).

### 5.2 Frontend

Frontendová část pluginu *Radio Manager* zahrnuje UI a logiku rádiových stanic na webové stránce. Tato část byla naprogramována pomocí skriptovacích jazyků JavaScript a Sass jako balíček NPM, který pracuje na straně klienta. Při jeho vývoji byly použity i další technologie, jako např. Babel, Bootstrap, Gulp, React či Swiper. Veškerý frontend je obsažen ve dvou balíčcích zkompilovaného a minifikovaného kódu – `rm-scripts.min.js` a `rm-styles.min.css`.

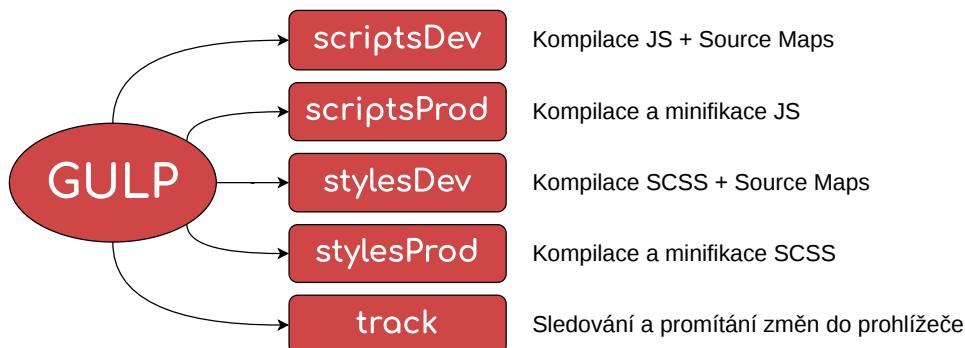
## 5. IMPLEMENTACE

---

První balíček obsahuje ES6 třídy a React komponenty. Druhý balíček obsahuje kaskádové styly pro tyto komponenty. Názvy balíčků obsahují unikátní prefix „rm“ (tj. zkratka názvu *Radio Manager*) jako ochranu proti případným kolizím. Po zobrazení webové stránky jsou uvedené balíčky načteny do prohlížeče uživatele.

### 5.2.1 Automatizace procesů

Zmíněné balíčky kódu jsou výstupy nástroje Gulp. Tento nástroj umožňuje zjednodušit, automatizovat a urychlit různé procesy, se kterými se lze při vývoji software setkat. V případě pluginu *Radio Manager* se Gulp používá pro zpracování skriptů, kaskádových stylů a sledování změn příslušných souborů, které jsou následně promítány do prohlížeče. Zpracování JS a SCSS se dále liší podle toho, zdali jsou určené pro vývojové nebo produkční prostředí. Pro všechny uvedené procesy byly vytvořeny následující tasks – **scriptsDev**, **scriptsProd**, **stylesDev**, **stylesProd** a **track**. Tyto tasks jsou veřejné (lze je spustit pomocí příkazu `gulp <názevTasku>`).



Obrázek 5.1: Diagram Gulp tasks

Pomocí Gulp metod `series()` a `parallel()` lze spustit více tasks, a to buď sekvenčně (za sebou), nebo paralelně (zároveň). Pro vývoj frontendové části byly vytvořeny následující kombinace:

- `series( parallel( scriptsDev, stylesDev ), track )`
- `parallel( scriptsProd, stylesProd )`

První kombinace provádí paralelní komplikaci JS a SCSS včetně tzv. „Source Maps“ (více níže). Po dokončení těchto tasks je spuštěno sledování a promítání změn příslušných souborů do prohlížeče. Tato kombinace je nastavena jako tzv. výchozí task (spouští se po zadání příkazu `gulp`), protože byla při vývoji frontendové části nejvíce používaná.

Druhá kombinace opět provádí paralelní komplikaci JS a SCSS (tentokrát bez „Source Maps“). Výsledné soubory jsou následně minifikovány. Výsledek této kombinace lze použít na webové stránce v produkčním prostředí.

Během vývoje stačí pomocí CLI zadat příkaz `npm start` (tj. spuštění vlastního NPM balíčku s frontendovou částí), který je namapovaný na příkaz `gulp`. Tento příkaz následně spustí výchozí task. Programátorovi je spuštěno prostředí, ve kterém může snáze a rychleji vyvíjet. Zároveň jej osvobozuje od velkého množství ruční práce spojené např. s komplikací, sestavením a minifikací JS nebo SCSS souborů. V následujících podkapitolách jsou popsány jednotlivé procesy podrobněji.

### 5.2.1.1 Skripty

Frontendová část byla napsána pomocí jazyka JavaScript. Při tvorbě skriptů byly použity i novější prvky a konstrukty (např. generátory) ze standardů ECMAScript 2015 (ES6) a vyšších. Zároveň byla použita technologie React včetně syntaxe JSX. Tasky `scriptsDev` a `scriptsProd` provádějí inkrementální komplikaci všech JS souborů pomocí technologie Babel, pro kterou jsou nastaveny presety `@babel/preset-env` a `@babel/preset-react`. Zkomplikované soubory jsou následně sloučeny do jediného souboru (tzv. „bundle“).

V případě tasku `scriptsDev` jsou součástí kódu také tzv. „Source Maps“, které mapují zkomplikovaný JavaScript na původní soubory, čímž usnadňují jeho ladění a hledání případných chyb. V případě tasku `scriptsProd` je zkomplikovaný kód jen minifikován (bez Source Maps). Task `scriptsDev` je určen pro vývoj v rámci testovacího prostředí, zatímco task `scriptsProd` pouze generuje finální JS balíček pro produkční prostředí.

### 5.2.1.2 Kaskádové styly

Při tvorbě frontendové části byl kromě jazyka JavaScript použit i jazyk Sass, ve kterém jsou napsané veškeré kaskádové styly. Pomocí tohoto jazyka lze psát CSS mnohem úsporněji a bez opakujících se částí kódu. Tasky `stylesDev` a `stylesProd` provádějí komplikaci všech SCSS souborů do obyčejných CSS, kterým rozumí běžné webové prohlížeče.

V případě tasku `stylesDev` jsou součástí kódu také Source Maps, které usnadňují hledání chyb v CSS. V případě tasku `stylesProd` je zkomplikovaný kód jen minifikován (bez Source Maps). Podobně jako u skriptů je task `stylesDev` určen pouze pro vývoj, zatímco task `stylesProd` slouží pro vygenerování finálního CSS balíčku pro produkční prostředí.

### 5.2.1.3 Sledování změn

Task `track` umožňuje pomocí technologie Browsersync spustit vlastní lokální server s rádiovou stanicí přímo v prohlížeči. Gulp nadále sleduje HTML, SCSS a JS soubory pomocí metody `watch()`. V případě modifikace jakéhokoli

## 5. IMPLEMENTACE

---

z těchto souborů se změny ihned projeví v prohlížeči. Okamžitá zpětná vazba bez nutnosti stránku aktualizovat (např. pomocí klávesy F5) urychluje vývoj. Spuštěná rádiová stanice je navíc přístupná i pro ostatní elektronická zařízení připojená v dané síti. Výsledek je tedy možné otestovat v různých prohlížečích a zařízeních, aniž by se musel nahrávat na internet.

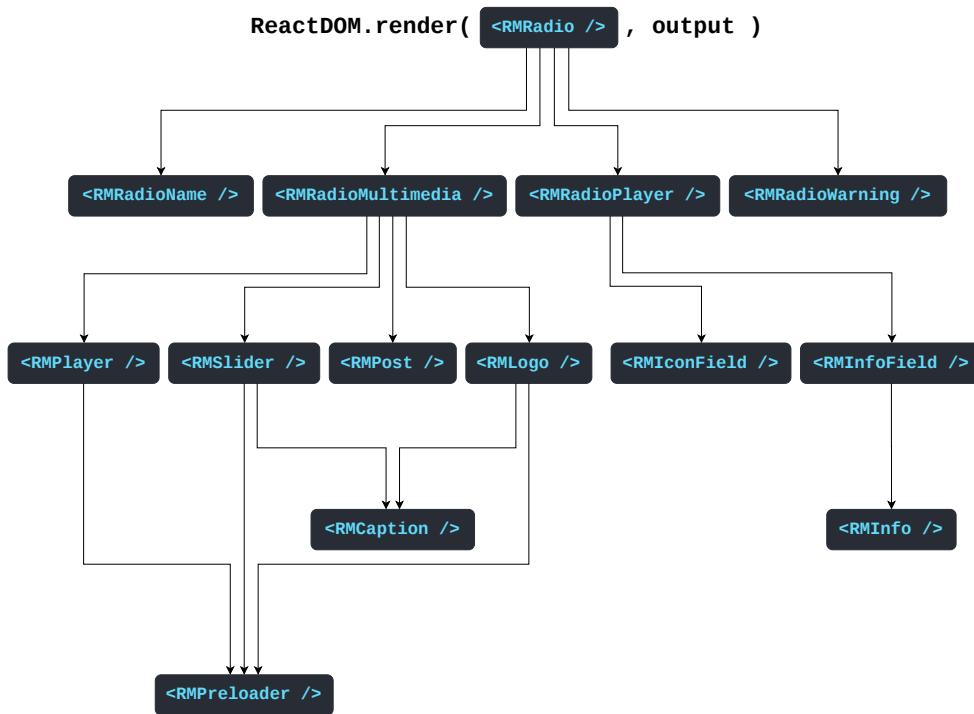
### 5.2.2 Komponenty UI

Uživatelské rozhraní každé rádiové stanice je rozdělené do menších, samostatných a znovupoužitelných částí kódu – komponent. Pro jejich tvorbu byla použita technologie React a syntaxe JSX, která umožňuje zapisovat obdobu HTML do JS kódu. Každá komponenta je implementovaná jako ES6 třída, která přijímá určitý vstup (tzv. Props) a generuje určitý výstup v podobě React elementů nebo dalších React komponent.

Nejvyšší komponentou v hierarchii je `RMRadio`, která je pomocí knihovny ReactDOM a metody `render()` vložena na předem určené místo v modelu DOM. Této komponentě jsou předány počáteční parametry pro inicializaci jejího obsahu a dalších komponent. Komponenta také obsahuje stav (tzv. State) – privátní data, která jsou důležitá pro správnou funkčnost dané rádiové stanice. Tato stavová data (příp. metody, pomocí kterých je lze měnit) jsou následně předávána dalším komponentám umístěným níže v hierarchii. Každá komponenta obsahuje informace o UI, příp. logiku.

Z bezpečnostních důvodů React implicitně odstraňuje HTML z výrazů použitých v syntaxi JSX (např. z JS proměnných). Vše převádí na řetězce znaků, a tudíž předchází škodlivým XSS útokům. Tato ochrana je u rádiových stanic potlačena pouze v případě textů o hudebnících (příp. webových příspěvků) a zpráv výstražných hlášení. V prvním případě se jedná o zobrazení formátovaného příspěvku ze systému WordPress. Ve druhém případě se jedná o zobrazení formátovaného textu, který je uložený v systému pomocí pluginu Advanced Custom Fields. Pro oba případy je vyžadováno parsování HTML, aby byl daný obsah korektně zobrazen uživateli.

Následující diagram zobrazuje hierarchii React komponent, které zajišťují vzhled a logiku rádiové stanice na webové stránce.



Obrázek 5.2: Diagram React komponent

Každá komponenta má v rámci hierarchie určitý účel a zodpovědnost. V následujícím seznamu jsou stručně popsány všechny uvedené komponenty:

- `RMRadio` je nejvyšší komponenta, která obsahuje stav. Vykresluje rádiovou stanici jako celek rozdelený na 4 hlavní části – název, multimédia, vizuální přehrávač a výstražná hlášení.
- `RMRadioName` představuje název rádiové stanice.
- `RMRadioMultimedia` je proměnlivou částí rádia, ve které se přehrávají videa, promítají obrázky (příp. logo) a zobrazují texty.
- `RMRadioPlayer` vykresluje vizuální přehrávač, který obsahuje ikonky pro ovládání a informace o přehrávaném hudebníkovi a písni.
- `RMRadioWarning` není součástí rádiové stanice vizuálně. Zobrazuje se v průběhu přehrávání jako výstražné okno. Využívá také další externí komponenty `Button` a `Modal` pro vykreslení ovládacích tlačítek a samotného výstražného okna z knihovny Bootstrap.

## 5. IMPLEMENTACE

---

- **RMPlayer** obsahuje veškerou logiku pro přehrávání a skrytý audio/video přehrávač, který postupně prochází playlist rádiové stanice.
- **RMSlider** je složený z externích komponent **Swiper** a **SwiperSlide** od technologie Swiper. Pomocí těchto komponent je vytvořen slider, který promítá různé obrázky (např. fotografie skladatele).
- **RMPost** představuje textový příspěvek, ve kterém se lze dočíst více informací o přehrávaném hudebníkovi. Zároveň se může jednat o jakýkoliv jiný článek z webové stránky (pokud jej administrátor nastaví k promítání). V této komponentě se používá také další externí komponenta **Button** pro vykreslení ovládacích tlačítek z knihovny Bootstrap.
- **RMLogo** vykresluje logo rádiové stanice.
- **RMIIconField** je část vizuálního přehrávače s ikonami pro spuštění, příp. zastavení přehrávání. Tyto ikonky byly převzaty z knihovny Bootstrap.
- **RMInfoField** je část vizuálního přehrávače s informacemi o právě přehrávaném hudebníkovi a písni.
- **RMCaption** představuje horní či spodní titulek promítaného obrázku.
- **RMInfo** je konkrétní informace zobrazená ve vizuálním přehrávači, např. jméno přehrávaného hudebníka nebo název písni.
- **RMPreloader** lze spatřit v případě pomalého načítání videí nebo obrázků. Jedná se o bílou rotující kružnici, která signalizuje načítání. Tento prvek byl převzat od technologie Swiper.

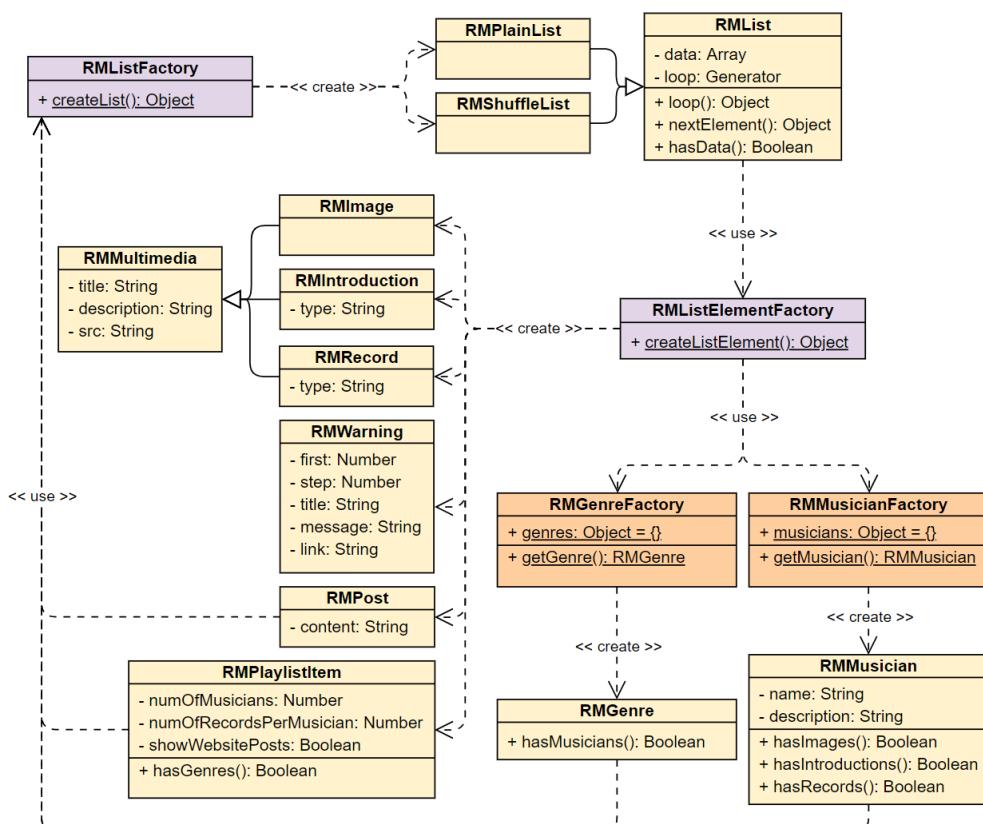
### 5.2.3 Reprezentace dat

Veškerá data přichází do frontendové části ve formě jediného JS objektu. Tento objekt je následně zpracován React komponentami. Příchozí data obsahují následující informace (v závorce je vždy uvedena React komponenta, do které tato data směřují):

- název rádia (→ **RMRadioName**),
- logo rádia (→ **RMLogo**),
- popisky pro jméno hudebníka a název písni (→ **RMInfo**),
- doba zobrazení jednoho obrázku (→ **RMSlider**),
- playlistové položky, žánry, hudebníci (včetně jejich obrázků, úvodních nahrávek a písni), webové příspěvky a výstražná hlášení (→ **RMPlayer**).

Data směřující do komponenty `RMPlayer` jsou velmi důležitá pro přehrávání dané rádiové stanice. Jedná se o velice dynamická data, se kterými přehrávač pravidelně pracuje. Jejich manipulace by tak neměla být příliš náročná a složitá. Z tohoto důvodu pro ně byla zvolena reprezentace pomocí ES6 tříd a OOP. Zároveň byly použity i návrhové vzory Factory a Flyweight, o kterých se lze dočíst více v následujících podkapitolách.

Uvedená data jsou reprezentována jako seznamy prvků. Při tvorbě takového seznamu je nejprve určen jeho typ. Následně je určen typ prvků, které bude obsahovat. Nakonec je těmito prvky (daty) naplněn. Každý seznam je implementován pomocí ES6 generátoru, který cyklicky prochází a vrací jednotlivé prvky. V případě potřeby se přehrávač odvolá na daný seznam (generátor), který mu vrátí další prvek. Na této logice je postavené procházení playlistových položek a žánrů, střídání hudebníků a promítání jejich obrázků, přehrávání úvodních nahrávek a písni, promítání webových příspěvků a zobrazování výstražných hlášení.



Obrázek 5.3: Diagram tříd frontendové části

## 5. IMPLEMENTACE

---

K atributům všech tříd je přistupováno pouze skrze `get` metody. Z tohoto důvodu byly všechny atributy označeny jako privátní, i když jazyk JavaScript v současné době takové atributy nepodporuje. V následujícím seznamu jsou stručně popsány všechny třídy uvedené v diagramu:

- **RMListFactory** je továrna na tvorbu různých druhů seznamů.
- **RMPplainList** reprezentuje obyčejný seznam (resp. generátor, který cyklicky iteruje přes všechny prvky seznamu v takovém pořadí, v jakém do něj byly uloženy)
- **RMSshuffleList** reprezentuje míchaný seznam (resp. generátor, který cyklicky iteruje přes všechny prvky seznamu, přičemž na začátku každé smyčky jsou tyto prvky náhodně zamíchány). Pro míchání prvků se používá NPM balíček `array-unsort` (resp. algoritmus Fisher-Yates).
- **RMList** je rodičovská třída pro všechny druhy seznamů.
- **RMListElementFactory** je továrna na tvorbu různých prvků seznamů.
- **RMPplaylistItem** reprezentuje playlistovou položku. Rádiová stanice má obyčejný seznam těchto položek (playlist). Každá položka má míchaný seznam žánrů.
- **RMGenreFactory** je továrna muší váhy, která se používá pro tvorbu a ukládání hudebních žánrů.
- **RMGenre** představuje hudební žánr. Každý žánr má míchaný seznam hudebníků.
- **RMMusicianFactory** je továrna muší váhy, která se používá pro tvorbu a ukládání hudebníků.
- **RMMusician** představuje hudebníka. Každý hudebník má míchané seznamy obrázků, úvodních nahrávek a nahrávek.
- **RMMultimedia** je rodičovská třída pro obrázky, úvodní nahrávky a nahrávky daného hudebníka.
- **RMIImage** je obrázek daného hudebníka.
- **RMIIntroduction** je úvodní nahrávka, která se přehrává vždy před písňemi daného hudebníka.
- **RMRecord** je píseň daného hudebníka.
- **RMWarning** reprezentuje výstražné hlášení, které se posluchači pravidelně zobrazuje v průběhu přehrávání. Rádiová stanice má obyčejný seznam těchto hlášení.

- **RMPost** reprezentuje libovolný příspěvek z webové stránky, kterým lze vyplnit obsah rádia v průběhu přehrávání (např. pokud vícero hudebníkům chybí textové popisy i obrázky). Rádiová stanice má míchaný seznam těchto příspěvků.

### 5.2.3.1 Použití návrhového vzoru Factory

Návrhový vzor Factory patří do skupiny tvořivých vzorů, které se zabývají vytvářením nových objektů. Jeho použitím lze eliminovat počet výskytů klíčového slova **new** v programu. Továrnou je myšlena speciální třída, která definuje rozhraní pro tvorbu nových objektů. Pro vytvoření konkrétního objektu je pak zavolána tovární metoda, které je předána informace o typu nového objektu. Metoda tento objekt vytvoří a vrátí volajícímu. Pomocí návrhového vzoru Factory lze snadněji vytvářet různé objekty, příp. měnit jejich inicializaci na jediném místě v programu. [28]

V programu existují dvě továrny, které implementují statické tovární metody pro tvorbu nových objektů. První továrna vytváří různé druhy seznamů – obyčejný (plain) a míchaný (shuffle). Druhá továrna vytváří prvky těchto seznamů – playlistové položky, žánry, hudebníky, obrázky, úvodní nahrávky, nahrávky, výstražná hlášení a webové příspěvky. Tyto továrny umožňují vytvářet seznamy a jejich prvky mnohem snadněji a přehledněji.

Následující ukázka obsahuje statickou tovární metodu, která vytváří prvky seznamů. Z ukázky byly odstraněny doprovodné komentáře. Samotná tovární metoda je zkrácená. Při tvorbě hudebníka (příp. žánru) je používána další továrna, která souvisí s následujícím návrhovým vzorem Flyweight.

```
static createElement( elementType, object )
{
    if ( elementType == "musician" )
    {
        return RMMusicianFactory.getMusician( object )
    }
    else if ( elementType == "image" )
    {
        return new RMImage( object )
    }
    else if ( elementType == "introduction" )
    {
        return new RMIntroduction( object )
    }
    ...
}
```

Kód 5.1: Statická tovární metoda pro tvorbu prvků seznamů

### 5.2.3.2 Použití návrhového vzoru Flyweight

Návrhový vzor Flyweight patří do skupiny strukturálních vzorů, které se zabývají uspořádáním jednotlivých tříd v programu. Cílem tohoto návrhového vzoru je sdílení co největšího množství dat mezi podobnými objekty. Tím lze ušetřit paměťové prostředky, ba dokonce snížit počet instancí potřebných k chodu aplikace. V programu se nachází továrny muší váhy, které spravují sdílené objekty a vytváří nové. Při tvorbě nového objektu si daná továrna nejdříve zkонтroluje, zdali tento objekt již nespravuje. Pokud ne, pak jej vytvoří a uloží do továrny. V obou případech vrátí objekt volajícímu. [28]

Návrhový vzor Flyweight se používá při tvorbě hudebníků a žánrů. Hudebník může být přiřazen více žánrům. Bez použití muší váhy by každý žánr pracoval s vlastní instancí identického hudebníka, což by navíc vedlo k problémům v podobě špatně fungujícího shuffle módu při přehrávání jeho písni. Podobná situace by nastala i při tvorbě žánrů. Playlist rádiové stanice může obsahovat více položek se stejným hudebním žánrem. Bez muší váhy by každá položka pracovala s vlastní instancí daného žánru. To by opět vedlo k problémům s shuffle módem a zbytečnému navýšení počtu objemných instancí. Díky návrhovému vzoru Flyweight existuje pouze jediná instance hudebníka přiřazeného více žánrům a žánru přiřazeného více playlistovým položkám.

Následující ukázka obsahuje továrnu muší váhy pro tvorbu hudebníků. Z ukázky byly odstraněny doprovodné komentáře.

```
export default class RMMusicianFactory
{
    static musicians = {}

    static getMusician( object )
    {
        if ( !this.musicians[ object.id ] )
        {
            this.musicians[ object.id ] = new RMMusician( object )
        }

        return this.musicians[ object.id ]
    }
}
```

Kód 5.2: Továrna muší váhy na hudebníky

## 5.3 Backend

Backendová část pluginu *Radio Manager* zahrnuje správu rádiových stanic, hudebníků a žánrů v administraci systému WordPress. Tato část byla napro-

gramována pomocí skriptovacího jazyka PHP jako Composer balíček, který pracuje na straně serveru. Při jeho vývoji byly použity vestavěné PHP funkce systému WordPress a dalších API, jako např. Plugin API, Shortcode API či Rewrite API. Backendová část má jediné dvě externí závislosti – WordPress pluginy Advanced Custom Fields (s rozšířením „Repeater Field“) a s2Member Framework. Tyto pluginy byly na webových stránkách zadavatele zavedeny již v minulosti. První slouží pro tvorbu různých administračních prvků, zatímco druhá se používá při kontrole oprávnění posluchačů během zpracování výstražných hlášení. Technologie Advanced Custom Fields navíc přinesla výrazné usnadnění tvorby této části pluginu. Pomocí ní byly pro správu rádiových stanic vytvořeny tyto prvky (v závorce je vždy uveden typ daného prvku):

- nastavení pro rádiovou stanici (typ „Group Field“)
- kolekce pro playlistové položky rádiové stanice (typ „Repeater Field“)
- kolekce pro výstražná hlášení rádiové stanice (typ „Repeater Field“)
- kolekce pro obrázky hudebníka (typ „Repeater Field“)
- kolekce pro úvodní nahrávky hudebníka (typ „Repeater Field“)
- kolekce pro nahrávky hudebníka (typ „Repeater Field“)

Do těchto administračních prvků může administrátor (příp. editor) ukládat všechna potřebná data. Uvedené prvky se nachází u každého příspěvku rádiové stanice či hudebníka. Mezi nastavení konkrétní rádiové stanice patří doba zobrazení jednoho obrázku, webové přískěvky a dále popisky pro jméno hudebníka a název písni.

### 5.3.1 Architektura

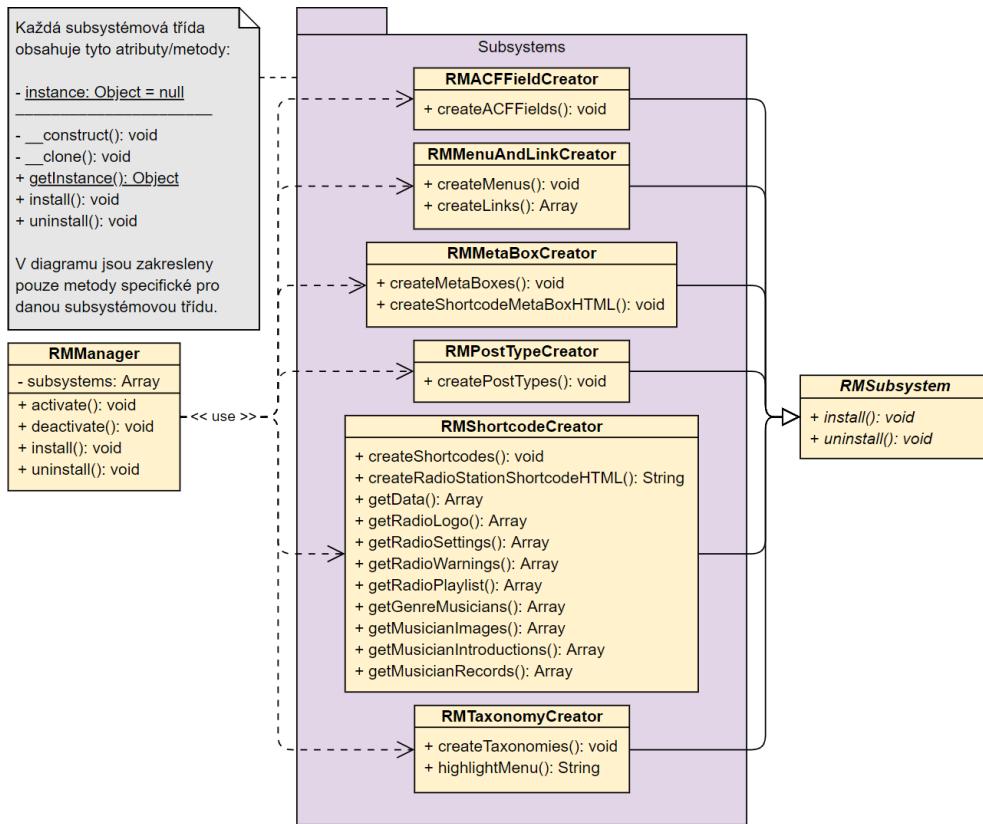
Hlavním souborem pluginu je `radio-manager.php`, ve kterém se nachází základní informace o pluginu pro systém WordPress (v podobě hlavičkového komentáře). Tento soubor je z celého pluginu načten jako první. Jeho kód nejdříve aktivuje automatické nahrávání tříd (tzv. „Autoloading“), kterého je dosaženo pomocí nástroje Composer. Následně zaregistrouje balíčky JS a CSS kódu pro frontendovou a backendovou část. Nakonec definuje potřebné konstanty a vytvoří instanci třídy `RManager`, která inicializuje ostatní subsystémové třídy. Tyto typy tříd zastupují jednotlivé součásti pluginu a nesou zodpovědnost za jejich správnou od/instalaci a funkčnost.

Při tvorbě backendové části byla zvolena objektově orientovaná architektura *Main plugin file, then one or more class files* uvedená mezi Best Practices [8]. Složka s pluginem navíc (s mírnými úpravami) dodržuje doporučovanou adresářovou strukturu. Všechny třídy jsou umístěny ve stejném jmenném prostoru a jejich názvy obsahují unikátní prefix „RM“ (jako zkratka názvu

## 5. IMPLEMENTACE

---

pluginu), čímž je předcházeno případným kolizím. V této architektuře byly zároveň použity i návrhové vzory Facade a Singleton, o kterých se lze dočíst více v následujících podkapitolách.



Obrázek 5.4: Diagram tříd backendové části

V následujícím seznamu jsou stručně popsány všechny třídy uvedené v diagramu. Každá třída má v rámci pluginu určitou zodpovědnost. Třídy vyznačené polem „Subsystems“ značí subsystémy, které představují jednotlivé součásti pluginu.

- **RMManager** je hlavní třídou, která volá a integruje obsah ostatních tříd (subsystémů) do systému WordPress.
- **RMS subsystem** je abstraktní rodičovská třída, která je společná pro všechny subsystémové třídy. Tyto třídy ji dědí a definují požadovanou funkci (metody pro od/instalaci).
- **RMACFFieldCreator** je subsystém zodpovědný za tvorbu administračních prvků pomocí pluginu Advanced Custom Fields.

- **RMMenusAndLinkCreator** je subsystém, který je zodpovědný za tvorbu menu a jeho položek v levém navigačním panelu systému WordPress (pomocí funkcí `add_menu_page()` a `add_submenu_page()`). Zároveň je zodpovědný za tvorbu odkazů, které se zobrazují u pluginu *Radio Manager* ve výpisu všech pluginů v systému (např. vytváří odkaz „Help“, který administrátora přesměruje na uživatelskou příručku).
- **RMMetaBoxCreator** je subsystém, který vytváří administrační box s shortcode dané rádiové stanice (pomocí funkce `add_meta_box()`). Administrátor následně zkopíruje tento shortcode pomocí tlačítka „Copy“ a vloží na webovou stránku, kde se místo něj vykreslí přehrávač. Funkcionalita kopírovacího tlačítka představuje jediný JS kód, který je z pluginu načítaný do backendové části.
- **RMPostTypeCreator** je subsystém, který vytváří příspěvkové typy pro rádiové stanice a hudebníky (pomocí funkce `register_post_type()`). Administrátor (příp. editor) tak může vytvářet, upravovat, příp. mazat jednotlivé příspěvky v rozhraní dodaném systémem WordPress.
- **RMShortcodeCreator** je důležitý subsystém, který zodpovídá za shortcode (způsob, jakým lze rádiové stanice zobrazit na webové stránce). Jeho zodpovědností je vykreslení přehrávače daného rádia a přenos všech dat z backendové do frontendové části.
- **RMTaxonomyCreator** je subsystém zodpovědný za tvorbu taxonomie pro hudební žánry (pomocí funkce `register_taxonomy()`). Administrátor tak může vytvářet, upravovat, příp. mazat jednotlivé žánry v rozhraní dodaném systémem WordPress.

#### 5.3.1.1 Použití návrhového vzoru Facade

Návrhový vzor Facade je jedním ze strukturálních vzorů, které se zabývají uspořádáním tříd v programu. S postupujícím vývojem přibývá množství kódu, tříd, metod a dalších funkcionalit. Celý program se tak velice snadno může stát komplexním. Jeho další rozšiřování může být velmi obtížné. Cílem návrhového vzoru Facade je vytvoření nového zjednodušeného rozhraní (fasády), které umožní snadné používání daného programu a jeho tříd bez nutnosti znát veškeré implementační detaily. [29]

Jako fasáda je implementována třída **RManager**, která obsahuje metody pro aktivaci, deaktivaci, instalaci a odinstalaci daného pluginu. Tato třída si udržuje instance jednotlivých subsystémů, na kterých následně provádí jednotlivé operace. Bez existence fasády by kupříkladu instalace pluginu a jeho součástí představovala tvorbu instancí všech subsystémů a volání jejich vlastních metod ručně. Díky tomuto návrhovému vzoru postačuje vytvořit instanci fasády a zavolat jedinou metodu pro instalaci pluginu.

## 5. IMPLEMENTACE

---

Následující ukázka obsahuje zmíněnou třídu `RMManager`, která slouží jako fasáda. Z ukázky byly odstraněny doprovodné komentáře a vypuštěny nepodstatné části kódů.

```
<?php

class RMManager
{
    private $subsystems;

    public function __construct()
    {
        $this->subsystems =
        [
            RMACFFieldCreator::getInstance(),
            RMMenuAndLinkCreator::getInstance(),
            RMMetaBoxCreator::getInstance(),
            RMPostTypeCreator::getInstance(),
            RMS shortcodeCreator::getInstance(),
            RMTaxonomyCreator::getInstance(),
        ];
    }

    public function activate() { ... }
    public function deactivate() { ... }
    public function install() { ... }
    public function uninstall() { ... }
}

?>
```

Kód 5.3: Fasáda pro snadné řízení subsystémů

### 5.3.1.2 Použití návrhového vzoru Singleton

Návrhový vzor Singleton je jedním z nejjednodušších návrhových vzorů. Patří do skupiny tvořivých vzorů, které se zabývají vytvářením nových objektů. Jeho cílem je omezit počet instancí dané třídy na nejvíše jednu, ke které poskytne přístupový bod. Místo toho, aby třída na různé požadavky odpovídala vytvářením dalších a dalších instancí, je vytvořena a uložena pouze jediná instance, která se v případě potřeby vrátí volajícímu. Tím jsou také šetřeny paměťové prostředky. [29]

Všechny subsystémové třídy jsou implementovány jako jedináčci. Existence více instancí těchto tříd by totiž ze samotné podstaty subsystémů nedávala

smysl. Každý subsystém obsahuje privátní konstruktor a statickou metodu, pomocí které lze získat, příp. vytvořit jeho instanci. Navíc má omezené možnosti klonování. Instance daného subsystému je uložena v příslušném privátém statickém atributu.

Následující ukázka obsahuje třídu `RMPostTypeCreator`, která představuje jedináčka. Z ukázky byly odstraněny doprovodné komentáře a vypuštěny ne-podstatné části kódu.

```
<?php

class RMPostTypeCreator extends RMSubsystem
{
    private static $instance = null;

    private function __construct() {}
    private function __clone() {}

    public static function getInstance()
    {
        if ( self::$instance == null )
        {
            self::$instance = new self();
        }

        return self::$instance;
    }

    public function install() { ... }
    public function uninstall() { ... }

    public function createPostTypes() { ... }
}

?>
```

Kód 5.4: Subsystém pro tvorbu příspěvkových typů

## 5.4 Propojení frontendové a backendové části

Nejdůležitější součástí pluginu *Radio Manager* je správa rádiových stanic, ve které lze přidávat, upravovat a mazat příspěvky jednotlivých rádií. U každého z nich je k dispozici speciální kód (tzv. shortcode), pomocí kterého lze danou rádiovou stanici vložit na webovou stránku. Tam si ji mohou poslechnout

## 5. IMPLEMENTACE

---

i ostatní uživatelé a návštěvníci webu. Shortcode rádiové stanice je krátký kód ve tvaru:

```
[rm_radio_station id=""]
```

Každý shortcode obsahuje atribut `id`. Jedná se o automaticky vygenerovaný číselný identifikátor příspěvku rádiové stanice, který je unikátní v rámci systému WordPress. Administrátor následně vloží daný shortcode do konkrétního příspěvku na webové stránce. Aby byl přehrávač rádiové stanice plně funkční a korektně zobrazený všem posluchačům, musí backendová a frontendová část pluginu provést specifické činnosti.

V rámci backendové části je nejdříve zaregistrován výše uvedený shortcode. Registraci provádí třída `RMShortcodeCreator` pomocí WordPress funkce `add_shortcode()`, která se vykoná po vyvolání hooku `init`. Funkci jsou předány 2 parametry – tag daného shortcode (tj. `rm_radio_station`) a callback, který vrací následující HTML kód:

```
<div data-rm-output-id="></div>
```

Při parsování je shortcode nahrazen tímto HTML kódem, kterému je předán identifikátor příspěvku dané rádiové stanice. Pokud tento identifikátor neodpovídá žádnému takovému příspěvku, je místo shortcode zobrazeno:

**There are no data for this radio station.**

Uvedený callback zároveň získá veškerá data dané rádiové stanice ze správy, kam je dříve uložil administrátor (příp. editor). Ta jsou následně pomocí vestavěné funkce `wp_add_inline_script()` odeslána na frontendovou část, kde se objeví v podobě JS objektu uloženého v proměnné `rmData<cislo>` (název opět obsahuje výše uvedený identifikátor). Tato proměnná je deklarována v hlavičce webu před balíčkem minifikovaného a zkompilovaného JS kódu (s komponentami a logikou rádiových stanic).

V rámci frontendové části je po načtení HTML dokumentu zavolán tzv. IIFE („Immediately Invoked Function Expression“). IIFE vyhledá všechny HTML výstupy, které byly vykresleny místo shortcodes. Podle obsahu datového atributu `data-rm-output-id` je pro každý takový výstup vyhledána korespondující proměnná `rmData<cislo>`, která obsahuje data dané rádiové stanice. Těmito daty je následně inicializována React komponenta `RMRadio`, která je vložena do odpovídajícího HTML výstupu. Po načtení webové stránky je místo shortcode zobrazen přehrávač rádiové stanice. Výhoda takového přístupu spočívá v tom, že na jediné webové stránce může být i více přehrávačů různých rádiových stanic, které pracují nezávisle na sobě.

## 5.5 Nasazení

Plugin *Radio Manager* je v současné době zprovozněný na testovacím a produkčním webu organizace CATV USA. Jeho instalace proběhla pomocí vestavěného rozhraní pro nahrávání pluginů, které je k dispozici přímo v systému WordPress. Veškeré další potřebné aktualizace byly prováděny na příslušném FTP serveru pomocí klienta FileZilla. Samotné nasazení a zprovoznění bylo doprovázeno drobnými kolizemi kaskádových stylů, kdy se CSS vzhledové šablony promítaly do CSS rádiových stanic. Tento problém nastával především u prvků knihovny Bootstrap (např. tlačítek či dialogových oken). Kolize byly vyřešeny dopsáním potřebných protideklarací. Současná vzhledová šablona je zastaralá a v budoucnosti je plánováno její zrenovování.

Od svého prvního nasazení byl plugin pravidelně testován, refaktorován a vylepšován. Ve spolupráci se zadavatelem byly pro americké posluchače vytvořeny následující rádiové stanice – Classical Music Audio Program (původní rádio Classic), Folk Music Audio Program (původní rádio Folk) a nejnovější Cultural Audio Program. První rádiová stanice je zaměřena na českou vážnou hudbu, druhá přehrává české folklorní písničky a ve třetí si lze poslechnout různé podcastové nahrávky s českou tematikou. Specialitou každé z nich je, že si posluchač může při poslechu číst různé zajímavé články přímo v přehrávači (o životech jednotlivých hudebních skladatelů, českých tradicích nebo osobách, které v podcastech hovoří). Každá stránka s rádiovou stanicí obsahuje úvodní poutavý text nad rádiem. Pod ním se následně nachází technické informace a představení moderátorů daného rádia (ve formě rozbalovacích boxů).



# KAPITOLA **6**

---

## Testování

Poslední kapitola se zabývá testováním zhotoveného pluginu *Radio Manager*. Testování bylo zaměřené na správnou funkčnost, kompatibilitu různých prohlížečů a zařízení, prototypy UI, použitelnost a překlady. Na konci kapitoly se nachází shrnutí všech dosažených výsledků.

### 6.1 Funkčnost

Testování funkčnosti probíhalo na pravidelné bázi v průběhu celého vývoje, zejména pak na jeho konci. Spočívalo v kompletním systematickém průchodu celým zdrojovým kódem, během kterého byly vyzkoušeny veškeré funkcionality a možnosti. Zároveň byly ošetřeny různé mezní případy a stanovena omezení při tvorbě rádiových stanic a hudebníků. Následující výčet uvádí příklady těchto restrikcí:

- Playlist rádiové stanice musí obsahovat alespoň jednu položku.
- Každý hudebník musí mít alespoň jednu nahrávku.
- Minimální hodnota pro dobu zobrazení jednoho obrázku je 1 sekunda.
- Každé výstražné hlášení musí obsahovat neprázdnou zprávu.
- Seznam nahrávek hudebníka nesmí obsahovat prázdné položky (stejné omezení platí i pro seznamy obrázků a úvodních nahrávek).

Tato omezení zabraňují vzniku výjimečných situací, které by jinak představovaly nefunkční rádiovou stanici (např. rádio s prázdným playistem nic nepřehrává, i když posluchač kliká na ikonky pro spuštění či zastavení přehrávání). Zároveň představují podporu pro uživatele, kterému pomáhají vytvářet kompletní a kvalitní obsah. Pokud uživatel některé z těchto omezení poruší, je na ně upozorněn při ukládání, příp. publikování daného příspěvku.

## 6. TESTOVÁNÍ

---

Na testování funkčnosti se tedy nahlíželo jak z pohledu tvůrce (v kódu), tak z pohledu uživatele (v administraci). V prvním případě se jednalo o časově náročnou činnost, která však odhalila a vedla k opravám velkého množství chyb. Ve druhém případě se jednalo o prevenci chyb ze strany uživatele, podporu při tvorbě a především zamezení případů, ve kterých by koncový posluchač vyhodnotil rádiovou stanici jako nefunkční.

V NPM balíčku s frontendovou částí je pro základní otestování funkčnosti připravena sada testovacích dat (konkrétně v souboru *index.html* na přiloženém CD). Tuto sadu je možné rozšířit a nakonfigurovat podle vlastních potřeb. Týká se však pouze frontendové části a jako taková nepostačuje pro kompletní otestování funkčnosti.

### 6.2 Kompatibilita prohlížečů a zařízení

K otestování kompatibility prohlížečů a zařízení výrazně přispěly použité moderní technologie. Kompilátor Babel umožňoval přeložit moderní JavaScript (verze ES6 a vyšší) do takové podoby, která je srozumitelná současným i starším prohlížečům. Technologie Browsersync umožňovala spustit vlastní lokální server s rádiovou stanicí, kterou bylo možné vyzkoušet v různých prohlížečích a na různých zařízeních připojených k lokální síti. O pomoc s testováním byli požádáni i další techničtí pomocníci a přispěvatelé organizace CATV USA. Testování bylo zaměřené především na následující aspekty:

- Základní funkcionality přehrávače (včetně výstražných hlášení).
- Vzhled a responzivita přehrávače.
- Reakční rychlosť a načítání (obrázků, videí).

#### 6.2.1 Testované prohlížeče

Následující seznam obsahuje všechny otestované prohlížeče (v závorce je vždy uvedena konkrétní verze):

- Google Chrome (verze 87, 88),
- Mozilla Firefox (verze 85),
- Opera (verze 74),
- Safari (verze 14),
- Microsoft Edge (verze 87, 88),
- Internet Explorer (verze 11).

### 6.2.2 Testovaná zařízení

Následující seznam obsahuje všechna otestovaná zařízení (v závorce je vždy uveden systém, příp. značka daného zařízení):

- Mobilní telefon (Android 6.0, Samsung Galaxy A5)
- Mobilní telefon (iOS 14.4, iPhone 8)
- Mobilní telefon (EMUI 9.1, Huawei Honor 9)
- Mobilní telefon (Android 8.1, Xiaomi Redmi 5)
- Mobilní telefon (Android 9, Sony)
- Tablet (iOS 14.2, iPad 6)
- Notebook (Windows 10 Enterprise, Acer)
- Notebook (Windows 10 Home, Lenovo)
- Notebook (Linux Ubuntu 20, Acer Aspire 3)
- Stolní počítač (Windows 10 Pro)
- Stolní počítač (Windows 10 Home)
- Stolní počítač (Windows 7 Pro)
- Stolní počítač (MacOS Big Sur)

### 6.2.3 Vyskytlé problémy

Problémy s kompatibilitou nastaly pouze u prohlížeče Internet Explorer 11 (dále jen „IE 11“). Tento prohlížeč ohlásil chybu u tzv. „Template Literals“, které v jazyce JavaScript zjednodušují práci s řetězci. Jedná se o nový syntaktický prvek ze standardu ES6, který byl obsažen v kódu balíčku **array-unsort** (pro realizaci shuffle módu přehrávání). Tato závislost tak musela být do komplikace přidána navíc. IE 11 následně ohlásil další problémy (např. chybějící podpora metody **Object.assign()**), které byly vyřešeny zahrnutím potřebných polyfillů (ve formě balíčku **core-js**) do projektu.

U prohlížeče Mozilla Firefox byly opraveny drobné vzhledové odlišnosti. Žádné další chyby nebyly nalezeny. Přehrávač je kompatibilní s výše uvedenými prohlížeči a zařízeními.

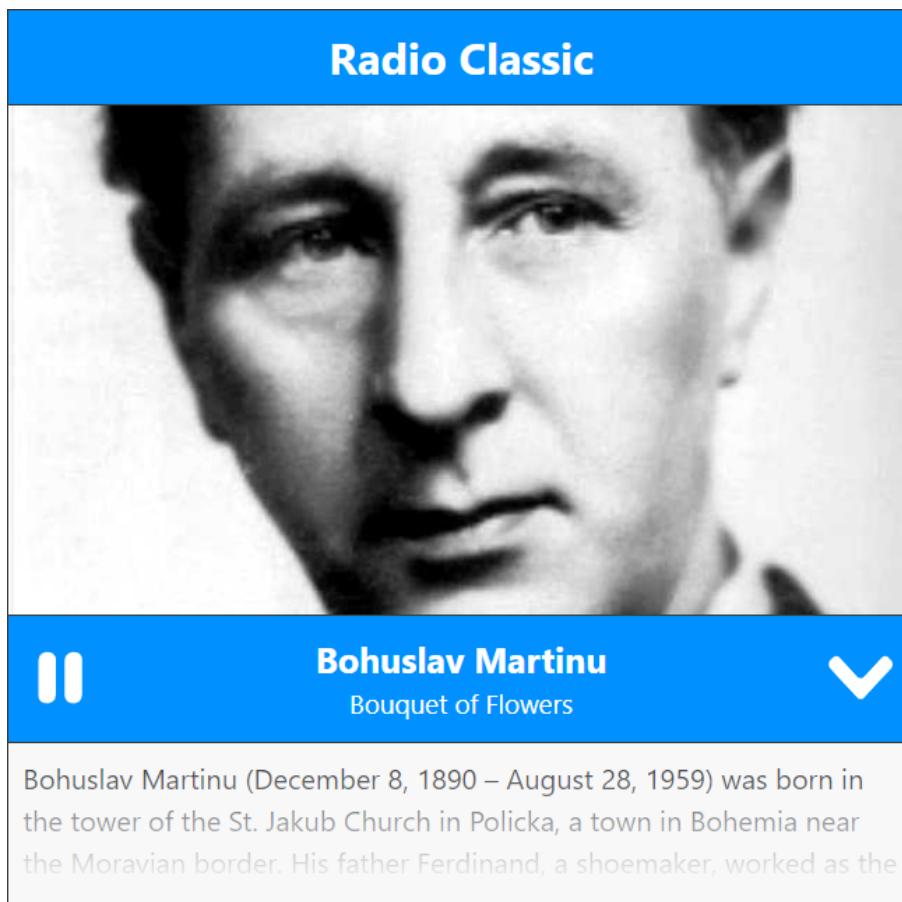
## 6. TESTOVÁNÍ

---

### 6.3 Prototypy

V průběhu tvorby návrhu backendové a frontendové části probíhalo pravidelné testování prototypu UI přehrávače. Na jeho testování se podílely 3 stakeholders (zájmové skupiny): návrhář (autor práce), zadavatel a skupina 2–3 uživatelů (vybraných zadavatelem). Uživatelé byli převážně rodilými mluvčími z USA, budoucími posluchači a zároveň zkušenými návrháři, jejichž poznámky výrazně přispěly ke tvorbě finálního prototypu.

V první verzi prototypu byla fotografie hudebníka vizuálně oddělená od jeho popisu. Mezi témoto částmi se nacházel přehrávač se jménem hudebníka, názvem písni a ovládacími ikonkami. Tento prototyp byl úspěšný pouze v případě rádia Classic. V rádiu Folk se však promítaly webové příspěvky s českými tradicemi, které s názvem skupiny a písni neměly nic společného. Prototyp měl i další nevýhodu v podobě nedostatku informací. Samostatné české názvy (hudebníků, písni) jsou srozumitelné českým uživatelům, ale v případě amerických uživatelů způsobují zmatení (z jazykových důvodů).



Obrázek 6.1: Zachovaná fotografie staršího prototypu

Ve druhé (finální) verzi prototypu byly názvům hudebníka a písni přidány popisky (např. „Music Band:“ či „Song:“). Zároveň bylo odstraněno vizuální oddělení fotografie a textu. Text se zobrazí po kliknutí na fotografii, ke které mohou být navíc připojeny titulky. Změnilo se také uspořádání prvků ve vizuálním přehrávači. Ikonka pro spuštění či zastavení přehrávání je červeně podbarvena pro upoutání pozornosti. O finálním prototypu si lze přečíst více v kapitole 4.2.1. Tento prototyp byl úspěšný i v případě nového rádia Cultural.

Uvedené prototypy se týkaly jen frontendové části. V rámci backendové části byly zadavateli předloženy jen různé náčrty a wireframes, které zobrazovaly logické rozmístění jednotlivých prvků (např. v příspěvku rádiové stanice či hudebníka). Při tvorbě této části byl použit WordPress plugin Advanced Custom Fields, který uvedené prvky dodával.

## 6.4 Použitelnost

Testování použitelnosti proběhlo po dokončení celého pluginu. Jednalo se o nejdůležitější testování, jelikož se při něm pracovalo se skutečnými americkými uživateli. Jeho cílem bylo získat zpětnou vazbu od těchto uživatelů (týkající se zejména jednoduchosti používání, srozumitelnosti či přehlednosti) a podle ní případně upravit plugin.

Testování probíhalo vzdáleně formou Skype videohovorů. Mezi účastníky byl zadavatel a jím vybraní spolupracovníci CATV USA (v celkovém počtu 7 testerů). Většina těchto účastníků nerozuměla programování a neznala WordPress. Mezi testery byli i budoucí uživatelé (celkem 2), kteří se správou rádiových stanic plánují pracovat.

### 6.4.1 Příprava

Pro testování použitelnosti byl určen testovací web CATV USA, na kterém byl zřízen účet **Tester** s administrátorskými právy. Na tento účet se jednotliví testeři přihlašovali. Obsahem testování byly tři testovací scénáře (k dispozici v příloze C), kterými musel každý tester projít. Pracovní prostředí moderátora testů se skládalo z následujících věcí:

- stůl,
- židle,
- hodiny,
- tužka,
- papír na poznámky,
- kamera,

## 6. TESTOVÁNÍ

---

- mikrofon,
- reproduktory,
- PC s testerem,
- notebook s testovacími scénáři.

Samotnému testování použitelnosti předcházela Skype konference, na které byl formou 30 minutové prezentace představen testovaný plugin. Konference se zúčastnil zadavatel a všichni ostatní testeři. Jejím účelem bylo seznámení testerů s autorem této práce, jeho projektem a průběhem testování.

### 6.4.2 Průběh

Testování použitelnosti probíhalo s každým testerem individuálně dle jeho časových možností a technických schopností. Samotný test byl veden v anglickém jazyce a skládal se z následujících úkonů:

1. Představení moderátora.
2. Nasdílení testerovy obrazovky (včetně zapnutí jeho kamery).
3. Přihlášení testera do administrace testovacího webu pod účtem **Tester**.
4. Krátké zopakování celého projektu a průběhu testování.
5. Uklidnění testera.
6. Postupný průchod všemi testovacími scénáři.
7. Závěrečné poděkování, pochvala testera a rozloučení.
8. Návrat testovacího prostředí do původního stavu.

### 6.4.3 Výsledky

Testování použitelnosti bylo úspěšné u všech testerů. Každý test trval min. 30 minut a max. 1 hodinu. 4 ze 7 testerů reagovali opravdu pozitivně (s poukázaním na jednoduchost, příjemnou práci a přehlednost). Zároveň byla získána další data ohledně kompatibility prohlížečů a zařízení od samotných testerů. Testování použitelnosti přineslo následující poznatky:

- Starší uživatelé, kteří neznají WordPress, potřebují větší asistenci, příp. vysvětlení jednotlivých funkcionalit tohoto systému a pluginu.
- Při ukládání příspěvku měli 4 ze 7 testerů tendenci hledat tlačítko pro uložení ve spodní části obrazovky. Toto tlačítko se však ve výchozím nastavení nachází v boxu v pravém postranním panelu. To ovšem ne-predstavuje velký problém, jelikož jej lze přesouvat.

- Při otevřání příspěvku 2 ze 7 testerů nevědělo, zdali mají kliknout na odkaz příspěvku nebo jej zaškrtnout (jako „Checkbox“).
- 1 ze 7 testerů neustále klikal na položku „Radio Manager“ v levém navičním menu, když měl otevřít správu hudebníků (tj. podpoložka, která se objeví po najetí myší na uvedenou položku).
- 1 ze 7 testerů chvíli trvalo, než pochopil nové pojetí playlistu. To samé se stalo i u nastavení výstražných hlášení (volby „First“ a „Step“).
- 1 ze 7 testerů vůbec nepostřehl výstražné hlášení, které se objevilo v průběhu přehrávání a které chvíli před tím sám vyřídil.
- Červeně podbarvená ikonka pro spuštění či zastavení přehrávání se osvědčila u většiny testerů. Jednalo se o první bod, kam směřoval kurzor jejich myší po otevření stránky s rádiovou stanicí.

Na základě těchto poznatků byly vylepšeny textové návodě, které se nachází u všech nastavení ve správě. Úpravy proběhly i v uživatelské příručce, která byla výrazně rozšířena.

Testování se zadavatelem proběhlo vícekrát. Při každém z nich bylo provedeno velké množství drobných úprav. Zadavatel také otestoval instalaci pluginu, kterou zvládl bez jakékoli asistence (i přesto je v uživatelské příručce k dispozici návod). Po určité době používání pluginu přišel od zadavatele další pozitivní ohlas – pracovat se správou se naučil velmi rychle.

## 6.5 Překlady

Plugin je určen pro americké prostředí. Správnost všech textových výstupů tudíž znamenala nejvyšší prioritu. Případné chyby by vedly k nedorozumění a zbytečnému zmatení uživatele. Z tohoto důvodu byl překlad z českého jazyka do americké angličtiny průběžně korekturován dvěma rodilými mluvčími z USA. Mezi zkorekturované výstupy patřily:

- textové výstupy ve frontendové části (zejména materiály dodané zadavatelem, např. životy hudebních skladatelů, titulky fotografií nebo technické informace),
- textové výstupy v backendové části (návodы pro jednotlivá nastavení, popisy jednotlivých boxů),
- uživatelská příručka,
- soubor README v repozitáři s pluginem na GitHub účtu CATV USA (obsahuje informace pro budoucí vývojáře),
- testovací scénáře (pro srozumitelnost výkladu při testování).

## 6.6 Shrnutí výsledků

Výsledkem této práce je nový WordPress plugin *Radio Manager*, který slouží pro tvorbu automatizovaných rádiových stanic a jejich obsahu. Nejdříve byly ve spolupráci s televizním producentem Johnem Honnerem analyzovány požadavky na novou správu rádiových stanic (kap. 3.3). Následně bylo pomocí moderních webových technologií (kap. 2.3) navrhnuto (kap. 4) a implementováno (kap. 5) nové řešení v podobě pluginu do systému WordPress. Ve správě rádiových stanic lze upravovat jednotlivá rádia, jejich playlisty, nastavení, příp. výstražná hlášení (kap. 4.3.1). Ve správě hudebníků lze upravovat jednotlivé hudebníky, jejich popisy, audio/video soubory a obrázky (kap. 4.3.2). Ve správě žánrů lze upravovat jednotlivé žánry, pomocí kterých se hudebníci zařazují do playlistu rádia (kap. 4.3.3). V případě nesnází je k dispozici uživatelská příručka (příloha B) dostupná přímo v administraci. Výstupem každého rádia je responzivní přehrávač, který lze pomocí shortcode (kap. 4.4) vložit na webové stránky. Přehrávač umožňuje přehrávat jednotlivé písni (audio/video soubory), promítat obrázky v podobě slideshow, zobrazovat doprovodné texty a webové příspěvky (kap. 4.2.1). Plugin byl otestován z hlediska správné funknosti, kompatibility, UI, použitelnosti a překladů (kap. 6). V následujícím seznamu je pro větší přehled uvedeno propojení zadavatelových požadavků s kapitolami, ve kterých byly řešeny:

- (F1 – Zobrazování rádiových stanic) Vyřešen v návrhu frontendové části pluginu (kap. 4.2).
- (F2 – Zobrazování výstražných hlášení) Vyřešen v návrhu frontendové části pluginu (kap. 4.2).
- (F3 – Správa hudebníků) Vyřešen v návrhu backendové části pluginu (kap. 4.3).
- (F4 – Správa rádiových stanic) Vyřešen v návrhu backendové části pluginu (kap. 4.3).
- (N1 – Automatizace) Rádiové stanice pracují na straně klienta. Používají jen ta data ze správy, která jsou potřebná k jejich chodu. Tato data jsou předána z backendové na frontendovou část při načtení webové stránky s rádiem (kap. 5.4). Rádiová stanice následně pracuje s těmito daty samostatně a v nepřetržitém provozu. Data ze správy jsou předem připravena. Rádiové stanice jsou tudíž plně automatizované.
- (N2 – Dostupnost) Vyřešen při testování kompatibility různých prohlížečů a zařízení (kap. 6.2).
- (N3 – Použitelnost) Z výsledků testování použitelnosti (kap. 6.4) vyplýnulo, že se testerům pracovalo velmi příjemně. Pozitivně reagoval i sám

## 6.6. Shrnutí výsledků

---

zadavatel. K testování byli vybráni především netechničtí typy testerů. Pro větší podporu uživatele jsou u všech nastavených textové nápovery. V administraci je také k dispozici uživatelská příručka (příloha B).

- (N4 – Jazyková lokalizace) Vyřešen při testování překladů (kap. 6.5).
- (N5 – Architektura) Samotný plugin je modulem, který se skládá z jednotlivých submodulů. Tyto submoduly představují jednotlivé OOP třídy v backendové i frontendové části, příp. znovupoužitelné React komponenty. Každý submodul má v rámci systému určitou zodpovědnost a pracuje samostatně. V architektuře byly použity i návrhové vzory Factory, Flyweight, Facade a Singleton (kap. 5). V celé implementaci se používají prefixované názvy (jako prevence proti kolizím). Bezpečnost pluginu je založena především na adekvátních oprávněních uživatelů (kap. 4.3).
- (N6 – Migrace) Pro ukládání obrázků, úvodních nahrávek a nahrávek je používána knihovna médií (kap. 4.3.2). Pomocí specializovaných WordPress pluginů lze její obsah snadněji migrovat do objektových úložišť.



---

## Závěr

V teoretické části práce byl popsán redakční systém WordPress a tvorba pluginů, které jej rozšiřují o nové možnosti. Dále byl uveden přehled a stručný popis technologií, které byly použity při vývoji kýzeného pluginu. Součástí teoretické části byla také analýza zadavatele a analýza původního a požadovaného stavu rádiových stanic na webu CATV USA.

V praktické části práce bylo na základě provedených analýz navrhnutu řešení v podobě nového zásuvného modulu do systému WordPress. Dále byla popsána implementace tohoto pluginu včetně jejího otestování. Návrh byl vytvořen se zaměřením na UI/UX. Implementace byla zhodovena pomocí moderních webových technologií.

Hlavním cílem této bakalářské práce bylo vytvořit nový plugin do systému WordPress, který usnadní řízení rádií na webu CATV USA. Tento cíl byl splněn v podobě nového WordPress pluginu *Radio Manager*, který řeší veškeré požadavky zadavatele. Jednotlivé dílků cíle (analýza, návrh, implementace, automatizace a testování) byly taktéž splněny. Nabytí jednotlivých požadavků a cílů bylo okomentováno v předešlé podkapitole 6.6. *Radio Manager* je již v provozu na testovacím a produkčním webu CATV USA. Původní implementace rádií Folk a Classic, vestavěná ve vzhledové šabloně, byla odstraněna.

Plugin *Radio Manager* přinesl zcela nové pohledy, možnosti a příležitosti pro CATV USA a její spolupracovníky. V průběhu vypracování této práce a vývoje pluginu se pojetí obou rádií Folk a Classic různě měnilo. Od běžných internetových rádií až po současné hudební programy, které si kladou za cíl svého posluchače vzdělávat v oblasti české hudby. Dále vzniklo nové rádio pojmenované jako „Cultural Audio Program“. Jeho obsahem jsou nahrávky mluveného slova (tzv. „podcasty“), ve kterých se rozmlouvá o zajímavých témaech spojených s Českou republikou (např. české tradice, výuka češtiny atp.). Nahrávky jsou namlované různými hlasateli, kteří mají přístup do administrace webu CATV USA. Pomocí nového pluginu si v rádiu mohou vytvářet své vlastní „shows“ a začít svou potenciální kariéru. CATV USA tak může odstartovat nové projekty s různými uměleckými školami.

## ZÁVĚR

---

Do budoucnosti není plánován další vývoj pluginu *Radio Manager* (s výjimkou refaktorování a případných oprav). Může však být implementována podpora pro novější WordPress editor Gutenberg. Vzhledem k tomu, že CATV USA používá primárně klasický WordPress editor, není tato úprava v současné době vyžadována.

---

## Literatura

1. *Usage Statistics and Market Share of WordPress* [online]. W3Techs. [Cit. 2021-01-25]. Dostupné z: <https://w3techs.com/technologies/details/cm-wordpress>
2. *WordPress / Meetup Pro* [online]. Meetup. [Cit. 2021-03-02]. Dostupné z: <https://www.meetup.com/pro/wordpress/>.
3. *The new Gutenberg editing experience* [online]. WordPress.org. [Cit. 2021-01-25]. Dostupné z: <https://wordpress.org/gutenberg/>.
4. *Roles and Capabilities* [online]. WordPress.org. [Cit. 2021-03-05]. Dostupné z: <https://wordpress.org/support/article/roles-and-capabilities/>.
5. MARSH, J.; MARZÁN, J.; SUCHÁNEK, T. *UX pro začátečníky: (rychlokurz – 100 lekcí)*. Brno: Zoner Press, 2019. 256 s. ISBN 9788074133978.
6. *WordPress Themes* [online]. WordPress.org. [Cit. 2021-01-25]. Dostupné z: <https://wordpress.org/themes/>.
7. *WordPress Plugins* [online]. WordPress.org. [Cit. 2021-01-25]. Dostupné z: <https://wordpress.org/plugins/>.
8. *Plugin Developer Handbook* [online]. WordPress.org. [Cit. 2021-03-02]. Dostupné z: <https://developer.wordpress.org/plugins/>.
9. *Hooks / WordPress Developer Resources* [online]. WordPress.org. [Cit. 2021-03-29]. Dostupné z: <https://developer.wordpress.org/reference/hooks/>.
10. *Functions / WordPress Developer Resources* [online]. WordPress.org. [Cit. 2021-03-02]. Dostupné z: <https://developer.wordpress.org/reference/functions/>.
11. MESSENLEHNER, B. a COLEMAN, J. *Building Web Apps with WordPress, 2nd edition*. Sebastopol, CA: O'Reilly Media, Inc., 2019. 546 s. ISBN 9781491990087.

## LITERATURA

---

12. *Theme Developer Handbook* [online]. WordPress.org. [Cit. 2021-03-04]. Dostupné z: <https://developer.wordpress.org/themes/>.
13. *Custom Fields* [online]. WordPress.org. [Cit. 2021-03-05]. Dostupné z: <https://wordpress.org/support/article/custom-fields/>.
14. *Resources, Documentation, API, How to and Tutorial Articles* [online]. Advanced Custom Fields. [Cit. 2021-01-25]. Dostupné z: <https://www.advancedcustomfields.com/resources/>.
15. *Babel – The compiler for next generation JavaScript* [online]. Babel. [Cit. 2021-03-30]. Dostupné z: <https://babeljs.io/>.
16. *Bootstrap – The most popular HTML, CSS, and JS library in the world* [online]. Bootstrap. [Cit. 2021-03-30]. Dostupné z: <https://getbootstrap.com/>.
17. *Composer – A Dependency Manager for PHP* [online]. Composer. [Cit. 2021-03-30]. Dostupné z: <https://getcomposer.org/>.
18. *Gulp.js* [online]. Gulp. [Cit. 2021-03-30]. Dostupné z: <https://gulpjs.com/>.
19. *Npm* [online]. npm, Inc. [Cit. 2021-03-30]. Dostupné z: <https://www.npmjs.com/>.
20. *React – A JavaScript library for building user interfaces* [online]. Facebook, Inc. [Cit. 2021-03-30]. Dostupné z: <https://reactjs.org/>.
21. *Sass: Syntactically Awesome Style Sheets* [online]. Sass. [Cit. 2021-03-08]. Dostupné z: <https://sass-lang.com/>.
22. *Swiper – The Most Modern Mobile Touch Slider* [online]. Vladimir Kharlampidi. [Cit. 2021-03-30]. Dostupné z: <https://swiperjs.com/>.
23. *CATV USA – obecné informace, cíle, činnosti*. On-line rozhovor s Johnem HONNERREM, současným ředitelem CATV USA. Lány 8. 3. 2020.
24. Oficiální logo organizace CATV USA. In: *CATV USA* [online]. CATV USA, 2020. [Cit. 2021-01-26]. Dostupné z: <https://catvusa.com/about-us/>.
25. *Historie a současnost CATV USA*. On-line rozhovor s Johnem HONNERREM, současným ředitelem CATV USA. Lány 24. 4. 2020.
26. *Rádiové stanice na webu CATV USA*. Telefonický rozhovor s Johnem HONNERREM, současným ředitelem CATV USA. Lány 29. 2. 2020.
27. *Audio Control: Understanding Success Criterion 1.4.2* [online]. World Wide Web Consortium (W3C). [Cit. 2021-03-26]. Dostupné z: <https://www.w3.org/TR/UNDERSTANDING-WCAG20/visual-audio-contrast-dis-audio.html>.
28. OSMANI, A. *Learning JavaScript Design Patterns*. Sebastopol, CA: O'Reilly Media, 2013. 254 s. ISBN 9781449331818.

29. BÖHMER, M. *Návrhové vzory v PHP: 23 vzorových postupů pro rychlejší vývoj*. Brno: Computer Press, 2012. 320 s. ISBN 9788025133385.



## **Seznam použitých zkratek**

**API** Application Programming Interface

**CATV USA** Czech-American TV

**CLI** Command Line Interface

**CMS** Content Management System

**CSS** Cascading Style Sheets

**DOM** Document Object Model

**DRY** Don't Repeat Yourself

**FTP** File Transfer Protocol

**HTML** HyperText Markup Language

**IE** Internet Explorer

**IIFE** Immediately Invoked Function Expression

**JS** JavaScript

**JSX** JavaScript XML

**MySQL** My Structured Query Language

**NPM** Node Package Manager

**OOP** Object-Oriented Programming

**PDF** Portable Document Format

**PHP** PHP: Hypertext Preprocessor

## A. SEZNAM POUŽITÝCH ZKRATEK

---

**SASS** Syntactically Awesome Style Sheets

**SCSS** Sassy CSS

**SPA** Single-Page Application

**UI** User Interface

**URL** Uniform Resource Locator

**UX** User eXperience

**VDOM** Virtual Document Object Model

**WYSIWYG** What You See Is What You Get

**XSS** Cross-Site Scripting

# Uživatelská příručka

Kompletní verze tohoto dokumentu je k dispozici na přiloženém CD.

## B.1 Dependencies

To make the plugin work properly, you must have the following WordPress plugins installed and activated:

- Advanced Custom Fields
- Advanced Custom Fields: Repeater Field
- s2Member Framework

Do not expect 100 % functionality if any of the plugins above is missing. Your website should also have **the Bootstrap library (version 4.6)** installed. Contact your webmaster to get more information about it.

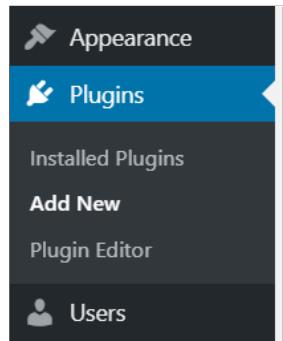
## B.2 Installation

We expect you to be an administrator.

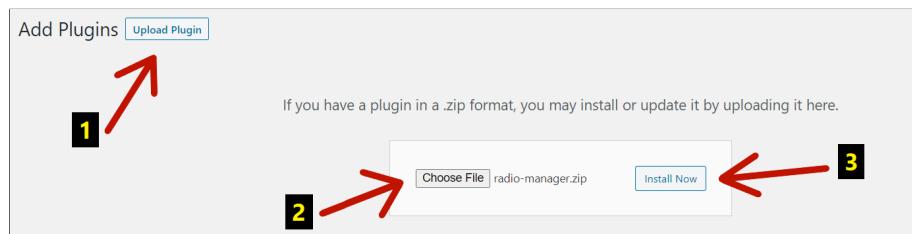
1. Contact the creator (see the last chapter) to get a ZIP archive of the plugin.
2. Log into your WordPress account.
3. Go to *Plugins* → *Add New* in the WordPress navigation menu on the left.

## B. UŽIVATELSKÁ PŘÍRUČKA

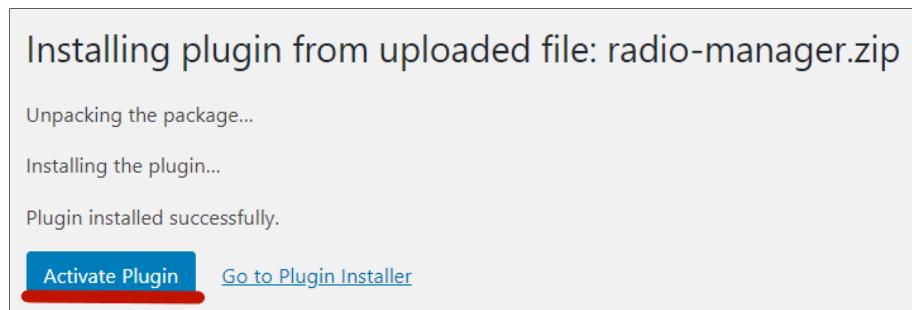
---



4. Click on the *Upload Plugin* button on top of the page. Find the ZIP archive of the plugin in your computer and click on the *Install Now* button to start the installation process.



5. After the installation, activate the plugin by clicking on the *Activate Plugin* button.



6. Find the plugin in *Plugins* → *Installed Plugins* and click on the *Help* link nearby to continue reading this user guide right in the website administration.

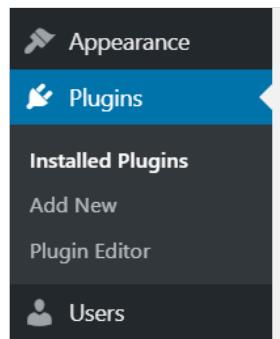


7. You have successfully completed the installation.

## B.3 Uninstallation

We expect you to be an administrator.

1. Log into your WordPress account.
2. Go to *Plugins* → *Installed Plugins* in the WordPress navigation menu on the left.



3. Find the plugin in the list and click on the *Deactivate* link nearby.



4. After that, click on the *Delete* link and confirm the uninstallation.



5. You have successfully completed the uninstallation.

## B.4 PART 1: Basics

The *Radio Manager* WordPress plugin is a software that enables you to create e-radios for your website.

### B.4.1 Features

An e-radio (also “radio”, “radio station” or “jukebox”) can do the following things:

- play audio/video recordings;
- play audio/video introductions (e.g., entries by an e-radio presenter);
- display images (with the top and bottom title of each);
- display other posts from the website;
- show the musician and recording names;
- show the musician description;
- show the e-radio logo;
- show the e-radio name; and
- show dialogs (so-called “warnings”) that interrupt the playback and prompt the listener to take a certain action.

### B.4.2 Different playlist

The **e-radio** has a **playlist**. The playlist is composed of **playlist items** that are being played one after another. But playlist items **ARE NOT** recordings (as one might expect). Every playlist item is a group of music genres you can manage. A **genre** (e.g., *Folk*, *Classical*, *Announcements*, *Jingles*, etc.) is a group of musicians. A **musician** is a group of recordings (and also images, etc.).

When the e-radio starts to play, it sets the first playlist item from the playlist. Then, it sets a music genre from the playlist item and a musician from this genre. Finally, it sets a recording from this musician to be played (and also images to be shown, etc.). Genres, musicians, and recordings are shuffled (see below). The whole process continues for other playlist items. This is the main principle of all e-radios.

### B.4.3 Shuffle mode

*“Shuffle play is a mode of music playback in which songs are played in a randomized order that is decided upon for all tracks at once.” – Wikipedia*

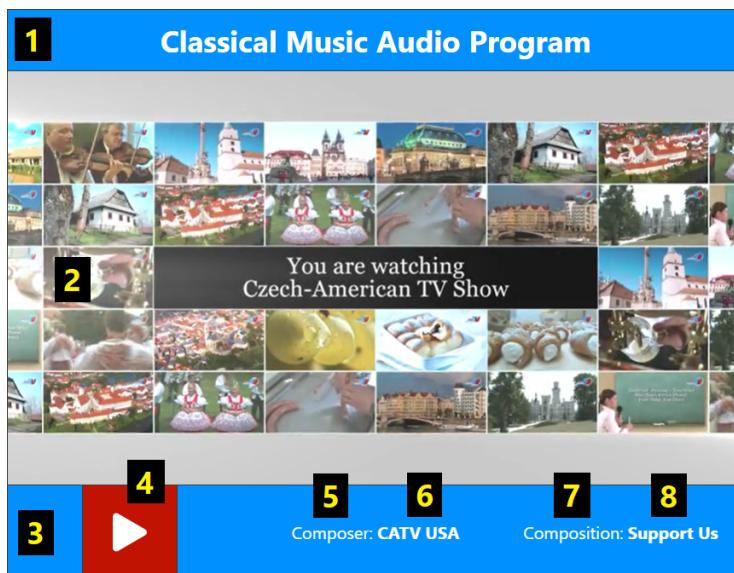
- All the **genres** attached to a particular playlist item are shuffled.
- All the **musicians** attached to a particular music genre are shuffled.
- All the **images**, **introductions**, and **recordings** attached to a particular musician are shuffled.
- If all the genres, musicians, introductions, or recordings were played, they are **shuffled again** (except the images).

## B.5 PART 2: Website

You can insert the e-radio into an arbitrary WordPress post or page (details in the next chapter). After that, there is a blue-styled responsive universal jukebox on the website. The jukebox can do all the things mentioned in the *Features* in the previous chapter. It is also accessible for all your website visitors who can enjoy it. Let us describe the jukebox and all the situations you may face.

### B.5.1 Videos

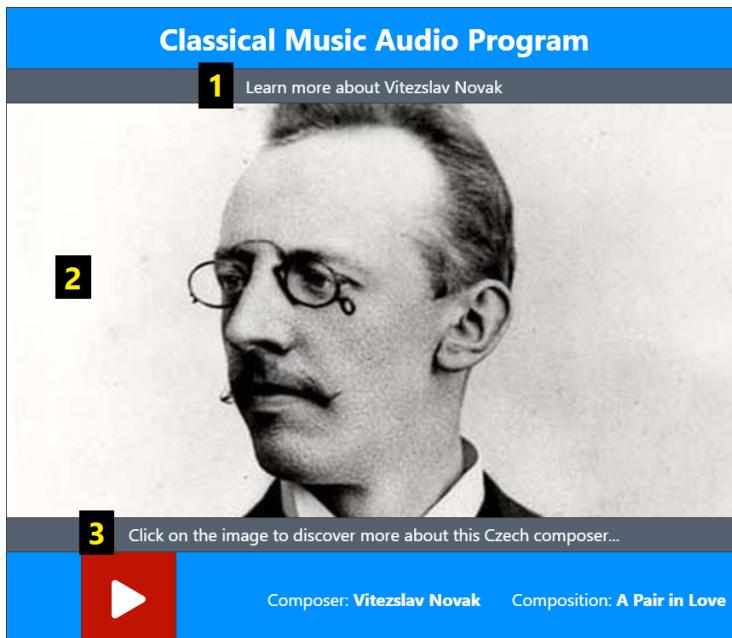
This is the jukebox playing the *Support Us* video recording by the *CATV USA* musician.



1. A field for the **name** of the e-radio.
2. A field for the **multimedia** (the video in this case).
3. A field for the e-radio **controls and information**.
4. A button that can **play or pause** the playback.
5. A **caption** for a musician's name.
6. The **musician's name**.
7. A **caption** for a recording's name.
8. The **recording's name**.

### B.5.2 Images

This is the jukebox playing the *A Pair in Love* audio recording by the *Vitezslav Novak* musician. It also projects images of this musician (even if you stop the playback). Each image is shown for a specific length of time (e.g., three seconds). The transition between images is *fade*.



1. The **top title** of the image.
2. The **image** of the musician.
3. The **bottom title** of the image.

### B.5.3 Description

In the case of *Images*, you can click on the image. After that, the musician's description appears.

## B. UŽIVATELSKÁ PŘÍRUČKA

---

The screenshot shows a mobile application interface for a classical music audio program. At the top, a blue header bar displays the text "Classical Music Audio Program". Below the header, there is a navigation bar with a blue "Back" button on the left and a black button with the number "1" in white on the right. A section titled "Learn more about Vitezslav Novak" follows, containing a paragraph of text about the composer's life and style. Below this, another section titled "Biography - Early years" is present with a detailed paragraph. At the bottom of the screen, there is a control bar featuring a blue "Back" button, a red button with a white play icon, and text indicating the "Composer: Vitezslav Novak" and the "Song: A Pair in Love".

1. With the help of the *Back* buttons, you can return to the image.

### B.5.4 Description (no images)

If the musician has no images, the description appears by default (notice that there are no *Back* buttons). If the description is very long, there is a side slider for scrolling up and down.

This screenshot shows the same mobile application interface as the previous one, but with a longer description of Vitezslav Novak's biography. A vertical grey scroll bar is visible on the right side of the screen, indicating that the text can be scrolled. The rest of the interface elements, including the header, navigation buttons, and bottom control bar, are identical to the first screenshot.

### B.5.5 Logo

If there are no images nor description, there is a logo of the e-radio by default.

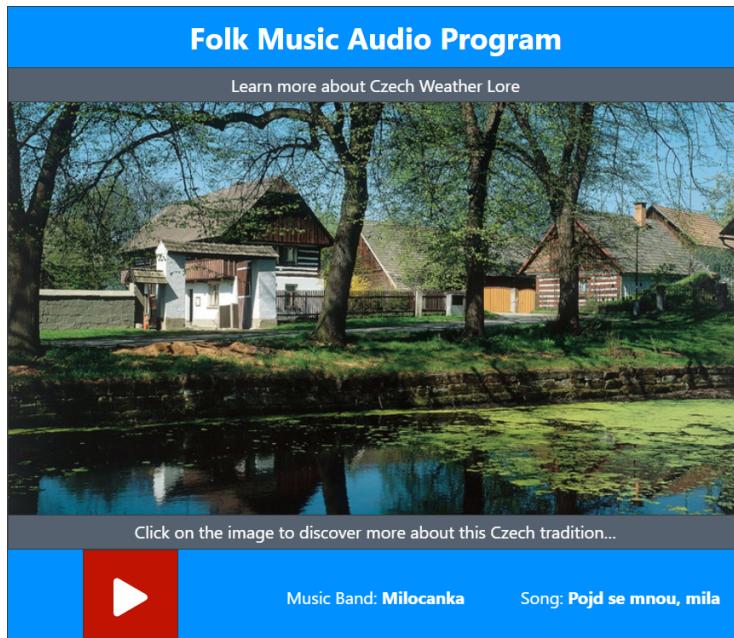


### B.5.6 Website posts

There is the possibility of projecting some website posts in the e-radio. It can be useful if all the musicians have no images nor descriptions. In this case, there is a **featured image** of that website post accompanied by the top and bottom title of that image. If you click on the featured image, there is the **content** of the website post. The website posts are **shuffled**. A particular post is shown for the whole length of the recording. If the next recording starts to play, another website post appears. For instance, there are no images nor descriptions for the music bands in the Folk Music Audio Program. Therefore, you see the website posts about Czech traditions.

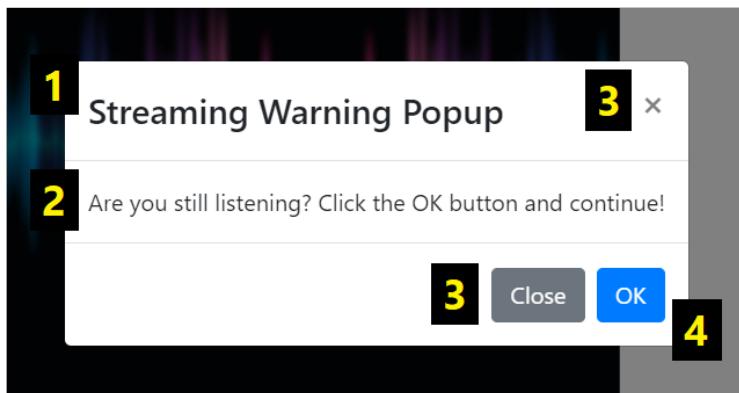
## B. UŽIVATELSKÁ PŘÍRUČKA

---



### B.5.7 Warnings

These dialogs interrupt the playback and require a listener's attention. They can be used for example as a prevention against the loss of streaming or as adverts. Every warning comes with a dark grey background to center the listener's attention onto it.

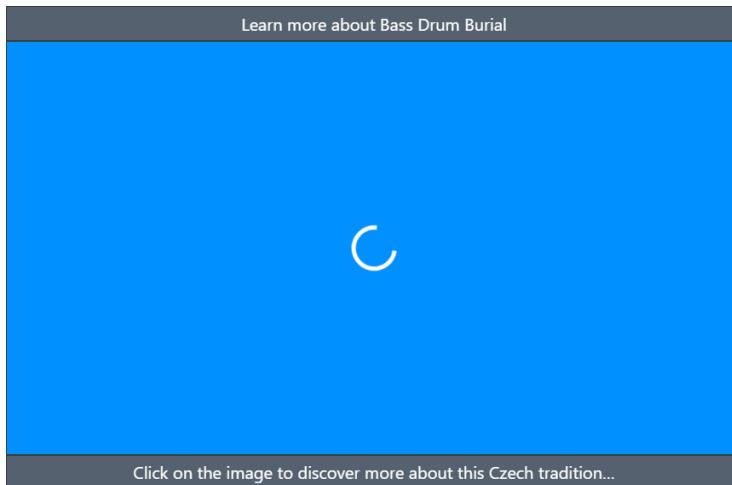


1. The **title** of the warning.
2. The **message** of the warning.
3. The ways to **finish** the warning and continue listening.

4. The *OK* button can finish the warning too, but it can also be used to redirect the listener to a new URL link in a new tab after clicking it.

### B.5.8 Preloader

If there is a high-quality image or video, or your internet connection is slow, there is a small white circle turning around while the multimedia are loading.



### B.5.9 Understand the multimedia rule

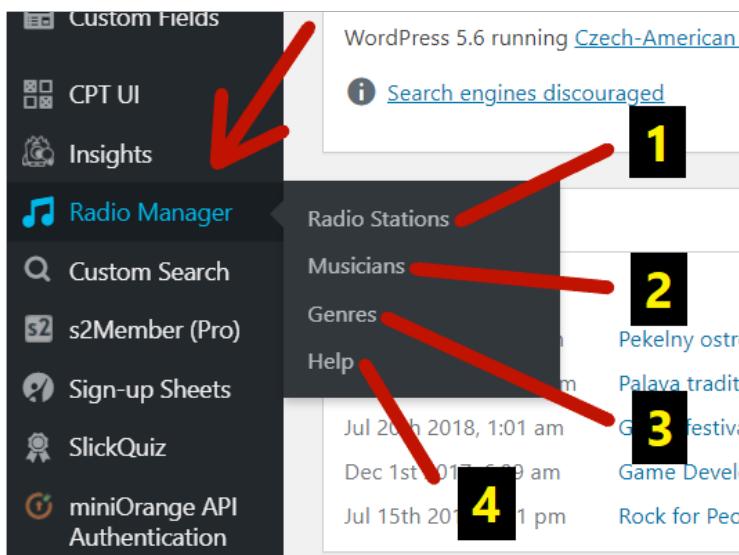
This is the multimedia rule:

**Videos > Website posts > Images > Description > Logo**

- If the e-radio plays a **video** recording, everything to the right is hidden.
- If the e-radio plays an audio recording and the **website posts** are activated, everything to the right is hidden.
- If the e-radio plays the audio recording and the website posts are deactivated, there are **images** attached to a musician and everything to the right is hidden.
- If the musician has no images, there is just a **description** of this musician and everything to the right is hidden.
- If the musician has no images nor description, there is a **logo** of the e-radio.
- If the e-radio has no logo, there is simply **nothing**.

## B.6 PART 3: Administration

You can manage all the e-radios, musicians, and genres in the WordPress administration. First, log into your WordPress account. After that, move the mouse on the *Radio Manager* item in the WordPress navigation menu on the left.



1. There you can manage all the **e-radios**.
2. There you can manage all the **musicians**.
3. There you can manage all the **music genres**.
4. There you can find **this document**.

If you cannot find the *Radio Manager* menu item, you probably do not have the respective user capabilities. You must be able to **edit posts** at least (in the case of this capability, you can see only *Musicians* and *Help* submenu items). Administrators, editors, authors, and contributors can edit posts by default. Moreover, only administrators can also **manage options** and see the rest in the image above.

### B.6.1 Radio stations

After clicking on the *Radio Stations* item in the submenu, you can see a list with all the existing e-radios.

| Featured Image           | Title                    | Date                                   | Expires | (s2) |
|--------------------------|--------------------------|--|---------|------|
| <input type="checkbox"/> | Cultural Audio Program   | Published<br>2021/01/14 at<br>11:28 am | Never   | —    |
| <input type="checkbox"/> | Folk Music Audio Program | Published<br>2021/01/14 at<br>11:24 am | Never   | —    |

- There you can **start creating** a new e-radio.
- There is a **list** of all the existing e-radios that you can manage.
- There you can **search for** an existing e-radio from the list.

### B.6.2 Editing a radio station

After clicking on the specific e-radio or creating a new one, you will land on the editing page. Let us now consider everything this page contains.

#### B.6.2.1 Name

The name is shown in the jukebox on top.

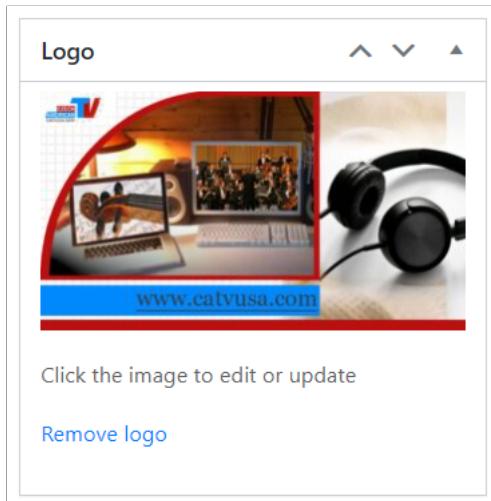


#### B.6.2.2 Logo

The logo is shown in the jukebox by default.

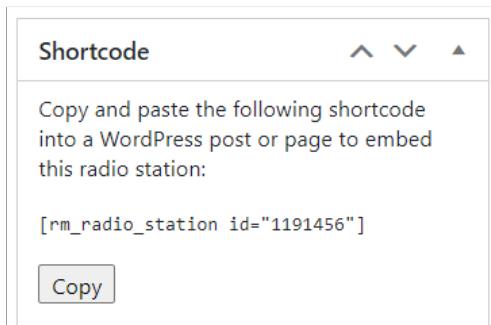
## B. UŽIVATELSKÁ PŘÍRUČKA

---



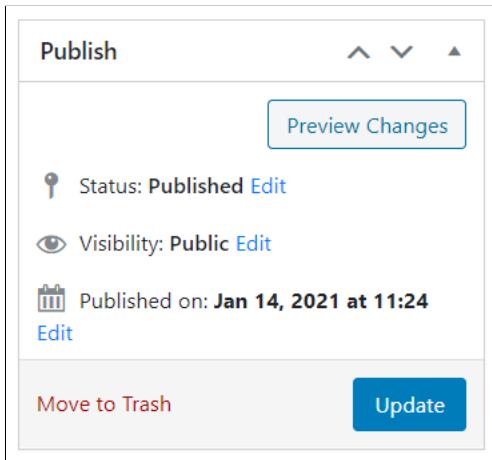
### B.6.2.3 Shortcode

This is the way you can **import a jukebox** into an arbitrary WordPress post or page. Just click on the *Copy* button, find a particular post or page, and insert the shortcode to the content. After opening the post or page on the website, you will see the jukebox.



### B.6.2.4 Publish

Every time you change something, go here to **update** or **publish** all the changes.



### B.6.2.5 Settings

The screenshot shows a 'Settings' interface with the following sections and numbered callouts:

- Image duration**: Set the time for which every image of the musician is displayed when playing. (Callout 1)
- Website posts**: Set all the website posts to be projected when playing. Then enable projecting these posts for the specific playlist items in the playlist. Every website post is displayed for the duration of the recording. (Callout 2)
- Musician caption**: Set a caption for the musician's name displayed in the player.
- Recording caption**: Set a caption for the recording's name displayed in the player.
- Post**: A list of posts including:
  - Brn Majales 2014
  - Festival Let it Roll Open Air
  - Forum 2000 Conference
  - Game Developers Session 2017
  - Game Developers Session in Prague – GDS 2016
  - Arriving of Krakonos Festival
  - Bass Drum Burial
  - Bohemian and Moravian Regional Cuisine
  - Chodsko Area Folk Traditions
  - Czech Gingerbread Traditions
  - Czech Hunting Traditions
- Music Band:** (Callout 3)
- Song:** (Callout 4)

1. *Image duration* – There you can set the time (in seconds) for which every image of the musician is displayed when playing recordings from this musician.
2. *Website posts* – There you can set all the website posts to be projected when playing recordings from the particular musician. But you must enable projecting these posts for the specific playlist items in the playlist below.

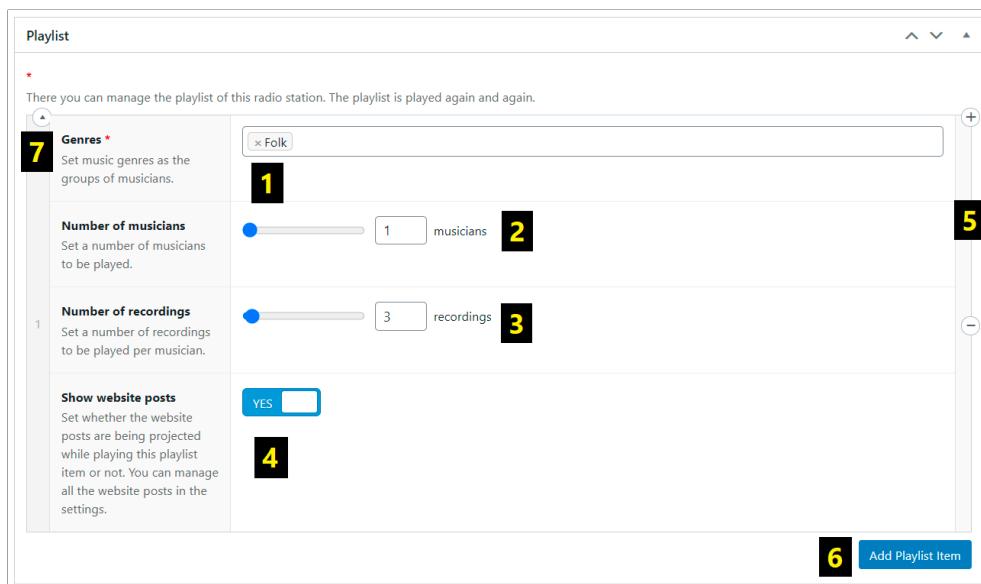
## B. UŽIVATELSKÁ PŘÍRUČKA

---

3. *Musician caption* – There you can set a caption for the musician's name displayed in the jukebox.
4. *Recording caption* – There you can set a caption for the recording's name displayed in the jukebox.

### B.6.2.6 Playlist

There you can add, edit, swap, or delete playlist items from the playlist. There must be at least one playlist item. For each of the playlist items, at least one music genre must be set.



1. *Genres* – There you can set music genres as the groups of musicians.
2. *Number of musicians* – There you can set the number of musicians to be played per genre.
3. *Number of recordings* – There you can set the number of recordings to be played per musician.
4. *Show website posts* – There you can set whether the website posts are being projected while playing this playlist item or not.
5. After moving the mouse on the specific playlist item, you can delete it or even add a new one by clicking on the *Plus* or *Minus* icon on the right.
6. There you can add a new playlist item composed of the previous information – *Genres*, *Number of musicians*, *Number of recordings* and *Show website posts*.

## B.6. PART 3: Administration

7. After moving the mouse on the specific playlist item, you can collapse or expand it by clicking on the *Arrow* icon in the top left corner of the item.

### B.6.2.7 Warnings

There you can add, edit, swap, or delete warnings. For every warning, the message must be set.

| Warnings  |  |
|---|--|
| There you can add, edit, or remove warnings of this radio station. These warnings are being projected while playing.                            |  |
| <b>10</b><br><b>Active</b><br>Set whether this warning shows or not.  | <input checked="" type="checkbox"/> <b>1</b>   |
| <b>Show to</b><br>Set the user group that is supposed to see this warning.  | <input type="text" value="Not registered in the s2Member"/> <b>2</b>   |
| <b>First</b><br>Set the first time when this warning appears (counted per musician).  | <input type="text" value="4"/> <b>3</b> musicians  |
| <b>Step</b><br>Set a regular step between the last time and the next time this warning appears (counted per musician).                          | <input type="text" value="4"/> <b>4</b> musicians  |
| <b>Title</b><br>Set the heading of this warning.  | <input type="text" value="Membership Warning Popup"/> <b>5</b>   |
| <b>Message*</b><br>Set the content of this warning.   | <input type="text" value="Do you want to continue listening without interruptions? Then click the OK button and register! Your membership will support our nonprofit projects. Thank you for your support."/> <b>6</b> |
| <b>Link</b><br>Set the URL link of the confirm button of this warning. If it is empty, then clicking the confirm button will close the warning. | <input type="text" value="http://www.testcatv.site/membership-signup/"/> <b>7</b>  |
| <b>9</b> <input type="button" value="Add Warning"/>   |  |

1. *Active* – There you can set whether this warning shows or not.
2. *Show to* – There you can set the user group that is supposed to see this warning (for example *All users*).
3. *First* – There you can set the first time when this warning will appear (counted per musician).

## B. UŽIVATELSKÁ PŘÍRUČKA

---

4. *Step* – There you can set a regular step between the last time and the next time this warning appears (counted per musician).
5. *Title* – There you can set the heading of this warning.
6. *Message* – There you can set the content of this warning.
7. *Link* – There you can set the URL link of the confirm button of this warning. If it is empty, then clicking on the confirm button will close the warning. If not, the listener will be redirected to the URL link in a new tab after clicking.
8. After moving the mouse on the specific warning, you can delete it or even add a new one by clicking on the *Plus* or *Minus* icon on the right.
9. There you can add a new warning composed of the previous information – *Active*, *Show to*, *First*, *Step*, *Title*, *Message* and *Link*.
10. After moving the mouse on the specific warning, you can collapse or expand it by clicking on the *Arrow* icon in the top left corner of the warning.

### B.6.3 Musicians

After clicking on the *Musicians* item in the submenu, you can see a list with all the existing musicians.

| Featured Image           | Title           | Genre             | Date                                   | Expires | (s2) |
|--------------------------|-----------------|-------------------|--|---------|------|
| <input type="checkbox"/> | (no title)      | Announcements     | Published<br>2021/01/14 at<br>11:02 am | Never   | —    |
| <input type="checkbox"/> | (no title)      | Classical Jingles | Published<br>2021/01/14 at<br>11:00 am | Never   | —    |
| <input type="checkbox"/> | Zatecanka       | Folk              | Published<br>2021/01/14 at<br>10:57 am | Never   | —    |
| <input type="checkbox"/> | Zahorska kapela | Folk              | Published<br>2021/01/14 at<br>10:56 am | Never   | —    |

1. There you can **start creating** a new musician.
2. There is a **list** of all the existing musicians that you can manage.
3. There you can **search for** an existing musician from the list.

#### B.6.4 Editing a musician

After clicking on the specific musician or creating a new one, you will land on the editing page. Let us now consider everything this page contains.

##### B.6.4.1 Name

The name is displayed in the jukebox next to the musician's caption.



##### B.6.4.2 Description

The description appears after clicking on the projected images.

A screenshot of a rich text editor. At the top, there is a toolbar with various icons for media, visual/text mode, and text styling (bold, italic, etc.). Below the toolbar, the text area contains two sections: "Learn more about Vitezslav Novak" and "Biography - Early years".

**Learn more about Vitezslav Novak**  
Vitezslav Augustin Rudolf Novak (1870 – 1949) was a Czech composer and pedagogue. Stylistically, he was part of the neo-romantic tradition, and his music has been occasionally considered an early example of Czech modernism.

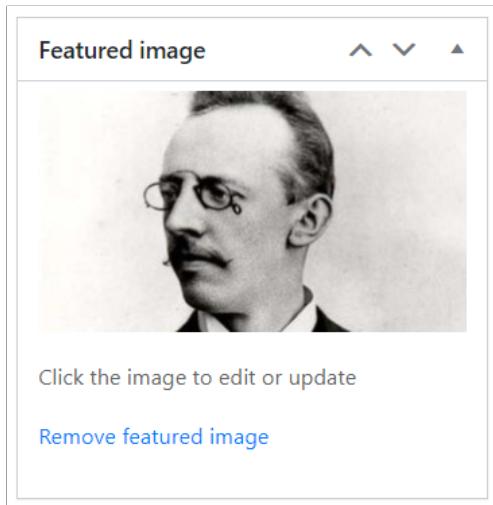
**Biography - Early years**  
Novak (baptized Viktor Novak) was born in Kamenice nad Lipou, a small town in Southern Bohemia. In 1872 the family moved to Pocatky, where Novak first studied the violin with Antonin Silhan and

##### B.6.4.3 Featured image

This image just represents the musician in a list of all the existing musicians. It does not show anywhere in the jukebox.

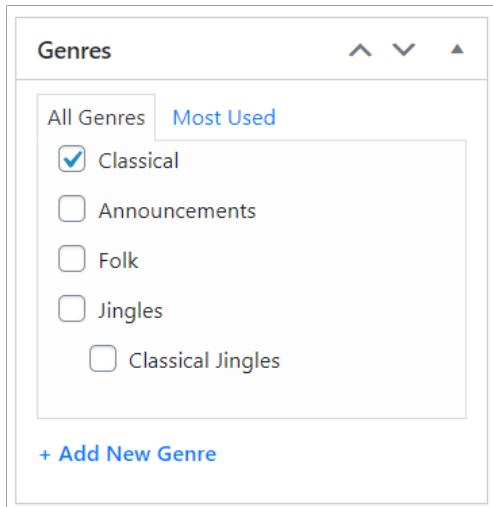
## B. UŽIVATELSKÁ PŘÍRUČKA

---



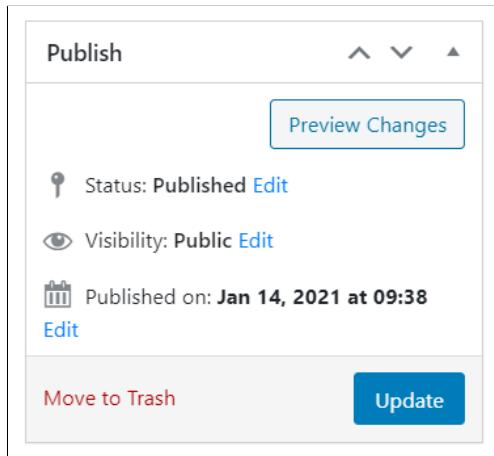
### B.6.4.4 Genres

If you assign the musician to a specific genre, he or she can be played in the playlist.



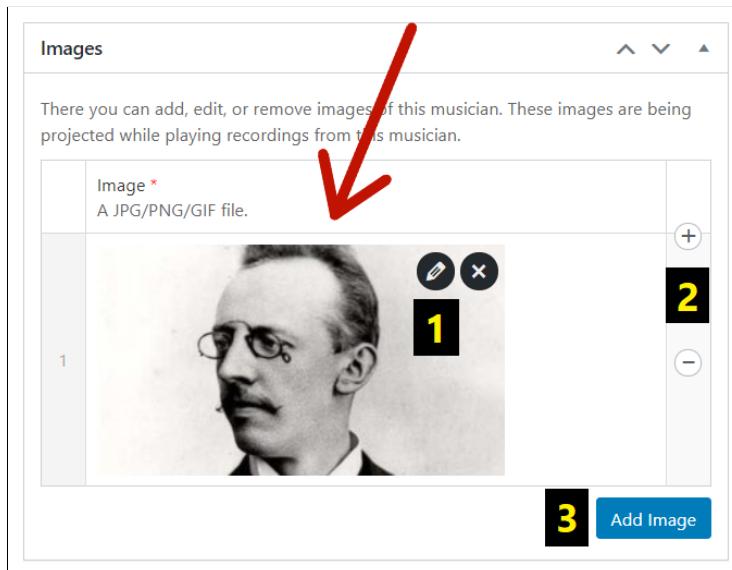
### B.6.4.5 Publish

Every time you change something, go here to **update** or **publish** all the changes.



#### B.6.4.6 Images

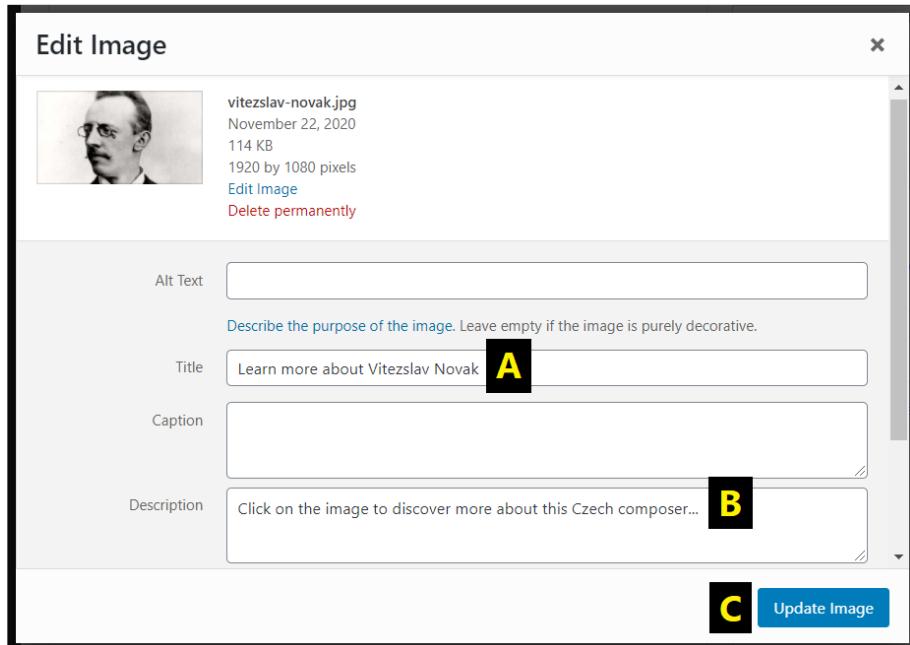
There you can add, edit, or delete images of the musician. The images are being projected while playing recordings from the musician. All the images of the musician should have the **same width and height**. To add, if you set a **top title** or a **bottom title** for one image (see below), it should also be set in the remaining images. If you move the mouse on the image, then the *Pencil*, *Remove*, *Minus* and *Plus* icons appear nearby.



## B. UŽIVATELSKÁ PŘÍRUČKA

---

1. By clicking on the *Pencil* icon, you can change the **top and bottom title** of the image.



- A. There you can edit the **top title** of the image.
  - B. There you can edit the **bottom title** of the image.
  - C. There you can **update** all the changes.
2. By clicking on the *Plus* or *Minus* icon, you can **add or remove** the image.
  3. There you can **add** a new image (from your computer or the Media Library).

### B.6.4.7 Introductions

There you can add, edit, or delete audio/video introductions of the musician. Before playing the recordings of the musician, one of them is set and played. Introductions are **shuffled** and can be used for example as the entries of an e-radio presenter. If you move the mouse on the introduction, then the *Minus* and *Plus* icons appear nearby.

## B.6. PART 3: Administration

The screenshot shows a 'Introductions' section with the following details:

- A header with a 'File name' and 'File size' entry: **File name:** brezovska-desitka-introduction.mp3 **File size:** 96 KB.
- An 'Add Introduction' button at the bottom right.
- Icons for adding (+) and removing (-) items on the right side.
- A large yellow number **1** is placed near the top right corner of the interface.
- A large yellow number **2** is placed near the bottom right corner of the interface.

1. By clicking on the *Plus* or *Minus* icon, you can **add or remove** the introduction.
2. There you can **add** a new introduction (from your computer or the Media Library).

### B.6.4.8 Recordings

There you can add, edit, or delete audio/video recordings of the musician. For every musician, there must be at least one recording. If you move the mouse on the recording, then the *Pencil*, *Remove*, *Minus* and *Plus* icons appear nearby.

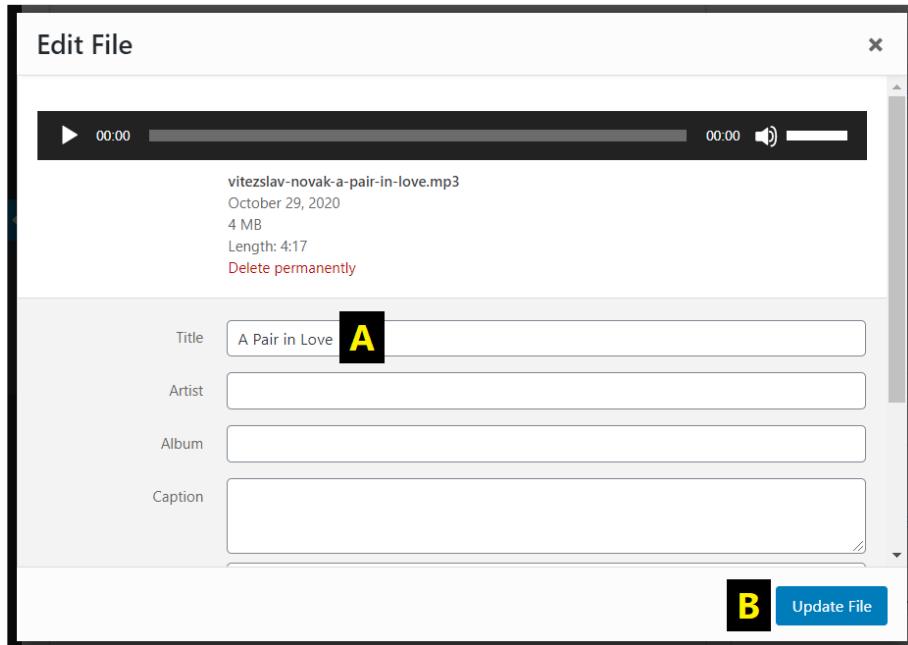
The screenshot shows a 'Recordings' section with the following details:

- A header with a 'File name' and 'File size' entry: **File name:** vitezslav-novak-a-pair-in-love.mp3 **File size:** 4 MB.
- An 'Add Recording' button at the bottom right.
- Icons for editing (pencil), removing (cross), adding (+), and removing (-) on the right side.
- A large red arrow points from the text 'If you move the mouse on the recording, then the Pencil, Remove, Minus and Plus icons appear nearby.' towards the editing icons.
- A large yellow number **1** is placed near the top right corner of the interface.
- A large yellow number **2** is placed near the bottom right corner of the interface.
- A large yellow number **3** is placed near the bottom right corner of the interface.

## B. UŽIVATELSKÁ PŘÍRUČKA

---

1. By clicking on the *Pencil* icon, you can change the **name** of the recording that is displayed next to the recording's caption in the e-radio.



- A. There you can edit the **name** of the recording.
  - B. There you can **update** all the changes.
2. By clicking on the *Plus* or *Minus* icon, you can **add or remove** the recording.
  3. There you can **add** a new recording (from your computer or the Media Library).

### B.6.5 Genres

After clicking on the *Genres* item in the submenu, you can see a list with all the existing music genres. Genres are very important for creating the playlist.

## B.6. PART 3: Administration

The screenshot shows the 'Genres' administration page in WordPress. On the left, there's a form for 'Add New Genre' with fields for 'Name', 'Slug', 'Parent Genre' (set to 'None'), and 'Description'. Below the form, a note says 'The description is not prominent by default; however, some themes may show it.' On the right, there's a list of existing genres with their details:

| Name                | Description | Slug              | Count |
|---------------------|-------------|-------------------|-------|
| Announcements       | —           | announcements     | 1     |
| Classical           | —           | classical         | 13    |
| Folk                | —           | folk              | 21    |
| Jingles             | —           | jingles           | 0     |
| — Classical Jingles | —           | classical_jingles | 1     |

At the top right, there's a search bar labeled 'Search Genres' and a 'Screen Options' dropdown. The number '3' is highlighted in a yellow box above the search bar.

1. There you can **create** a new genre.
2. There is a **list** of all the existing genres that you can manage.
3. There you can **search for** an existing genre from the list.

## B.7 PART 4: Bonus features

### B.7.1 Skipping

By clicking on the bottom right corner of the e-radio, you can skip the current recording/introduction. This hidden feature is very useful if you just want to test the playback without having to wait every time until the recording/introduction is finished.



### B.7.2 Clever images

If you click on the image of the musician in the e-radio, the musician's description appears. Then, if you get back with the help of the *Back* button, there is a **different** image than before. In that moment, images are **shuffled**. It works perfectly only if there are a lot of images to show. Otherwise, you will probably see the same image.

### B.7.3 Extended logo

The logo of an e-radio can have its top and bottom title too.



## B.8 Contact

If you have any troubles or do not understand something, please contact the creator of this plugin.

**Skype:** karelvrabeckv

**E-mail:** karelvrabeckv@gmail.com

Your message helps us to make this user guide and the whole plugin better for understanding.



## Testovací scénáře

### C.1 Tvorba hudebních žánrů

Hudební žánry jsou velice důležité, poněvadž umožňují zařadit větší množství hudebníků do playlistu rádiové stanice. Proto je jejich tvorba velice důležitá a je nutné ji otestovat jako první. Zároveň by tvorba žánrů měla být (dle odhadu) nejjednodušším procesem v administraci. Význam hudebních žánrů je testerovi dopodrobna vysvětlen až v následujících testovacích scénářích, kdy je hudebník k danému žánru přiřazen a kdy je na základě žánrů vytvořen playlist rádiové stanice.

#### Odhadovaný čas

Dvě minuty.

#### Počáteční bod

Tester je přihlášen na testovacím účtu a nachází se na hlavní stránce (tzv. „Dashboard“) v administraci systému WordPress na testovacím webu.

#### Instrukce v českém jazyce

1. Klikněte na položku *Radio Manager → Genres* v levém navigačním menu systému WordPress.
2. Vytvořte jeden hudební žánr – *Czech Classical Music*.

#### Instrukce v anglickém jazyce

1. Click on the *Radio Manager → Genres* item in the WordPress navigation menu on the left.
2. Create a new music genre – *Czech Classical Music*.

## C. TESTOVACÍ SCÉNÁŘE

---

### Koncový bod

Tester vytvořil jeden hudební žánr – *Czech Classical Music*.

### Očekávané kroky

1. Kliknout na položku *Radio Manager* → *Genres* v levém navigačním menu systému WordPress.
2. Vyplnit důležité údaje ve formuláři (tj. *Name*, volitelně i *Slug*, *Parent genre* či *Description*).
3. Kliknout na tlačítko *Add New Genre*.
4. Kroky 2–3 lze opakovat v závislosti na počtu přidávaných hudebních žánrů.

## C.2 Tvorba hudebníků

Z hudebníků je tvořen playlist rádiové stanice. Je proto důležité vyzkoušet jejich tvorbu a ověřit, zdali je vhodně navržena. Tester již musí mít vyzkoušenou tvorbu hudebních žánrů, jelikož hudebníka v tomto scénáři k některému z nich připojí.

### Odhadovaný čas

Pět minut.

### Počáteční bod

Tester je přihlášen na testovacím účtu a nachází se na hlavní stránce (tzv. „Dashboard“) v administraci systému WordPress na testovacím webu.

### Instrukce v českém jazyce

1. Klikněte na položku *Radio Manager* → *Musicians* v levém navigačním menu systému WordPress.
2. Vytvořte nového hudebníka s následujícími informacemi:
  - **Jméno:** Antonin Dvorak
  - **Popis:** One of the most famous Czech composers.
  - **Hlavní obrázek:** (vynechat)
  - **Žánr:** Czech Classical Music
  - **Obrázky:** antonin-dvorak.jpg

- **Úvodní nahrávky:** (vynechat)
  - **Nahrávky:** antonin-dvorak-new-world-symphony.mp3 a další (celkem čtyři)
3. Krok 2 lze opakovat v závislosti na počtu přidávaných hudebníků. Veškeré materiály (obrázky, úvodní nahrávky, nahrávky) jsou již k dispozici v administraci.

### Instrukce v anglickém jazyce

1. Click on the *Radio Manager* → *Musicians* item in the WordPress navigation menu on the left.
2. Create a new musician with the following information:
  - **Name:** Antonin Dvorak
  - **Description:** One of the most famous Czech composers.
  - **Featured image:** (skip)
  - **Genre:** Czech Classical Music
  - **Images:** antonin-dvorak.jpg
  - **Introductions:** (skip)
  - **Recordings:** antonin-dvorak-new-world-symphony.mp3 and so on (four files together)
3. Step 2 can be repeated depending on the number of musicians to be added. All materials (like images, introductions, and recordings) are already available in the website administration.

### Koncový bod

Tester vytvořil hudebníka Antonína Dvořáka.

### Očekávané kroky

1. Kliknout na položku *Radio Manager* → *Musicians* v levém navigačním menu systému WordPress.
2. Dále kliknout na tlačítko *Add New*.
3. Vyplnit „Antonin Dvorak“ do první položky (→ jméno).
4. Vyplnit „One of the most famous Czech composers.“ do druhé položky (→ popis).
5. V boxu *Genres* zaškrtnout „Czech Classical Music“.

## C. TESTOVACÍ SCÉNÁŘE

---

6. V boxu *Images* kliknout na tmavé tlačítko *Add Image*. Po zobrazení nové položky kliknout na světlé tlačítko *Add Image* a v knihovně médií vyhledat např. „antonin dvorak“. Vybrat obrázek/obrázky a kliknout na tlačítko *Select*.
7. V boxu *Recordings* kliknout na tmavé tlačítko *Add Recording*. Po zobrazení nové položky kliknout na světlé tlačítko *Add File* a v knihovně médií vyhledat např. „antonin dvorak“. Vybrat nahrávku/nahrávky a kliknout na tlačítko *Select*.
8. Kliknout na tlačítko *Publish* pro uložení nového hudebníka.
9. Kroky 2–8 lze opakovat v závislosti na počtu přidávaných hudebníků. Navíc může být přidán i hlavní obrázek a úvodní nahrávky (pro tento testovací scénář jsou tyto položky volitelné).

### C.3 Tvorba rádiových stanic

Rádiové stanice jsou základní výstupy pluginu, které lze přidat přímo na webové stránky. Je proto důležité vyzkoušet jejich tvorbu a ověřit, zdali je vhodně navržena. Dále je důležité vyzkoušet zmíněné dosazení na konkrétní webovou stránku. Tester již musí mít vyzkoušenou tvorbu hudebníků a hudebních žánrů, jelikož playlist bude v tomto testovacím scénáři vytvářen právě na základě žánrů a jim přidělených hudebníků.

#### Odhadovaný čas

Pět minut.

#### Počáteční bod

Tester je přihlášen na testovacím účtu a nachází se na hlavní stránce (tzv. „Dashboard“) v administraci systému WordPress na testovacím webu.

#### Instrukce v českém jazyce

1. Klikněte na položku *Radio Manager* → *Radio Stations* v levém navigačním menu systému WordPress.
2. Vytvořte novou rádiovou stanici s následujícími informacemi:

#### Základní informace

- **Název:** Czech Classical Music Radio
- **Logo:** classical-music-audio-program.jpg

### Nastavení

- **Doba zobrazení jednoho obrázku:** (vynechat)
- **Webové příspěvky:** (vynechat)
- **Popisek hudebníka:** Composer:
- **Popisek nahrávky:** Composition:

### Playlist (první položka)

- **Žánry:** Czech Classical Music
- **Počet přehrávaných hudebníků:** 1
- **Počet přehrávaných nahrávek na jednoho hudebníka:** 1
- **Promítat webové příspěvky:** (vynechat)

### Výstražná hlášení (první položka)

- **Aktivní:** Ano
- **Ukázat komu:** Všichni uživatelé
- **Poprvé zobrazit po:** 0 (hudebnících)
- **Pravidelně zobrazovat po:** 1 (hudebníkovi)
- **Titulek:** Streaming Warning Popup
- **Obsah:** Are you still listening? Click the OK button and continue!
- **Link:** (vynechat)

3. Zkopírujte kód v boxu *Shortcode* a vložte jej do libovolné stránky v systému WordPress (např. *Czech Classical Music Program*).
4. Podívejte se na tuto stránku na webu a ověřte, že se rádiová stanice zobrazila na daném místě a pracuje správně.
5. Kroky 2–4 lze opakovat v závislosti na počtu přidávaných rádiových stanic. Veškeré materiály (loga, webové příspěvky, hudebníci, žánry) jsou již k dispozici v administraci (a vytvořeny v předchozích scénářích).

### Instrukce v anglickém jazyce

1. Click on the *Radio Manager → Radio Stations* item in the WordPress navigation menu on the left.
2. Create a new radio station with the following information:

## C. TESTOVACÍ SCÉNÁŘE

---

### Basic information

- **Name:** Czech Classical Music Radio
- **Logo:** classical-music-audio-program.jpg

### Settings

- **Image duration:** (skip)
- **Website posts:** (skip)
- **Musician caption:** Composer:
- **Recording caption:** Composition:

### Playlist (first item)

- **Genres:** Czech Classical Music
- **Number of musicians:** 1
- **Number of recordings:** 1
- **Show website posts:** (skip)

### Warnings (first item)

- **Active:** Yes
- **Show to:** All users
- **First:** 0 (musicians)
- **Step:** 1 (musicians)
- **Title:** Streaming Warning Popup
- **Message:** Are you still listening? Click the OK button and continue!
- **Link:** (skip)

3. Copy the code in the *Shortcode* box and insert it into an arbitrary post or page in the WordPress (e.g. *Czech Classical Music Program*).
4. Look at the post or page on the website and check the functionality of the radio station.
5. Steps 2–4 can be repeated depending on the number of radio stations to be added. All materials (like logos, website posts, musicians, and music genres) are already available in the website administration.

### Koncový bod

Tester vytvořil rádiovou stanici Czech Classical Music Radio, vložil ji na konkrétní webovou stránku a ověřil její funkčnost.

### Očekávané kroky

1. Kliknout na položku *Radio Manager* → *Radio Stations* v levém navigačním menu systému WordPress.
2. Dále kliknout na tlačítko *Add New*.
3. Vyplnit „Czech Classical Music Radio“ do první položky (→ název).
4. V boxu *Logo* kliknout na *Set logo* a v knihovně médií vyhledat např. „classical music audio program“. Vybrat obrázek a kliknout na tlačítko *Set logo*.
5. V boxu *Settings* vyplnit „Composer:“ do položky *Musician caption* a „Composition:“ do položky *Recording caption*.
6. V boxu *Playlist* kliknout na tlačítko *Add Playlist Item*. Po zobrazení nové položky nastavit *Genres* na *Czech Classical Music* a dále *Number of musicians* a *Number of recordings* na 1.
7. V boxu *Warnings* kliknout na tlačítko *Add Warning*. Po zobrazení nové položky nastavit *Active* na *YES*, *Show to* na *All users*, *First* na 0 (hudebníků), *Step* na 1 (hudebníka) a dále vyplnit „Streaming Warning Popup“ do *Title* a „Are you still listening? Click the OK button and continue!“ do *Message*.
8. Kliknout na tlačítko *Publish* pro uložení nové rádiové stanice.
9. V boxu *Shortcode* kliknout na tlačítko *Copy* a zkopiovaný kód vložit do libovolné stránky vytvořené v systému WordPress (např. *Czech Classical Music Program*). Tuto stránku následně uložit, zobrazit v prohlížeči a zkontovalovat funkčnost vložené rádiové stanice.
10. Kroky 2–9 lze opakovat v závislosti na počtu přidávaných rádiových stanic. Zároveň lze opakovat i kroky 6 a 7 v závislosti na počtu přidávaných položek playlistu a výstražných hlášení. Navíc může být nastavena i doba zobrazení jednoho obrázku, webové příspěvky a jejich promítání u konkrétních položek playlistu a dále link tlačítka *OK* u výstražných hlášení (pro tento testovací scénář jsou tyto položky volitelné).



PŘÍLOHA **D**

---

## Obsah přiloženého CD

```
readme.txt ..... stručný popis obsahu CD
└── backend.rp ..... wireframes backendové části ve formátu RP
└── frontend.rp ..... prototyp frontendové části ve formátu RP
└── radio-manager ..... zdrojové kódy implementace
└── thesis.pdf ..... text práce ve formátu PDF
└── thesis.tex ..... zdrojová forma práce ve formátu LATEX
└── user-guide.pdf ..... uživatelská příručka ve formátu PDF
```