

UNIVERZITA PARDUBICE

FAKULTA ELEKTROTECHNIKY A INFORMATIKY

DIPLOMOVÁ PRÁCE

2020

Bc. David Vaníček

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Webová aplikace pro hudebníky

David Vaníček

Diplomová práce

2020

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2019/2020

ZADÁNÍ DIPLOMOVÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Bc. David Vaníček**
Osobní číslo: **I17226**
Studijní program: **N2646 Informační technologie**
Studijní obor: **Informační technologie**
Téma práce: **Webová aplikace pro hudebníky**
Zadávající katedra: **Katedra softwarových technologií**

Zásady pro vypracování

Cílem práce je vytvoření funkční webové aplikace. Webová aplikace bude rozdělena do kategorií interpreti, příspěvky a články. Možnost vytváření jednotlivého obsahu bude rozdělena pomocí uživatelských rolí, již vytvořený obsah bude možné vyhledat a ohodnotit číselným hodnocením nebo psaným textem. Webová aplikace také bude obsahovat komunikační rozhraní pro zpracování zpráv mezi uživateli a notifikační rozhraní pro upozornění na nevyřízené události.

Práce bude obsahovat: rešerši systémů, které se zabývají touto problematikou, analýzu procesů, které budou znázorněny v procesním BPMN diagramu s cílem dosažení maximální bezpečnosti aplikace, popis použitých technologií, návrh databáze a ER diagram s využitím „Crow's Foot“ notace entity-relationship.

Pro vytvoření aplikace bude využita databáze MySQL nebo Oracle a technologie ASP.NET MVC s možností využití dalších technologií pro podporu dynamičnosti (např. JavaScript).

Rozsah pracovní zprávy: **50-60 stran**
Rozsah grafických prací: **-**
Forma zpracování diplomové práce: **tištěná**

Seznam doporučené literatury:

ŘEPA, Václav. Analýza a návrh informačních systémů. 1. vyd. Praha : Ekopress, 1999. 403 s. ISBN 80-86119-13-0.
GROFF, James R. a Paul N. WEINBERG. SQL kompletní průvodce. Brno: Computer Press a.s., 2005. 936 s. ISBN 80-251-0369-2.
BRYLA, Bob a Kevin LONEY. Mistrovství v Oracle Database 10g. Brno: Computer Press a.s., 2006. 700 s. ISBN 80-251-1277-2.
GALLOWAY, Jon, Brad WILSON, K. Scott ALLEN a David MATSON. Professional ASP.NET MVC 5. Indianapolis, IN: Wrox, a Wiley brand, [2014]. Wrox professional guides. ISBN 11-187-9475-3.
LERMAN, Julia. Programming Entity framework. 2nd ed. Sebastopol: O'Reilly, c2010. ISBN 05-968-0726-0.
DUCKETT, Jon, Gilles RUPPERT a Jack MOORE. : interactive front-end web development. Indianapolis, IN: Wiley, [2014]. ISBN 11-185-3164-7.

Vedoucí diplomové práce: **Ing. Miloslav Macháček, Ph.D.**
Katedra informačních technologií

Datum zadání diplomové práce: **5. listopadu 2019**
Termín odevzdání diplomové práce: **15. května 2020**



Ing. Zdeněk Němec, Ph.D.
děkan

prof. Ing. Antonín Kavička, Ph.D.
vedoucí katedry

V Pardubicích dne 15. listopadu 2019

Prohlášení autora:

Prohlašuji, že jsem tuto práci vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne: 15.5.2020

David Vaníček

PODĚKOVÁNÍ

Děkuji vedoucímu mé diplomové práce, panu Ing. Miloslavu Macháčkovi, Ph.D. za cenné rady během vypracování této práce. Dále bych rád poděkoval své rodině, který mi byla podporou během studia.

ANOTACE

Diplomová práce se zabývá analýzou nedostatků doposud vytvořených webových aplikací obsahující materiál pro hudebníky různých úrovní. Mezi hlavní cíle praktické části patří vytvoření webové aplikace pomocí technologie ASP.NET MVC odstraňující nedostatky a vylepšení logické vrstvy.

KLÍČOVÁ SLOVA

ASP.NET MVC, Entity Framework, Database, ORM

TITLE

Web application for musicians.

ANNOTATION

Diploma thesis deals with shortcomings analysis for existing web applications containing contributions for musicians of different levels. The main objectives of the practical part are creating a web application using ASP.NET MVC technology to eliminate shortcomings and improve the logical layer.

KEYWORDS

ASP.NET MVC, Entity Framework, Database, ORM

OBSAH

ÚVOD	16
1 Webová aplikace	17
1.1 Výhody webové aplikace	17
1.2 Nevýhody webové aplikace	17
2 Aplikační server	17
2.1 HTTP	18
2.2 HTTPS	18
2.2.1 SSL	18
2.3 Životnost požadavku webové aplikace	18
3 Porovnání webových aplikací pro hudebníky	18
3.1 Supermusic	19
3.2 Pisnický-akordy	24
3.3 Ultimate-guitar	27
3.3.1 Guitar Pro	29
4 Vyhodnocení webových aplikací	30
5 Použité technologie	31
5.1 ASP.NET MVC	31
5.1.1 Programovací jazyk C#	31
5.1.2 MVC	31
5.1.3 Ostatní způsoby využití ASP.NET	33
5.2 Entity Framework	33
5.2.1 Model First	33
5.2.2 Database First	34
5.2.3 Code First	35
5.2.4 LINQ	35
5.2.5 Migrace	36

5.3	Technologie pro oživení View	36
5.3.1	JavaScript	36
5.3.2	jQuery	36
5.3.3	AJAX	36
5.4	MYSQL	37
5.5	SQL Server	37
5.6	Bootstrap	38
5.7	SendGrid	38
5.8	GitHub	39
5.9	Struktura projektu ASP.NET	40
6	Alternativní technologie	41
6.1	Java	41
6.2	Hibernate	41
6.2.1	Mapovací soubory	41
6.2.2	Anotace	42
6.3	Vytvoření projektu	42
6.4	Porovnání implementačních technologií	46
7	Bezpečnost aplikace a databáze	47
7.1	SQL injection	47
7.2	Cross-site scripting	47
7.3	Odepření služby	47
7.4	Šifrování přenášených dat	48
7.5	Nedostatečná autentizace	48
7.6	Autorizace	48
8	BPMN	49
8.1	Tokové objekty	49
8.2	Plavecké dráhy	49

8.3	Spojovací objekty	49
8.4	Artefakty	49
8.5	Ukázka BPMN	49
8.5.1	Přihlášení uživatele	49
8.5.2	Prohlížení příspěvků	50
8.5.3	Vytvoření příspěvku	52
8.6	Změna stavu požadavku o příspěvek	53
9	Základy hudební teorie	54
9.1	Tóny	54
9.2	Akordy	54
9.3	Tabulatury	56
10	analýza projektu	58
10.1	Požadavky	58
10.2	Aktéři	62
10.2.1	Nepřihlášený uživatel	63
10.2.2	Přihlášený uživatel	63
10.2.3	Správce interpreta	63
10.2.4	Redaktor	63
10.2.5	Administrátor	63
10.3	Případy užití	63
11	Vytvoření projektu	68
12	Databázový model	77
12.1	ER model	77
12.2	Entity Framework Identity	77
12.3	Uživatelská část	78
13	Nasazení webové aplikace	81
13.1	Publikování projektu ASP pomocí programu Visual Studio	81

13.1.1	FTP.....	81
13.1.2	Lokální adresář	81
14	Uživatelská příručka	83
14.1	Registrace	83
14.2	Zapomenuté heslo.....	84
14.3	Přihlášení	85
14.4	Interpreti	86
14.5	Příspěvky	90
14.6	Články.....	91
14.7	Vyhledání uživatele.....	94
14.8	Implementovaný systém zpráv	94
14.9	Profil uživatele.....	94
14.10	Vlastní příspěvky.....	95
14.11	Správa žádosti.....	96
14.12	Správa interpretů	98
14.13	Často kladené otázky	100
14.14	Podpora.....	100
	Závěr	101
	SEZNAM POUŽITÉ LITERATURY	102

SEZNAM OBRÁZKŮ

Obrázek 1 - Supermusic.cz domovská stránka	19
Obrázek 2 - Supermusic.cz vyhledání interpreta	20
Obrázek 3 - Supermusic.cz nové příspěvky	21
Obrázek 4 - Supermusic.cz konkrétní interpret	21
Obrázek 5 - Supermusic.cz příklad výpisu písní interpreta	22
Obrázek 6 - Supermusic.cz příklad příspěvku	23
Obrázek 7 - Supermusic.cz vlastní příspěvky	23
Obrázek 8 - Supermusic.cz zobrazení profilu ostatních uživatelů	24
Obrázek 9 - Pisnickyy-akordy.cz domovská stránka	24
Obrázek 10 - Pisnickyy-akordy.cz ukázka výpisu vyhledání interpreta	25
Obrázek 11 - Pisnickyy-akordy.cz výpis písní	26
Obrázek 12 - Pisnickyy-akordy.cz vyhledaný akord	27
Obrázek 13 - Ultimate-guitar domovská stránka	27
Obrázek 14 - Ultimate-guitar výpis písní	28
Obrázek 15 - Ultimate-guitar ukázka správného formátu	29
Obrázek 16 - Guitar Pro ukázka	29
Obrázek 17 - ASP.NET MVC ukázka modelu s anotacemi	32
Obrázek 18 - ASP.NET MVC ukázka view s Razor syntaxí	32
Obrázek 19 - EDMX model a SQL skript ve Visual Studiu	34
Obrázek 20 - příklad modelové třídy	35
Obrázek 21 - Bootstrap styly ve Visual Studiu	38
Obrázek 22 - GitHub přehled provedených změn	39
Obrázek 23 - použití GitHub pomocí Visual Studia	40
Obrázek 24 - Hibernate mapovací soubory	42
Obrázek 25 - Hibernate anotace	42
Obrázek 26 - vytvoření webové aplikace pomocí NetBeans IDE	43
Obrázek 27 - vložení knihoven do vytvořeného projektu	43
Obrázek 28 - konfigurační soubor Hibernate (GUI)	44
Obrázek 29 - konfigurační soubor Hibernate (XML)	44
Obrázek 30 - Hibernate dialog	45
Obrázek 31 - manipulace s daty pomocí HQL	45
Obrázek 32 - model přihlášení uživatele	50

Obrázek 33 - model prohlížení příspěvků.....	51
Obrázek 34 - model vytvoření příspěvku	52
Obrázek 35 - model změny stavu požadavku	53
Obrázek 36 - využití znalosti tónů v případě kytary.....	55
Obrázek 37 - využití znalosti tónů v případě kláves.....	56
Obrázek 38 - příklad prázdné tabulatury	56
Obrázek 39 - příklad jednoduché tabulatury.....	57
Obrázek 40 - příklad složitější tabulatury	57
Obrázek 41 - případ užití – uživatelé.....	64
Obrázek 42 - případy užití – žádosti	65
Obrázek 43 - případy užití – příspěvky	66
Obrázek 44 - případy užití – články.....	67
Obrázek 45 - vytvoření projektu	68
Obrázek 46 - výběr typu aplikace	69
Obrázek 47 - import Entity Frameworku do projektu	69
Obrázek 48 - přidání modelu (.edmx).....	70
Obrázek 49 - výsledný model webové aplikace	71
Obrázek 50 - připojení k databázi určené pro spuštění SQL skriptu	72
Obrázek 51 - přidání DbContext Generatoru do projektu	73
Obrázek 52 - třída ModelContainer (DbContext).....	73
Obrázek 53 - vygenerována modelová třída	74
Obrázek 54 - modelová třída Článek doplněna o anotace	75
Obrázek 55 – struktura ve Visual Studiu (modelové třídy, skript, model).....	76
Obrázek 56 - Entity Framework kontext	78
Obrázek 57 - kontext uživatelské části	80
Obrázek 58 - publikace pomocí FTP serveru	81
Obrázek 59 - přenos pomocí FTP klienta WinSCP	82
Obrázek 60 - registrace uživatele.....	83
Obrázek 61 - oznámení úspěšné registrace.....	84
Obrázek 62 - potvrzení emailové adresy	84
Obrázek 63 - obnovení hesla	85
Obrázek 64 - přihlášení uživatele	85
Obrázek 65 - domovská stránka.....	86
Obrázek 66 - výchozí výpis interpretů.....	86

Obrázek 67 – příklad vyhledání interpreta	87
Obrázek 68 – názvy příspěvků interpreta	88
Obrázek 69 - příspěvky interpreta s vybraným názvem písně.....	88
Obrázek 70 - obsah příspěvku	89
Obrázek 71 - vložení příspěvku	89
Obrázek 72 - pravidla pro přidání příspěvku	90
Obrázek 73 - nejnovější příspěvky	90
Obrázek 74 - výpis článků	91
Obrázek 75 - články z pohledu redaktora	92
Obrázek 76 - vytvoření nového článku.....	92
Obrázek 77 - zobrazení konkrétního článku	93
Obrázek 78 - diskuze u článku.....	93
Obrázek 79 - vyhledání uživatele	94
Obrázek 80 - implementovaný systém zpráv.....	94
Obrázek 81 - profil uživatele	95
Obrázek 82 - vlastní příspěvky	95
Obrázek 83 – editace písně	96
Obrázek 84 - přijaté žádosti	96
Obrázek 85 - odeslané žádosti	97
Obrázek 86 - zobrazení konkrétních žádostí.....	98
Obrázek 87 - zobrazení konkrétní odeslané žádosti	98
Obrázek 88 - správa interpretů.....	99
Obrázek 89 - aktualizace správců interpreta.....	99
Obrázek 90 - často kladené otázky (FAQ).....	100
Obrázek 91 - kontaktní formulář	100

SEZNAM TABULEK

Tabulka 1 - porovnání webových aplikací.....	30
Tabulka 2 - výpis tónů	54
Tabulka 3 - frekvence základního ladění kytary.....	54
Tabulka 4 - porovnání frekvencí [Hz] mezi tóny a oktávami.....	54
Tabulka 5 - příklady vyhledání dur akordů pomocí masky 0-4-7	55
Tabulka 6 - příklady vyhledání moll akordů pomocí masky 0-3-7	55

Tabulka 7 - seznam požadavků.....	62
Tabulka 8 - atributy modelové třídy Článek.....	74
Tabulka 9 - anotace modelové třídy Článek.....	75

ÚVOD

V dnešní době, ve které je internet každodenní součástí většiny populace, ať z důvodu velkého počtu možností míst pro připojení (domov, pracovní prostředí, venkovní svět), způsobů připojení (Wi-Fi, mobilní data), ale i širokou škálou možností samotných mobilních zařízení. Dostupnost připojení k internetu nabízí prostor využití webovým aplikacím, které postupně nahrazují běžné desktopové verze programů. Cílem této práce je využít dostupnost připojení k zjednodušení práce začínajícím hudebníkům, ale i možnost pokročilým podělit se o své zkušenosti se začátečníky.

Teoretická část obsahuje popis webové aplikace a její výhody, porovnání s již existujícími webovými aplikacemi, popis použitých technologií, vysvětlení základních pojmů z oblasti hudební teorie pro kompletní pochopení diplomové práce.

Praktická část je složena z popisu funkčních a nefunkčních požadavků, základních případů užití, objektově relačního modelu vytvořeném v programu Microsoft Visual Studio a popisu jednotlivých částí webové aplikace, který může sloužit jako uživatelská příručka.

Webová aplikace je vytvořena pomocí technologie ASP MVC, objektově relační mapování pomocí Entity Framework, jQuery, a AJAX. Trvalé uložení dat je dosaženo pomocí databáze MySQL.

1 WEBOVÁ APLIKACE

Webová aplikace je taková aplikace, kterou není nutné instalovat na koncové zařízení uživatele (například počítač, tablet, smartphone) a může být spuštěna ze zařízení pomocí webového prohlížeče. Webová aplikace je spuštěna na straně serveru, webový prohlížeč výslednou stránku pouze zobrazí.^[1]

Webová aplikace je na první pohled podobná webové stránce, ve skutečnosti jde ale o složitější aplikaci, která provádí náročné úkoly (například vyhledávání dat v databázi).^[1]

1.1 Výhody webové aplikace

Mezi základní výhody webové aplikace patří jednoduché použití pro uživatele. Webové aplikace postupně nahrazují aplikace počítačové, které vyžadují instalaci a aktualizace. Instalace není potřebným krokem pro používání z důvodu aplikačního serveru, na kterém je webová aplikace spuštěna. V případě aktualizace tedy stačí nahrát novější verzi na aplikační server, který se již postará o zobrazení nové verze uživateli. Data uživatele jsou také uložena a zálohována na straně serveru a přístup k nim (většinou po patřičné autorizaci) je možný odkudkoliv.^[1]

1.2 Nevýhody webové aplikace

Mezi nevýhody webové aplikace patří nezbytné připojení k internetu, tento problém v dnešní době nepatří mezi největší. Větším problémem může být nedostatečná rychlost připojení z důvodu zvýšeného toku dat mezi uživatelem a webovou aplikací. Největší problém nastává, pokud poskytovatel webové služby nedostatečně zabezpečí uložená data uživatelů. Riziko ztráty nebo úniku dat je v dnešní době velké. Pokud poskytovatel webové služby nepodcení zmíněné riziko, pak výhody použití webové aplikace převyšují nevýhody.^[1]

2 APLIKAČNÍ SERVER

Aplikační server je specializovaný software pro poskytování webové aplikace. Na rozdíl od běžného webového serveru aplikační server je specializován na běh určité aplikace, například odesílání emailů.^[2]

Aplikační server zprostředkuje operace mezi koncovým zařízením a databází, bezpečnost (autentizaci uživatele, přidělení rolí), vyhledávání nebo ukládání dat, zpracování dat a událostí mezi viditelnou částí aplikace a logickou.^[2]

Aplikační server komunikuje s webovou aplikací a koncovým uživatelem pomocí protokolu HTTP nebo HTTPS.

2.1 HTTP

HTTP neboli Hypertext Transfer Protocol je protokol, který umožňuje zobrazovat jednotlivé webové stránky nebo přenášet libovolné soubory. HTTP patří mezi bezstavové protokoly fungující na modelu požadavek/odpověď – server vygeneruje jednu odpověď pro jeden požadavek uživatele. HTTP patří také mezi nezabezpečené protokoly, tudíž není vhodný pro přenos citlivých dat, mezi které patří hesla, PINy nebo čísla platebních karet., nedostatek řeší aplikační protokol HTTPS.^[3]

2.2 HTTPS

HTTPS neboli Hypertext Transfer Protocol Secure je šifrovaná varianta komunikace pro bezpečný přenos dat mezi aplikačním serverem a uživatelem. Vhodný pro přenos citlivých dat z důvodu šifrování dat pomocí vrstvy SSL.^[4]

2.2.1 SSL

SSL neboli Secure Sockets Layer je vrstva vložena mezi vrstvou transportní a aplikační. Poskytuje zabezpečení komunikace šifrováním a autentizací komunikujících stran. Pro zabezpečené spojení používá asymetrické šifrování, kde každá z komunikujících stran vlastní dvojici klíčů – veřejný a soukromý. Veřejný klíč je zveřejněn všem, kteří budou požadovat komunikaci. Pokud pomocí získaného veřejného klíče bude zašifrována zpráva, dešifrování zprávy bude možné pouze odpovídajícím soukromým klíčem.^[5]

2.3 Životnost požadavku webové aplikace

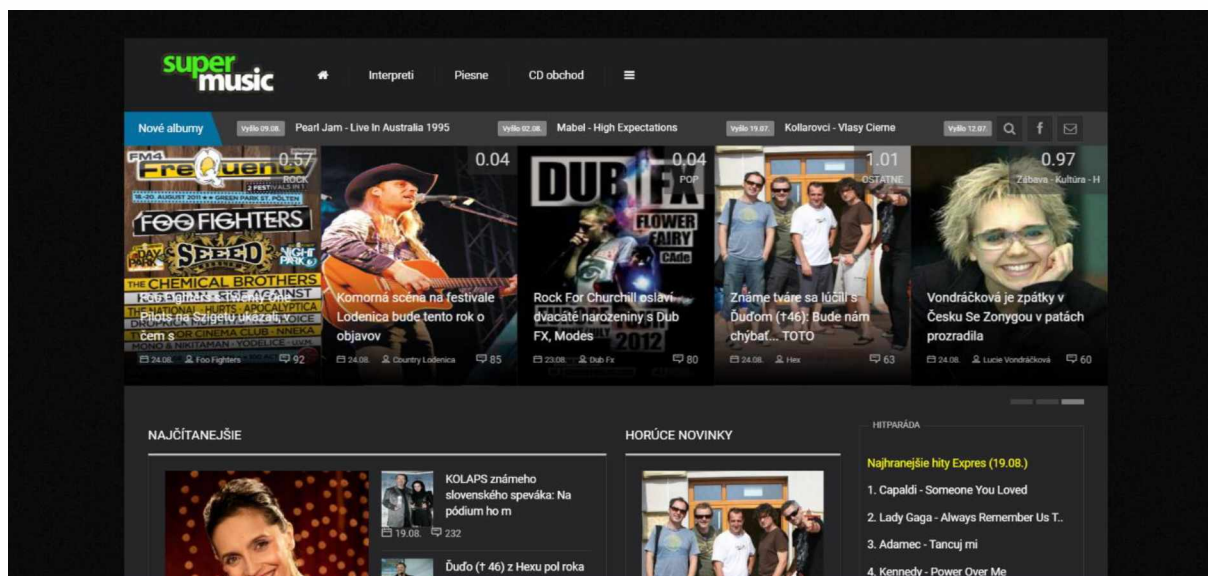
Vše začíná odesláním požadavku na webový server uživatelem pomocí webového prohlížeče. Odeslaná žádost je předána aplikačnímu serveru, který provede potřebné operace (například vyhledání dat, filtrování dat). Výsledná data odešle aplikační server ve formě odpovědi zpět na server webový. Webový server zobrazí výsledná data pomocí webového prohlížeče.

3 POROVNÁNÍ WEBOVÝCH APLIKACÍ PRO HUDEBNÍKY

Pro porovnání webových aplikací pro hudebníky byly vybrány nejznámější lokální i zahraniční webové stránky.

3.1 Supermusic

Supermusic.cz patří mezi nejznámější české stránky obsahující akordy k písním. Webová stránka disponuje návštěvností 270 000 návštěvníků měsíčně. Dalším obsahem vyplňujícím webovou stránku jsou aktuality ke konkrétním interpretům, koncerty a diskografie.



Obrázek 1 - Supermusic.cz domovská stránka

V horní liště webové stránky je navigační menu, které je rozděleno na kategorie interpreti, písně a obchod CD. Kategorie interpreti nabízí uživateli možnost vyhledat konkrétního interpreta pomocí postupného klikání písmen z názvu interpreta. Vyhledávání písní konkrétního interpreta probíhá totožným způsobem. Po vybrání konkrétního znaku je zobrazen výpis interpretů nebo písní, který se jeví jako nepřehledný vzhledem k množství zobrazených výsledků.

Interpreti: * **A B C Č D ě F G H I J K L Ľ M N O P Q R S Š T Ť U V X Y Z Ž**

Ka Kb Kc Kd Ke Kf Kg Kh Ki Kj Kk Kl Km Kn Ko Kp Kq Kr Ks Kt Ku Kv Kw Kx Ky
 (vidieť iba ■ slovenských ■ českých ■ zahraničných interpretov)

K ohni (1)	Kittner, Martin (13)	Kvokál (3)
K'naan (1)	KITTY IN A CASKET (5)	KWABS (1)
K's Choice (14)	Kiuas (37)	Kwan (32)
K-ci And Jo-jo (4)	Kivilaan, Triinu (25)	Kyanid (10)
K-maró (29)	Kizil, Bahar (1)	Kyau Vs Albert (8)
K-os (12)	Kladivo, Ján Boleslav (6)	Kybel Mačiek (26)
K-street (6)	Kladivoun (1)	KYGO (15)
K., Klára (1)	Klang, Donnie (1)	Kygo & Conrad (1)
K. Michelle (4)	Klangkarussell (2)	Kylähullut (1)
K.e.k. (1)	Klapa Intrade (5)	Kylian, Mash (2)
K.i.z. (5)	Klara. (4)	Kyo (29)
K.o.p.r. (2)	Klásek, Martin (2)	Kypck (1)
K.o.r (1)	Klaws, Alexander (37)	Kyra (6)
K.t.o. (22)	Klaxons (15)	Kysucky Pramen (1)
K1-unity (4)	Klee (20)	Kyte (2)
Kiló (2)	Kleerup, Andreas (2)	Kyuss (62)
K2 (14)	Klena, Edo (14)	K\`naan (40)
K:music (16)	Kleopatra (muzikal) (11)	L Plus (1)
k:song (1)	Klesniaková, Marika (3)	L&kids (4)
Kaas, Patricia (12)	Klezmatiks (1)	L-kama (1)
Kabát (545)	Klf (16)	L.a. (11)
Kabeer (2)	Klič (50)	L.a. Guns (77)
Kabeláč, Miloslav (1)	Klika (9)	L.a.x. (1)
Kabelková, Žofie (13)	Klika, Jiří (2)	L.m.c. (1)
Kachna A Zodiac (19)	Kliment, Peter (2)	L.o.s. (44)
Kachút, Marián (2)	Klimentová, Miriam (1)	L.u.z.a. (14)
Kaci (17)	Klimentová, Silvia (13)	L7 (10)
Kačírková, Irena (1)	Klimeš, Petr (13)	La Bouche (15)
Kadebostany (1)	Klímová, Simona (2)	La Dispute (8)
Kadence (1)	Klingande (2)	La Gioia (1)
Kadison Joshua (9)	klipers (5)	La Guns (2)
Kae, Jennifer (1)	Kloboučník A Kapela (3)	La Honda (21)
Kagrra (3)	Klubbingman (3)	La Madre Del Topo (1)
Kahl, Deborah (6)	Klus, Tomáš (253)	La Musicalite (2)
Kahovcová, Vlasta (1)	Klusák, Jan (1)	La Oreja De Van Gogh (72)
Kahovec, Karel (16)	Kme (1)	La Quinta Estación (5)
Kaiser, Miriam (9)	Kmeť, Slavó (4)	La Rose, Natalie (1)
Kaiser Chiefs (70)	Kmeťo, Igor (1)	La Roux (19)
Kája Diepold (16)	Kmet'oband (26)	La Rubia (10)
Kajaba, Martin (1)	Kmfdm (63)	La Slum Lords (1)
Kajagoogoo (27)	Kmoch, František (4)	La4 (2)
Kajiura, Yuki (12)	Knack (2)	Labajová, Janka (2)
Kajo Kajo., Kaidžas (55)	Knapp, Jennifer (36)	Labelle And Patti (2)
Kaktus (12)	Knatem (5)	Labelle Patti (1)
Kalafina (1)	Kňažko, Milan (2)	Labrinth (13)
Kalandra, Petr (37)	Knechtová, Katarína (61)	Labuda, Marián (2)
KALEO (1)	Knife (7)	Labyrinth (22)
Kali (138)	Knight, Sterling (16)	Lachey, Nick (27)

Obrázek 2 - Supermusic.cz vyhledání interpreta

Kategorie písne obsahuje možnosti vyhledání písne totožným způsobem jako interpreta, jak již bylo zmíněno, dále zobrazení nejnovějších písni s akordy a tabulaturami. Výpis opět obsahuje velké množství dat, jejichž přehlednost je navíc snížena přidáním data vložení písne před příspěvek. Přiložené obrázky zobrazují pouze část celkového výpisu a slouží pouze jako ukázka výpisu jednotlivých typů dat uživateli.

<p>2019-08-10: Země Draků intro - Divokej Bill</p> <p>2019-07-30: 1 - Samey</p> <p>2019-07-09: Inkvizícia - Davová Psychóza Boutique dog - Davová Psychóza</p> <p>2019-06-23: O mne a o Tebe - Para</p> <p>2019-06-13: Burlaci - basgitarra - Kabát</p> <p>2019-05-29: My Way - Frank Sinatra</p> <p>2019-05-23: El Condor Pasa (jednoduchá melodie) - Simon & Garfunkel</p> <p>2019-04-29: Rokeři - Alkehol Lihumil - Alkehol</p> <p>2019-04-28: Klikatý jsou cesty od sudu - Alkehol</p> <p>2019-04-21: Spalovač chlastu - Alkehol Máme spoustu chlastu - Alkehol</p>	<p>Substitute Version 2 - Who Substitute - Who Squeezebox Version 2 - Who Squeeze Box - Who Sparks Version 3 - Who Sparks Version 2 - Who Song Is Over - Who Smash The Mirror - Who I Go To Sleep - Sia</p> <p>2011-09-20: Slip Kid - Who Shakin All Over - Who Sensation - Who Papillon - Editors See Me Feel Me Version 3 - Who See Me Feel Me Version 2 - Who Sea And And Version 2 - Who Roadrunner - Who Relax - Who Real Good Looking Boy Version 2 - Who Munich - Editors Get Out Of My Face - Flatliners Fred's Got Slacks - Flatliners</p> <p>2011-09-19: How Deep Is Your Love - John Frusciante Madonna - Cocorosie South 2nd - Cocorosie Tekno Love - Cocorosie Werewolf - Cocorosie Maybe Not - Cat Power One Day - Matisyahu</p>
--	--

Obrázek 3 - Supermusic.cz nové příspěvky

Po vyhledání konkrétního interpreta je zobrazen profil s informacemi o interpretovi. V levé části obrazovky jsou zobrazeny aktuality o interpretovi, pravá část poskytuje možnost zobrazení klipů interpreta a výpis nejnovějších komentářů k interpretovi. Prostřední část je složena z možnosti zobrazit veškeré písně, diskografii, nebo fotogalerii interpreta.

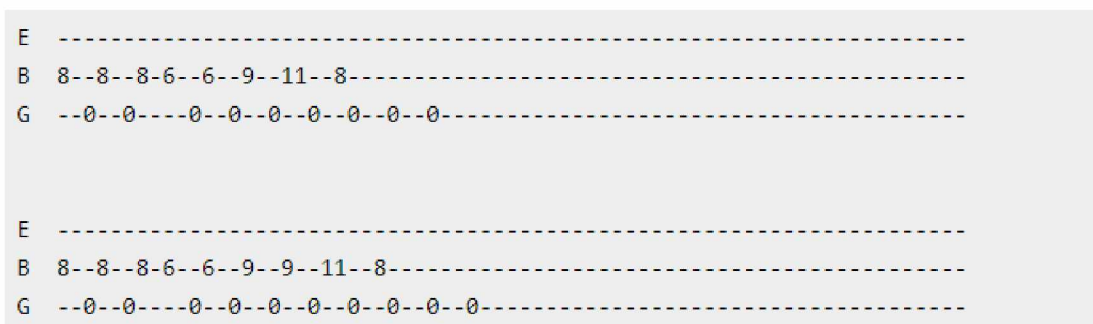
Obrázek 4 - Supermusic.cz konkrétní interpret

Výpis písní konkrétního interpreta je velmi obsáhlý z důvodu neřízeného přidávání písně uživateli, kterým stačí pouze provést úspěšně registraci. Z tohoto důvodu vzniká mnoho duplicit ale i přidání písní s akordy které mnohdy nejsou správné.



Obrázek 5 - Supermusic.cz příklad výpisu písní interpreta

Kvalita příspěvků není nikým nebo ničím kontrolována, pouze může být ohodnocena. Uživatel vyhledávající konkrétní píseň je obtěžován zobrazením veškerých příspěvků se shodným nebo podobným názvem, které obsahují pouze malou část písně, která ani nemusí být správně.



Obrázek 6 - Supermusic.cz příklad příspěvku

Na závěr pouze podotknu, že jsem osobně uživatelem této stránky již od roku 2009, ve kterém jsem byl aktivním přispěvatelem této stránky a velmi chyběla možnost zkontaktovat autora příspěvku s případným dotazem. Autora příspěvku nebylo možné kontaktovat, pokud nezanechal jinou možnost kontaktu přímo ve svém příspěvku.

Vítaj Krajta5			
Počet tvých piesní: 13. Zobrazujem záznamy: 0-13.			
Interpret:	Pieseň:	Videní:	Videní týždenne:
Chinaski	edituj Hvězdy nad hlavou	9029	17
Coldplay	edituj Clock 2	3447	7
Kabát	edituj Centryfuga solo	6816	13
Kabát	edituj Krutý krtek Joy	13037	24
Klus, Tomáš	edituj Láska	11816	23
Klus, Tomáš	edituj Marie	36246	71
Klus, Tomáš	edituj Neklid	11035	22
Klus, Tomáš	edituj Panenka	20659	41
Klus, Tomáš	edituj Pocity 2	14676	29
Landa, Daniel	edituj Walter, Ute, Tommy and Marry	6854	13
Landa, Daniel	edituj Walter, Ute, Tommy and Marry (nigredo live)	6673	13
Lordi	edituj Devil Is A Loser delane podle videa ozkousene	2564	5
Nightwish	edituj The Islander 2	3472	6

Obrázek 7 - Supermusic.cz vlastní příspěvky

krajta5
muz

Písní: 13
Článků: 0

Prispievateľ: krajta5 - piesní: 13

Chinaski	Hvězdy nad hlavou	9039
Coldplay	Clock 2	3455
Kabát	Centryfuga solo	6825
Kabát	Krutý krtek Joy	13054
Klus, Tomáš	Láska	11822
Klus, Tomáš	Marie	36291
Klus, Tomáš	Neklid	11045
Klus, Tomáš	Panenka	20682
Klus, Tomáš	Pocity 2	14695
Landa, Daniel	Walter, Ute, Tommy and Marry	6855
Landa, Daniel	Walter, Ute, Tommy and Marry (nigredo live)	6675
Lordi	Devil Is A Loser delane podle videa ozkousene	2565
Nightwish	The Islander 2	3472

Obrázek 8 - Supermusic.cz zobrazení profilu ostatních uživatelů

3.2 Pisnický-akordy

Následující porovnanou webovou stránkou je pisnický-akordy. Webová stránka působí přehledněji než supermusic.cz. Navigační menu obsahuje položky písně, diskografii, přehled akordů na kytaru, zpěvníky a komunita.

[Písníčky s akordy](#)
[Diskografie](#)
[Akordy na kytaru](#)
[Zpěvníky](#)
[Komunita](#)
[Přihlášení](#)

Interpreti
 písníčky
 alba

Texty písniček s akordy

Výběr: A B C D E F G H CH I J K
L M N O P Q R S T U V W X Y
Z 0-9

Oblíbení interpreti

Jaromír Nohavica

táborové písně

Zdeněk Svěrák, Ja...

Nedvědi

Nejoblíbenější písničky

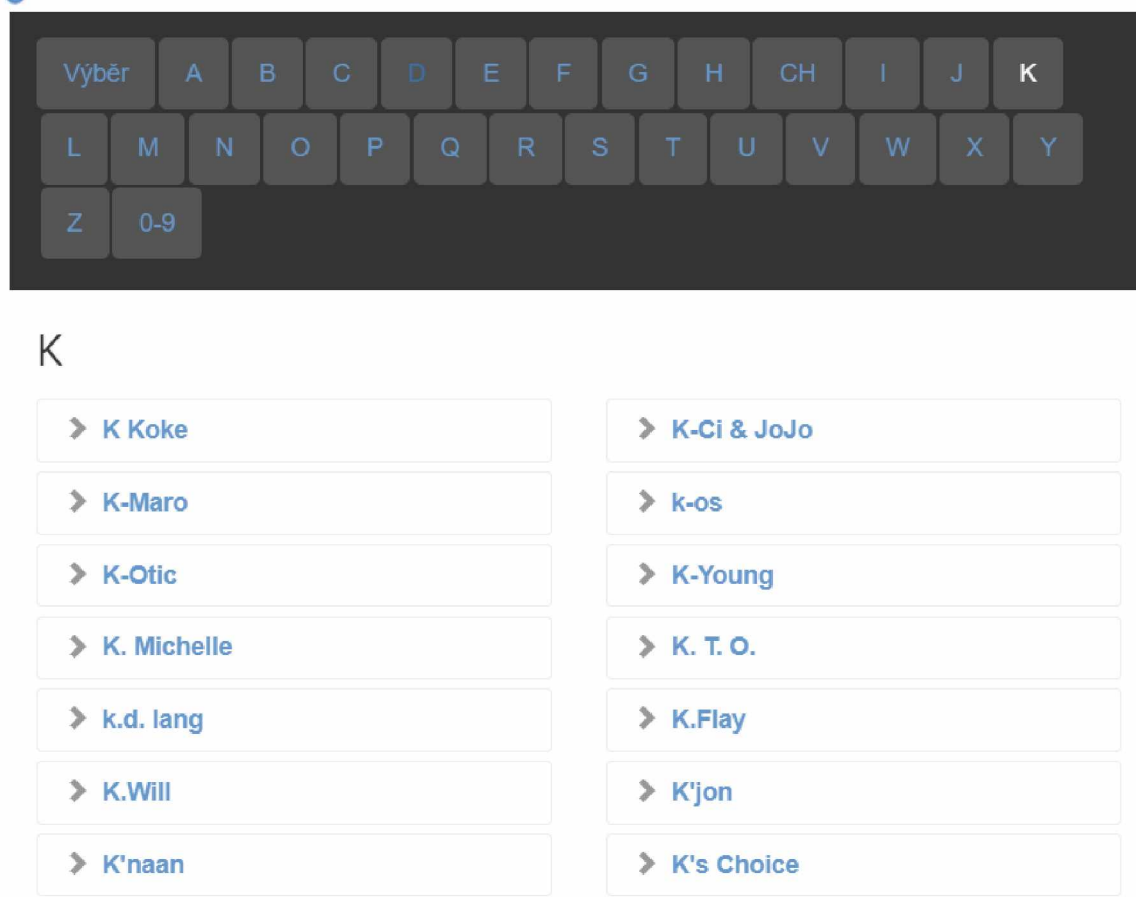
- > Cesta - Kryštof
- > Bláznova ukolébavka - táborové písně
- > Anděl - Karel Kryl
- > Okoř - táborové písně
- > Montgomery - táborové písně
- > Půlnoční - Václav Neckář
- > Komete - Jaromír Nohavica

Nově v diskuzi

Texty WTF???
v Hlavní Fórum / Počec o všem možném

Obrázek 9 - Pisnický-akordy.cz domovská stránka

První možnost vyhledání konkrétního interpreta je pomocí zadání prvního znaku z názvu, který odkáže na výpis veškerých interpretů způsobem podobným supermusic.cz. Výpis je čitelnější, ale obsah dat a vyhledávání ve výpisu není přehledné.



Obrázek 10 - Pisnicky-akordy.cz ukázka výpisu vyhledání interpreta

Druhá možnost vyhledání interpreta je zadáním celého názvu interpreta nebo písně. Po úspěšném vyhledání a vybrání interpreta nastane přesměrování uživatele na stránku s písněmi, které je možné řadit různými způsoby.

Kabát akordy a texty písní

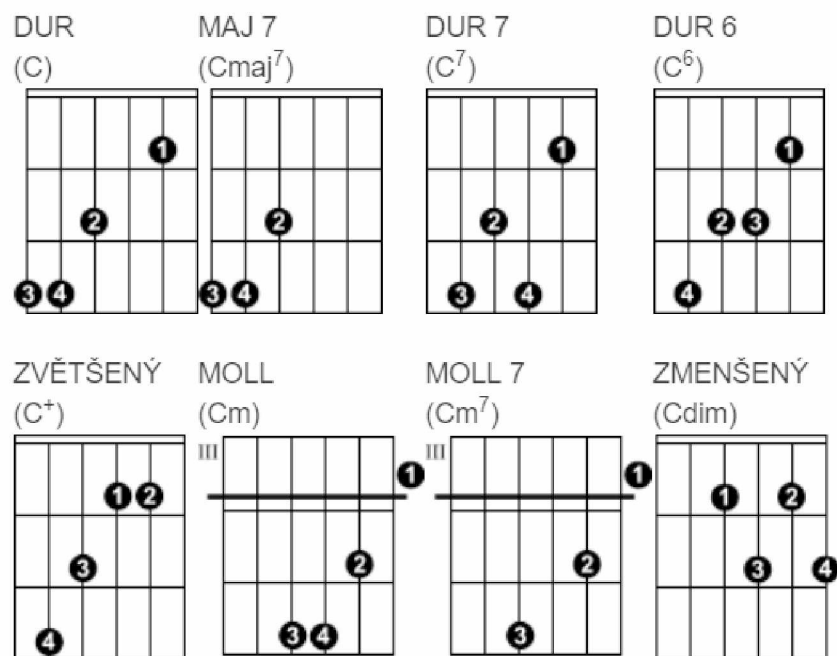
Nejhranější písničky A - Z Chronologicky na albech

> Pohoda	> Colorado
> Malá dáma	> V pekle sudy válej
> Šaman	> Burlaci
> Na sever	> Bára
> Kdoví jestli	> Moderní devče
> Láďa	> Starej bar
> Dábel a syn	> Houby magický
> Žizeň	> Dole v dole
> Raci v práci s.r.o.	> Corrida
> Stará Lou	> Čert na koze jel
> Schody	> Bruce Willis

Obrázek 11 - Pisnickyy-akordy.cz výpis písní

Výpis písní je nesrovnatelně přehlednější z důvodu pouze jednoho příspěvku k jedné písni na rozdíl od supermusic.cz. Nevýhoda spočívá v možnosti pouze jednomu uživateli vložit píseň se stejným názvem, pokud je tedy v písni chyba nebo není úplná, pak jiný uživatel nemá možnost doplnit chybějící část písně, případně vložit svoji verzi.

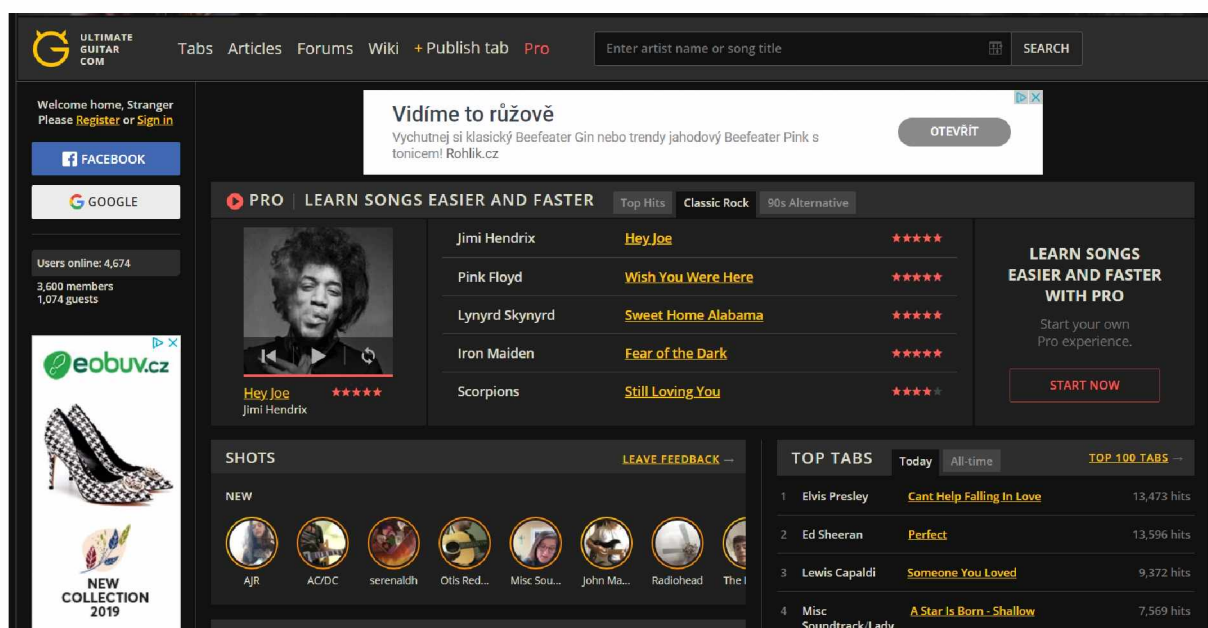
Možnost akordy na kytaru v navigačním menu obsahuje velké množství základních akordů, které nabízí uživateli možnost vyhledat akord, který zahlédl v písni a doposud jej nezná. Nevýhoda vybraného řešení spočívá ve složité dostupnosti. Uživatel, který potřebuje vyhledat neznámý akord je donucen otevřít novou záložku prohlížeče, vybrat možnost z navigačního menu, vybrat písmeno akordů a poté vyhledat konkrétní potřebný tvar akordu, tento postup je zapotřebí opakovat při každém vyhledání. Výpis tvarů akordu působí přehledně.



Obrázek 12 - Pisnicky-akordy.cz vyhledaný akord

3.3 Ultimate-guitar

Ultimate-guitar patří mezi velké zahraniční komerční a velmi propracované webové aplikace určené hudebníkům ke sdílení písni, tabulatur a mnoho dalších formátů.



Obrázek 13 - Ultimate-guitar domovská stránka

Webová aplikace obsahuje širokou databázi interpretů, písní, amatérských videí s ukázkami hraní písní na hudební nástroj, vlastní fórum, ve kterém uživatelé mohou sdílet své otázky, články týkající se hudby a mnoho dalších možností. O úspěchu webové aplikace svědčí počet zaregistrovaných uživatelů (přes 10 milionů zaregistrovaných uživatelských účtů) a celková aktivita uživatelů (přibližně 20 milionů zobrazení měsíčně).

Vysoký počet aktivních uživatelů koresponduje s počtem příspěvků, které jsou k jednotlivým interpretům přidávány. Široká možnost využití ale nemůže nahradit přehlednost. Vyhledání písně u známějšího interpreta může znamenat prohledání i 30 stránek s písněmi z důvodu nekontrolovaného přidávání příspěvků které vede mnohdy k duplicitám.

Afire Love	★★★★★ 488	Chords
Afire Love (ver 2)*	★★★★★ 33	Chords
Afire Love (ver 3)	★★★★★ 11	Chords
Afire Love (ver 4)	★★★★★ 3	Chords
Afire Love (ver 5)*	★★★★★ 3	Chords
Afire Love (ver 6)*	★★★★★ 78	Chords
Afire Love (ver 7)*	★★★★★ 3	Chords
Afire Love (ver 8)*	★★★★★ 5	Chords
Afire Love (ver 9)*		Chords
Afire Love (ver 10)*		Chords
Afire Love Acoustic*	★★★★★ 33	Chords
Afire Love	★★★★★ 18	Tab
Afire Love*	★★★★★ 4	Bass
Afire Love*		Video
Afire Love*	★★★★★ 11	Ukulele

Obrázek 14 - Ultimate-guitar výpis písní

Pro úspěšné vložení příspěvku musí být splněn dlouhý seznam podmínek a obsah příspěvku musí být ve správném formátu. Správný formát příspěvku je podrobně vysvětlen v sekci s nápovědou. Nutnost zdlouhavého studování podmínek pro přidání a správného formátu příspěvku může odradit mnoho zkušených hudebníků, kteří mohli vkládat kvalitně zpracované příspěvky. Mnohdy dochází k schválení příspěvků, které porušují různými způsoby podmínky vložení příspěvku.

A. Number of strings **REQUIRED**

Tab should consist of as many lines as there are strings on the instrument. (For example, 7 lines for a 7 string guitar).

```
e|-----15-----14-----|-----15-----14-----|
B|-----15-----15-----|-----15-----15-----|
G|-----14-12---14---14---|-----14-12---14---14---|
D|-12-----12-----|-----12-----12-----|
A|-----12-----12-----|-----12-----12-----|
E|-----12-----12-----|-----12-----12-----|
```

Correct
6 lines (for 6-string guitar)

```
e|-----15-----14-----|-----15-----14-----|
B|-----15-----15-----|-----15-----15-----|
```

Incorrect
Not enough lines

Obrázek 15 - Ultimate-guitar ukázka správného formátu

Ultimate-guitar nabízí uživateli možnost placeného členství. Placené členství nabídne uživateli kvalitnější obsah (ověřené akordy, tabulatury, návody, jak hrát písně, možnost stáhnout písně ve formátu pro program Guitar Pro). Cena členství je 1800 Kč na jeden rok.

3.3.1 Guitar Pro

Guitar Pro je počítačový program určený pro tvorbu hudby. Obsahuje velké množství různých nástrojů. Hudbu je možné tvořit a editovat pomocí not, tabulatur nebo přímo vybíráním tonů. Program obsahuje lokalizaci pro Českou republiku.



Obrázek 16 - Guitar Pro ukázka

4 VYHODNOCENÍ WEBOVÝCH APLIKACÍ

Vybrání nejlepší webové aplikace je vázáno na požadavky konkrétního uživatele. Uživatel potřebující znát akordy nebo tabulaturu na píseň nejspíše navštíví Ultimate-guitar. Nepřeberné množství zahraničních interpretů a jejich tvorby spolu s možností zaplacení členství a tím také přístupu k novým možnostem. Problém ovšem nastává, pokud uživatel potřebuje vyhledat lokální interprety, které na této zahraniční webové stránce nejsou k nalezení.

Pro písň lokálních interpretů zbývá použít webové aplikace Supermusic nebo Pisnický-akordy. Kvalita příspěvků nedosahuje úrovně Ultimate-guitar (mnohdy neúplné nebo nesprávné příspěvky), ale pro začátečníka je to šance rozpoznat nebo dokonce opravit chyby v příspěvku a vylepšit tímto způsobem svoji hudební úroveň. Zkráceně, pro začátečníka je lepší možnost alespoň částečně správně napsaný příspěvek než začínat úplně od začátku.

Webová aplikace Ultimate-guitar také jako jediná nabízí přímé zobrazení akordů v konkrétním příspěvku. Funkce patří mezi velmi užitečné, umožňuje uživateli zobrazit tvar konkrétního akordu bez dalšího vyhledávání.

Výběr nejlepší webové aplikace tedy silně závisí na preferencích uživatele. Pokud vynecháme absenci lokálních interpretů a funkcí dostupných bez zaplacení členství, patří Ultimate-guitar mezi nejlépe zpracované po technické stránce.

	Supermusic	Pisnický-akordy	Ultimate-guitar
Přidání interpreta uživatelem	Ne	Ne	Ne
Přidání písně uživatelem	Ano	Ne	Ano
Přehledný výpis interpretů	Ne	Ne	Ano
Přehledný výpis písní	Ne	Ano	Ne
Responzivní	Ne	Ano	Ne
Moderní vzhled stránky	Ne	Ano	Ano
Možnost přímého zobrazení akordu	Ne	Ne	Ano
Pravidla pro přidání příspěvku	Ne	Ne	Ano
Jednotný tvar příspěvků	Ne	Ne	Ano
Možnost komunikace mezi uživateli	Ne	Ne	Ano
Kvalitnější obsah za poplatek	Ne	Ne	Ano
Písně interpretů z České republiky a Slovenska	Ano	Ano	Ne

Tabulka 1 - porovnání webových aplikací

Výsledkem praktické části diplomové práce je webová aplikace spojující kladné stránky zmíněných webových aplikací, řešení nedostatků, implementace nových funkcí a možností řízení. Mezi hlavní nedostatky patří nepřehledný výpis interpretů a písni, duplicitní, neúplné nebo chybné záznamy, nemožnost přidávat interprety a příspěvky zaregistrovaným uživatelům, složitá pravidla pro vložení příspěvku, nemožnost kontaktovat ostatní uživatele, způsob řízení vkládání nových příspěvků, nedostatečná motivace uživatelů vkládat nové příspěvky nebo nemožnost ohodnotit konkrétní příspěvek. Popis jednotlivých částí webové aplikace a jejich funkčnosti bude popsán v kapitole uživatelská příručka.

5 POUŽITÉ TECHNOLOGIE

Následující kapitola slouží k popsání technologií použitých k vytvoření praktické části diplomové práce.

5.1 ASP.NET MVC

ASP.NET MVC je webový framework využívající architektonický vzor Model-View-Controller. ASP.NET používá programovací jazyk C#, pomocí kterého jsou vytvářeny modely, controllery i části pohledů (při použití Razor syntaxe). ASP.NET umožňuje již při vytváření projektu vybrat, zda je potřeba autorizace uživatelů. V případě potřeby vloží do nově vytvořeného projektu připravené metody pro autorizaci uživatele, vícefázové ověření uživatele a další způsoby zajišťující bezpečnost.

5.1.1 Programovací jazyk C#

Programovací jazyk C# je vysokoúrovňový objektově orientovaný jazyk vytvořený společností Microsoft. Jazyk je založen na jazycích C++ a Java. Programovací jazyk disponuje širokou škálou použití, například webové aplikace, databázové programy, webové služby, formulářové aplikace. Jazyk C# patří mezi jazyky s virtuálním strojem, který vznikl spojením kompilátoru a interpreta. Při zpracování je zdrojový kód přeložen do mezikódu (CLI – Common Language Intermediate) který je podobný binárnímu kódu spolu s jednodušší instrukční sadou která podporuje objektově orientované programování. Mezikód je poté interpretován virtuálním strojem, jehož výsledkem je zdrojový kód pro procesor.^[6]

5.1.2 MVC

Základní myšlenkou MVC je oddělení logické části od výstupu. Přístup eliminuje vytvoření jednoho velkého souboru obsahující logické operace, zobrazení výstupu nebo databázové

dotazy. Velké soubory s různými typy kódu jsou špatně udržovatelné i rozšiřovatelné. Problém může nastat také ve správném zvýraznění kódu ve vývojovém prostředí.

Základní části MVC:

- Model: Model je sada tříd popisující data, se kterými aplikace pracuje. Model může obsahovat například databázové dotazy nebo validace. Model netuší odkud data přišla ani jak budou data vypsána. Při použití Entity Frameworku je podoba modelů ekvivalentní s podobou databázových tabulek. Modelu lze také přidat anotace které mohou upřesnit využití jednotlivých atributů třídy.

```
public partial class Clanek
{
    [DisplayName("ID")]
    public int Id { get; set; }

    [DisplayName("Titulek")]
    [Required(ErrorMessage = "Titulek musí být vyplněný.")]
    public string Titulek { get; set; }

    [DisplayName("Popisek")]
    [DataType(DataType.MultilineText)]
    [Required(ErrorMessage = "Popisek musí být vyplněný.")]
    [AllowHtml]
    public string Popisek { get; set; }
}
```

Obrázek 17 - ASP.NET MVC ukázka modelu s anotacemi

- View: Základní vlastností view neboli pohledu je zobrazení výstupu uživateli. Pohled může obsahovat HTML tagy ale i tagy speciálních jazyků. V případě ASP je příkladem speciálního jazyka jazyk Razor. K označení začátku Razor příkazu slouží znak zavináč, který označuje přechod z jazyka HTML na jazyk C#. Razor může například odkazovat na hodnoty předané z kontroleru ale přesměrovávat na jiné stránky. Pohledy mohou také být typu partial neboli částečné. Částečné pohledy umožňují vnořit pohled do pohledu jiného. Využití částečného pohledu může být například při opakovaném zobrazení komponent (například navigační menu stránky).

```
<ul>
<li class="contact" data-value="@uzivatel.Id">
    <a href=@Url.Action("Index", "Zprava", new { idUzivatele = uzivatel.Id })>
        <div class="wrap">
            
            <div class="meta">
                <p class="name">@uzivatel.Prezdivka</p>
            </div>
        </div>
    </a>
</li>
</ul>
```

Obrázek 18 - ASP.NET MVC ukázka view s Razor syntaxí

- Controller: Controller je prvek který propojuje Model a View. Controller reaguje na žádost uživatele a následně rozhodne který pohled má být zobrazen a jakými daty naplněn. V ASP:NET MVC je zvyklostí pojmenovat třídu příponou Controller. Funkce controlleru může být také upřesněna anotacemi, například [RequireHttps] nebo [Authorize] které požaduje před použitím metody controlleru nebo celého controlleru po uživateli autorizaci.

5.1.3 Ostatní způsoby využití ASP.NET

Pro zajímavost stručný popis ostatních způsobů využití ASP.NET:

- WebForms: Pokus o přenesení WinForms (desktopová verze formulářových aplikací) na webové rozhraní. Princip spočívá v přenášení komponent a jejich poskládání pomocí designeru. Jednotlivým prvkům formuláře jednoduché přiřazení událostí.^[7]
- Single Page: Minimalizace zátěže serveru pomocí jednostránkové aplikace, která je kompletně stažena ze serveru při použití. Zobrazení a skrývání jednotlivých podstránek je dosaženo pomocí JavaScriptu, synchronizace dat se serverem pomocí AJAX.^[8]
- Core: Verze ASP.NET umožňující použití i mimo operační systém Windows. Hlavní rozdíl proti ASP.NET MVC spočívá v nahrávání projektu se kterým je spolu s projektem přiložen i ASP.NET framework, který již nemusí být na serveru nainstalován. V nejnovější verzi ASP.NET Core 2.1 je již plně podporován také Entity Framework.^[7]

5.2 Entity Framework

Entity Framework využívá objektově relační mapování (ORM), které umožňuje uložení .NET objektů v relační databázi. Získávání objektů z databáze probíhá pomocí jazyk LINQ. Použití Entity Frameworku lze odlišit pomocí různých přístupů, mezi možnosti patří Code First, Model First, Database First. Porovnání jednotlivých přístupů, výhody a nevýhody pochází ze zdroje [9].

5.2.1 Model First

Pokud uživatel nemá zkušenosti s většinou softwaru určených pro návrh (například XML-based DataSet Schema) a vizuální rozhraní (Entity Designer Model XM), pak může být přístup Model First pro uživatele matoucí. Klíčem k porozumění může být, že slovo Model má definovat vizuální diagram vytvořený pomocí nástrojů pro návrh. Vytvořený diagram bude poté použit frameworkem k vygenerování SQL skriptu pro vytvoření databáze.

Model First zjednodušeně znamená pro uživatele pouze práci s diagramem a zbytek práce nechat na ORM nástroji, v tomto případě provádí roli ORM nástroje Entity Framework.

Výhody Model First:

- Možnost vytvoření složitého databázového schéma pomocí vizuálního nástroje
- Pokaždé, kdy je provedena změna databáze, dojde k aktualizaci modelu bez ztráty dat

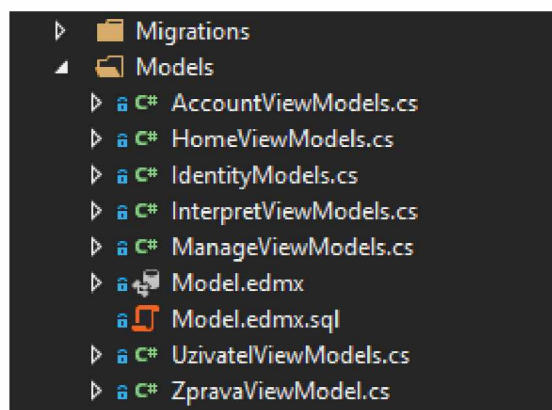
Nevýhody Model First:

- Automaticky generované SQL skripty mohou vést ke ztrátě dat. Řešením je generování skriptů na disk a jejich následná úprava, která vyžaduje znalost SQL.
- Nedostatečná kontrola nad modelovými třídami. Zdrojový kód bude automaticky vygenerován nástrojem.

5.2.2 Database First

Přístup Database First je vhodný, pokud pro projekt je již vytvořená databáze, zároveň také ruší nevýhody přístupu Model First z důvodu opačného směru vývoje. V případě Model First je prvně uživatelem vytvořen model (EDMX) ze kterého je poté vytvořen SQL skript pro vytvoření databáze. Pokud uživatel zvolí Database First, pak je model vytvořen z již vytvořené databáze.

Database First zjednodušeně znamená připojení projektu na konkrétní databázi, ze které ORM nástroj automaticky vygeneruje zbylé potřebné informace.



Obrázek 19 - EDMX model a SQL skript ve Visual Studiu

Výhody Database First:

- Vhodný způsob využití Entity Frameworku, pokud již existuje databáze.
- Riziko ztráty dat je minimální, protože jakákoliv změna nebo aktualizace bude vždy vykonána v databázi.

Nevýhody Database First:

- Menší kontrola nad vygenerovanými třídami ORM nástrojem než u přístupu Model First.

5.2.3 Code First

Code First je elegantním a vysoce výkonným přístupem vývoj datového modelu. Code First umožňuje uživateli definovat modelové třídy jako standardní třídy programovacího jazyka bez potřebné znalosti návrhových nástrojů nebo velkého množství automaticky vygenerovaného kódu.

Zjednodušeně Code First znamená napsání tříd které slouží jako modelové třídy, podle kterých bude vygenerována databáze k projektu.

```
public partial class Zprava
{
    [DisplayName("ID")]
    public int Id { get; set; }

    [DisplayName("Čas odeslání")]
    public System.DateTime CasOdeslani { get; set; }

    [DisplayName("Text")]
    [Required(ErrorMessage = "Text nemůže být prázdný.")]
    public string Text { get; set; }
}
```

Obrázek 20 - příklad modelové třídy

Výhody Code First:

- Použití ORM nástroje bez znalosti návrhových nástrojů, může být výhodné pro malé až středně velké projekty.
- Úspora času ručním nadefinováním modelových tříd.
- Umožnění vývojářům následovat konvenční přístup ke konfiguraci a zároveň použít vlastní implementaci při potřebě přizpůsobení mapování databáze.

Nevýhody Code First:

- Nutná dobrá znalost ORM programovacího jazyka a konvencí (v případě Entity Frameworku programovací jazyk C#)
- Udržování databáze bez ztráty dat může být náročná.
- Při aktualizaci dat může dojít ke ztrátě.

5.2.4 LINQ

LINQ neboli Language Integrated Query je integrovaný jazyk .NET Frameworku pro dotazování. Dotazy jsou vytvářeny jako jednoduché řetězce bez kontroly typu v době kompilace. Pomocí LINQ je možné data vyhledávat, filtrovat, řadit, seskupovat na zdrojích dat s minimální délkou kódu. Pomocí LINQ je také možné vytvářet anonymní třídy jejichž využití

může být vhodné potřebujeme-li pouze určitá data z objektu. Možnost využití lambda výrazů pro jednodušší zápis dotazu.^[10]

5.2.5 Migrace

V případě nutné změny v modelu (například přidání nebo odstranění atributu) je nutné provést migraci. Pomocí migrací aktualizujeme vytvořený projekt, podle vytvořené migrace může být automaticky aktualizována i databáze a kontext Entity Frameworku. Migrace probíhá pomocí Packové Manager Console ve Visual Studiu. Pomocí příkazů Enable Migrations dojde k povolení migrací, které jsou ve výchozím stavu zakázané. Příkazem Add-Migration vložíme nově vytvořenou migraci. Posledním příkazem je Update-Database pomocí kterého dojde k aktualizaci databáze. Migrace probíhá nad kontextem, pokud projekt obsahuje kontextů více musí být určeno nad kterým kontextem migrace probíhá spolu s určenou složkou pro uložení migrace. Migrace také mohou být prováděny automaticky.^[11]

5.3 Technologie pro oživení View

Následuje popis technologií použitých v jednotlivých View pro zajištění kompletní funkčnosti ale i přechodových efektů – přidání dynamiky zobrazených webových stránek.

5.3.1 JavaScript

JavaScript patří mezi technologie používané k vytvoření webové aplikace, které jsou zpracovány na straně klienta. Umožňuje implementaci komplexních částí webových stránek případně změnu konkrétních částí stránky (události, dynamické menu, odebrání komponent, přidání vlastností komponentě, validace hodnot). JavaScript je objektově orientovaný skriptovací jazyk s dynamickým přiřazováním datových typů.^[12]

5.3.2 jQuery

jQuery patří mezi JavaScriptové knihovny. Hlavním účelem je zjednodušení manipulace s obsahem stránky, reakcí na události, animace a použití AJAXu. Další výhodou je velké množství jQuery pluginů které je možné volně použít (interaktivní galerie, formulářové prvky). jQuery je možné stáhnout a umístit na stránku nebo možnost nalinkovat.^[13]

5.3.3 AJAX

AJAX neboli Asynchronous JavaScript and XML je technologie měnící stav stránky bez potřeby znovunačtení. Asynchronnost je vlastnost JavaScriptu zavolat na serveru skript a nečekat bezprostředně na odpověď, pokračuje tedy v provádění kódu bez zastavení. Takový kódu nebo funkce se nazývá neblokující, protože čekání na odpověď nezastaví průběh

provádění skriptu čekáním na odpověď nebo zpracováním. Na straně uživatele nastane například zobrazení stránky, se kterou může pracovat bez čekání na odpověď. V moment, kdy odpověď přijde, dojde k zavolání callback funkce, která bude provedena po obdržení odpovědi ze serveru.^[14]

5.4 MYSQL

SQL je standardizovaný dotazovací jazyk používaný v relačních databázích pro práci s daty. Příkazy jazyka SQL jsou rozděleny do následujících skupin:

- Příkazy pro definici dat (DDL): vytvářejí nebo upravují strukturu databáze (CREATE, ALTER, DROP)
- Příkazy pro manipulaci s daty (DML): určeny k získání, uložení a mazání dat v databázi (SELECT, INSERT, UPDATE, DELETE)
- Příkazy pro řízení přístupových práv (DCL): určeny pro správu uživatelských rolí a práv (GRANT, REVOKE)
- Příkazy pro řízení transakcí (TCL): pro správu databázových transakcí (COMMIT, ROLLBACK)

Veškeré informace o SQL byly získány ze zdroje [15].

5.5 SQL Server

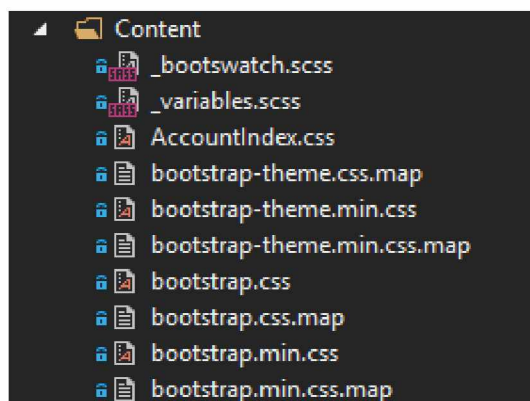
SQL server je relační systém řízení báze dat (RDBMS). Jedná se o nástroj, který za uživatele vyřeší mnoho problémů a zároveň je jednoduchý na použití. Uživatel komunikuje s databází pomocí jazyka SQL, který je dobře čitelný. RDBMS řeší úlohy zabezpečení, optimalizace výkonu, zajištění konzistence dat v databázi, umožňuje spojit více dotazů do transakcí. Zmíněné vlastnosti jsou shrnovány zkratkou ACID: Informace o SQL Serveru pochází ze zdroje [16].

- Nedělitelnost (Atomicity): operace prováděné v transakci jsou prováděny jako jedna atomická operace – pokud některá z částí selže, vrátí se databáze do původního stavu před započítím transakce a změny nebudou provedeny
- Zachování konzistence (Consistency): po každém dokončení transakce musí být stav databáze v konzistentním stavu
- Izolace (Isolation): jednotlivé operace prováděné uživateli jsou izolované a nemohou být navzájem ovlivňovány, pokud nastane situace, ve které existuje v jeden okamžik více dotazů pracujících se stejnými daty, pak jsou dotazy prováděny postupně ve stylu fronty

- Trvanlivost (Durability): veškerá úspěšně zapsaná data jsou uložena na trvalé uložení, v případě přerušení provozu RDBMS zůstávají data uložena ve stavu těsně před přerušením provozu

5.6 Bootstrap

Bootstrap je sada nástrojů určená pro tvorbu webových aplikací. Obsahuje již předpřipravené šablony, které je možné pouze upravit dle potřeby uživatele. Pro správné použití je nutná znalost HTML a CSS z důvodu vkládání komponent pomocí HTML a CSS. Při vytvoření nového projektu ASP.NET pomocí Visual Studia je Bootstrap již obsažen ve struktuře projektu.



Obrázek 21 - Bootstrap styly ve Visual Studiu

Frameworku Bootstrap zjednodušeně slouží k jednoduché úpravě grafické stránky webové aplikace. Bootstrap nabízí znovu využitelné komponenty (například pro uživatelské rozhraní, tlačítka, navigační menu, formulářové prvky, zobrazení obrázků, modální okna, rozbalovací tlačítka, stránkování a mnoho dalších možností).

Výhody použití Bootstrap:

- Responzivita designu webové stránky pro mobilní zařízení.
- Od verze Bootstrap 3 podpora Mobile First přístupu.
- Pozicování komponent pomocí 12 sloupcového systému (GRID).
- Použití zdarma i pro komerční účely.

Informace o Bootstrap čerpány ze zdroje [18].

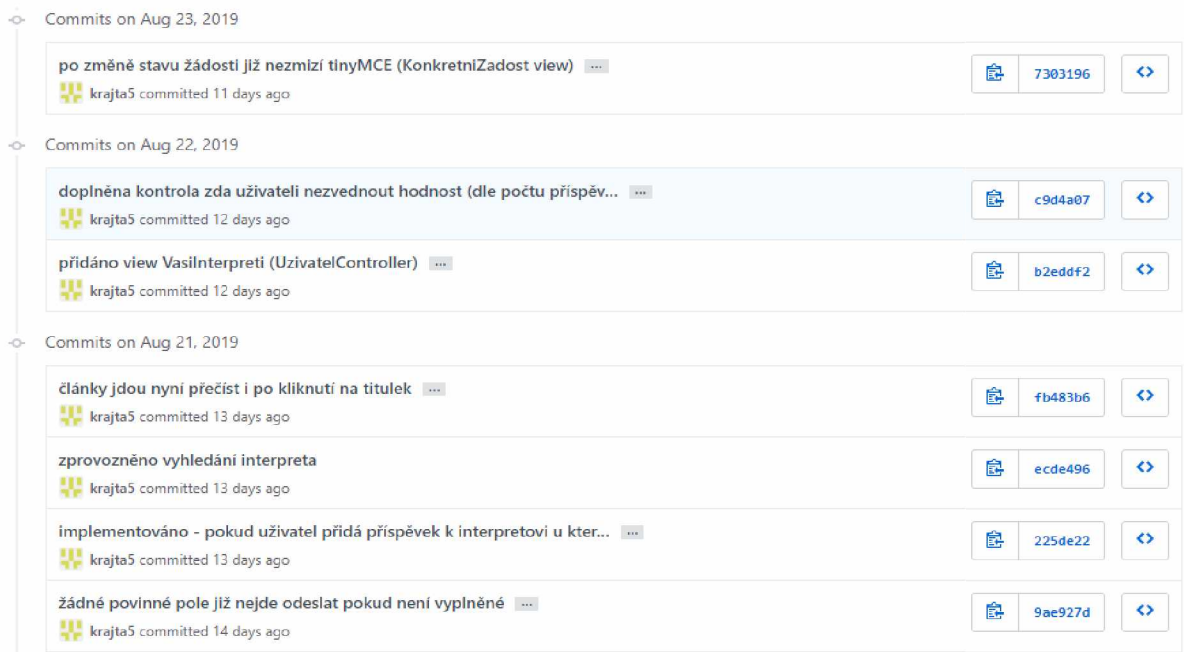
5.7 SendGrid

SendGrid patří mezi služby odesílající emailové zprávy. SendGrid patří mezi skupinu služeb Azure Web Apps které používají protokol SMTP k odesílání. Hlavní výhoda spočívá

v jednoduchosti připojení k webové aplikaci. Pokud je dostačující základní bezplatná verze, který disponuje možností odeslat 1 200 emailů měsíčně nebo 400 mailů denně.^[17]

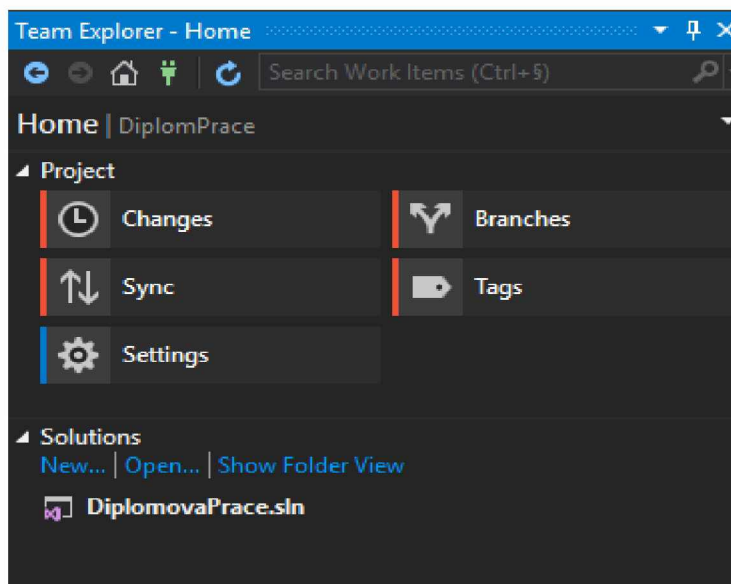
5.8 GitHub

GitHub je internetová služba poskytující bezplatný webhosting pro open source projekty. Nabízí funkce pro řízení přístupu, spolupráci, sledování chyb, správu úloh, rozdělení verzí projektu do více větví.



Obrázek 22 - GitHub přehled provedených změn

Možnost nahrání nových verzí projektů na server je vícero. Mezi první možnost patří nahrání projektu přímo přes webové prostředí, druhá možnost pomocí desktopové aplikace, kterou lze spustit bez instalace, třetí možností je propojení Visual Studia s konkrétním repositářem.



Obrázek 23 - použití GitHub pomocí Visual Studia

5.9 Struktura projektu ASP.NET

Pouze pro zajímavost připojuji výchozí strukturu projektu vytvořeného pomocí ASP.NET MVC. Projekt obsahuje následující kořenové složky: Informace o struktuře projektu pochází ze zdroje [18].

- App_Data: určena pro soubory do kterých je zapisováno a ze kterých je čteno (databázové soubory, XML soubory)
- App_Start: obsahuje třídy které jsou potřebné při startu webové aplikace, soubory obvykle obsahují konfiguraci webové aplikace
- Content: obsahuje statické soubory (například CSS, obrázky, ikony), nalezneme zde také Bootstrap šablony stylů
- Controllers: složka určená pro třídy typu Controller (potomky třídy Controller), zpracovávají požadavky uživatele na konkrétní URL adresy, přijímají parametry z URL adresy, se kterými poté pracují
- fonts: složka pro fonty použité ve webové aplikaci
- Models: obsahuje třídy typu Model nebo ViewModely, reprezentují objekty, se kterými je ve webové aplikaci manipulováno
- Scripts: obsahuje soubory typu skript (JavaScript, Bootstrap skripty, jQuery)
- Views: obsahuje .cshtml soubory které slouží k zobrazování a vyplňování dat webové aplikace v podobě HTML
- packages.config: složka řízená pomocí NuGet, obsahuje informace o tom, jaké balíčky a verze daných balíčků jsou v projektu používány

6 ALTERNATIVNÍ TECHNOLOGIE

Active Server Pages (ASP) není jedinou existující technologií pro vytvoření webové aplikace pomocí objektově relačního mapování. Kapitola alternativní technologie zahrnuje stručný popis vytvoření projektu pomocí technologie Java Enterprise Edition s použitím Hibernate jako objektově relačního mapování.

6.1 Java

Programovací jazyk Java patří mezi objektově orientované. Společně s programovacím jazykem C# patří mezi nejpobulárnější na světě. Java patří mezi interpretované jazyky, nevzniká tedy strojový kód ale bajtkód. Z tohoto důvodu Java disponuje možností pracovat na jakémkoliv zařízení obsahující interpret jazyka Java. Nezávislost na architektuře koncového zařízení je jeden z hlavních důvodů popularity jazyka Java.

6.2 Hibernate

Hibernate patří mezi ORM frameworky vytvořené pomocí jazyka Java. Hibernate převádí objekty jazyka Java na relační pomocí mapovacích souborů nebo anotací. Při použití Hibernate nelze použít Enterprise Java Beans (EJB), které slouží jako moduly ze kterých je složená výsledná webová aplikace ale pouze POJO (Plain Old Java Object) třídy obsahující konstruktor a metody pro získání nebo nastavení nové hodnoty. Dotazy nad objekty uloženými v databázi jsou prováděny pomocí jazyka Hibernate Query Language (HQL) který je velmi podobný jazyku SQL.

6.2.1 Mapovací soubory

Mapovací soubory patří mezi XML soubory, ve kterých probíhá mapování pomocí značek (tagů). Každá třída vlastní svůj mapovací soubor.

```
<hibernate-mapping package="interpretModel">
  <class name="Interpret" table="INTERPRET">
    <id name="interpretId" column="IDINTERPRET" type="int">
      <property name="nazev" column="NAZEV" not-null="true" unique="true"></property>
      <set name="prispevky">
        <key column="ID"> IDINTERPRET </key>
        <one-to-many class="Prispevek" />
      </set>
    </class>
</hibernate-mapping>
```

Obrázek 24 - Hibernate mapovací soubory

6.2.2 Anotace

Anotace plní roli dodatečných informací nad vybranými atributy nebo metodami třídy. Mapování pomocí anotací je velmi podobné mapování při použití Entity Frameworku.

```
@Entity(access = AccessType.PROPERTY)

@Table (name="INTERPRET")

public class Interpret {

    private int Id,

    private String nazev;

    @Id (generate = GenerationType.AUTO)

    public int getId() {

        return id;

    }

    @Column (length=100)

    public String getNazev() {

        return nazev;

    }

    public void setNazev(String nazev) {

        this.nazev = nazev;

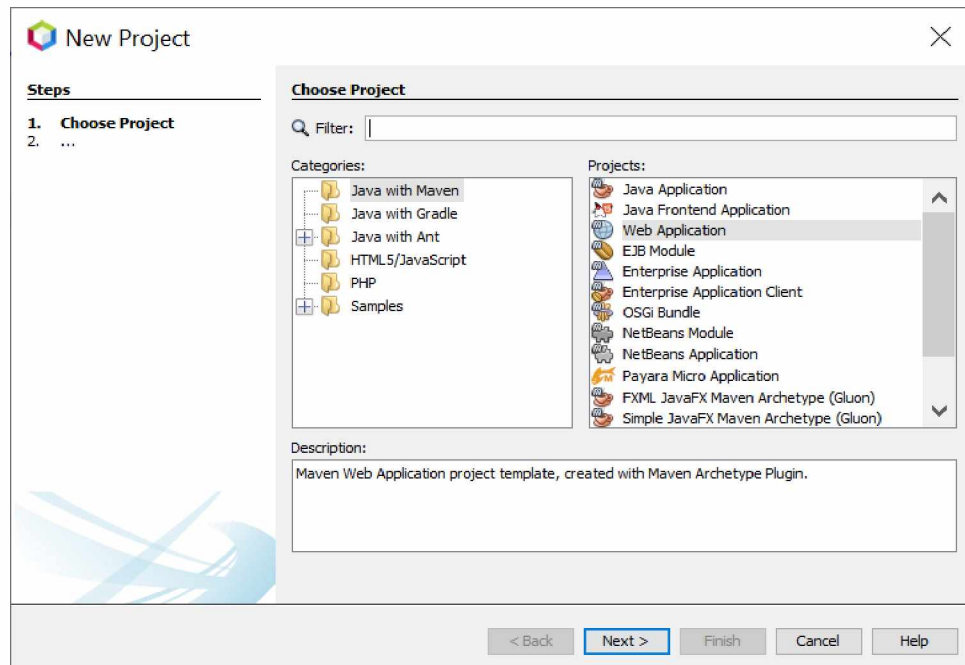
    }

}
```

Obrázek 25 - Hibernate anotace

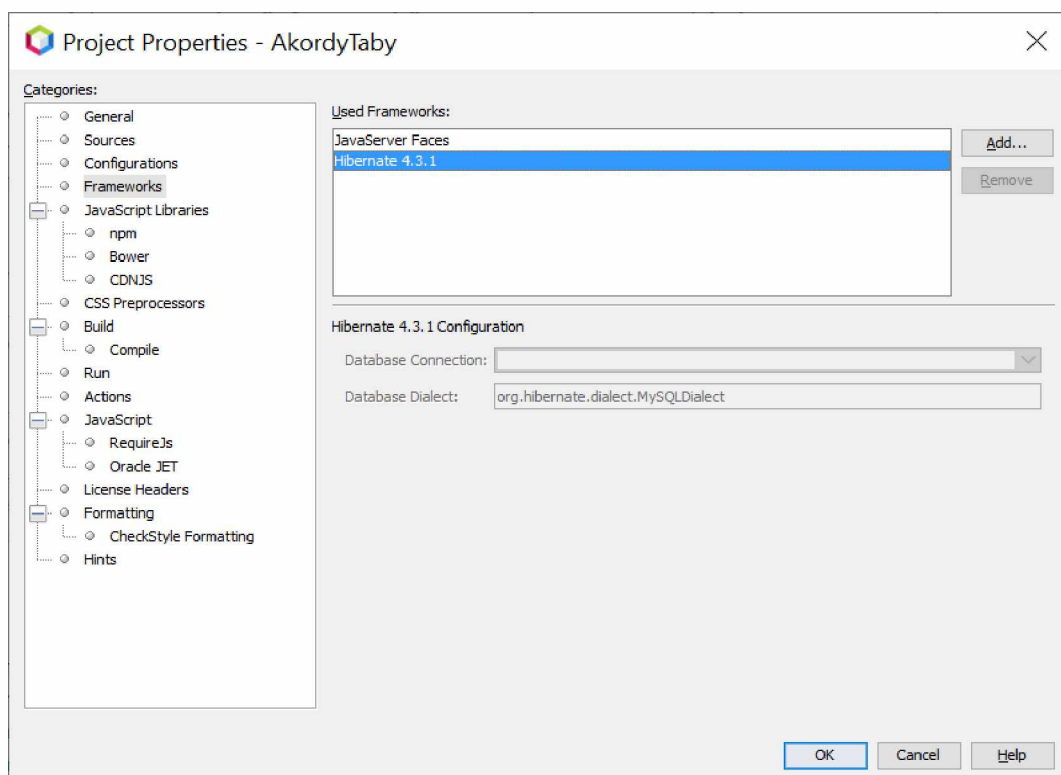
6.3 Vytvoření projektu

Pro vytvoření projektu bylo použito vývojové prostředí NetBeans. Při vytvoření nového projektu vývojář zvolí možnost Web Application a zvolí aplikační server který bude použit při tvorbě projektu (v tomto případě aplikační server GlassFish).



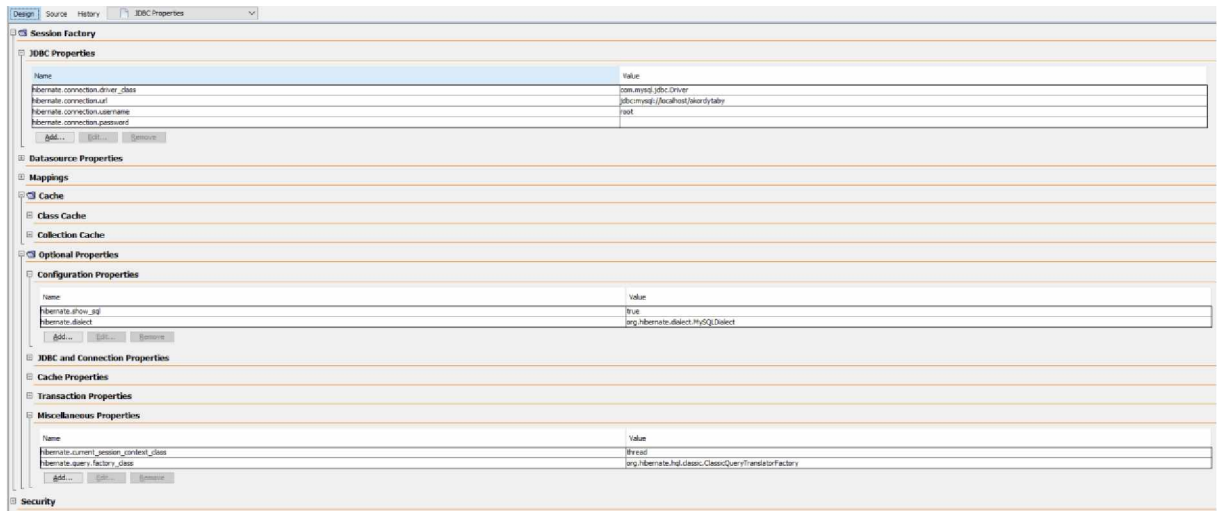
Obrázek 26 - vytvoření webové aplikace pomocí NetBeans IDE

Dalším krokem je importování potřebných knihoven (frameworků) do vytvořeného projektu. V našem případě vložení JavaServer Faces a Hibernate. Při vkládání knihovny Hibernate musí být specifikováno připojení k databázi. Mezi specifikaci patří typ ovladače a adresa databázové serveru.



Obrázek 27 - vložení knihoven do vytvořeného projektu

Po úspěšném vložení knihovny Hibernate do projektu je vytvořen soubor hibernate.cfg.xml obsahující konfiguraci pro konkrétní projekt. Soubor je možné editovat přímo pomocí XML nebo pomocí GUI editoru.



Obrázek 28 - konfigurační soubor Hibernate (GUI)

Nastavené vlastnosti jsou poté uloženy v podobě XML značek.

```
<hibernate-configuration>
```

```
  <session-factory>
```

```
    <property name="hibernate.show_sql">true</property>
```

```
    <property
```

```
name="hibernate.query.factory_class">org.hibernate.hql.classic.ClassicQueryTranslatorFactory</property>
```

```
    <property name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>
```

```
    <property
```

```
name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>
```

```
    <property
```

```
name="hibernate.connection.url">jdbc:mysql://localhost/akordytaby</property>
```

```
    <property name="hibernate.connection.username">root</property>
```

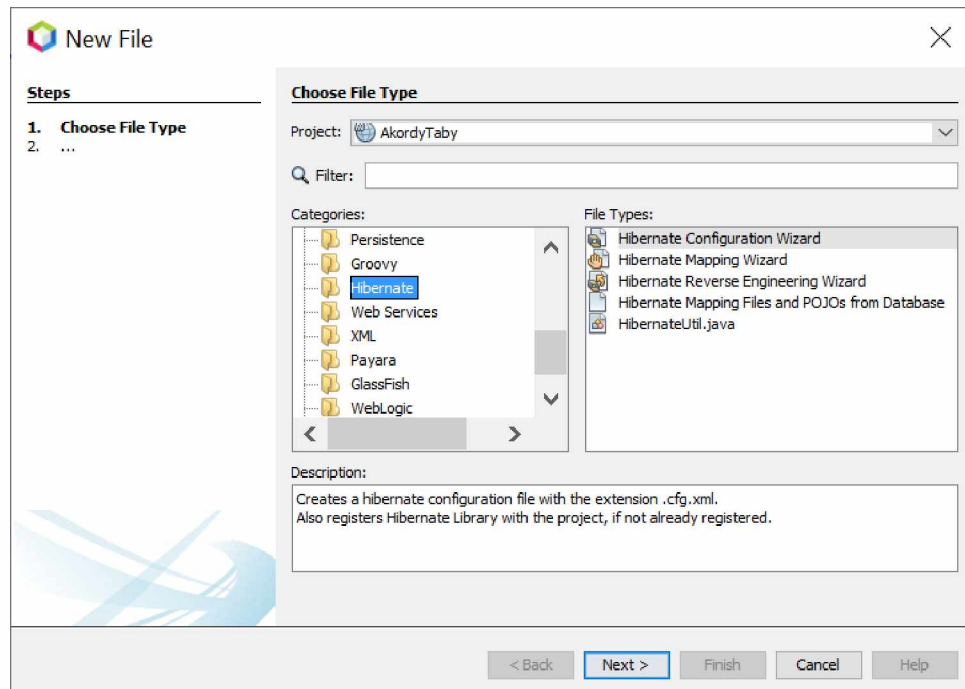
```
    <property name="hibernate.connection.password"/>
```

```
  </session-factory>
```

```
</hibernate-configuration>
```

Obrázek 29 - konfigurační soubor Hibernate (XML)

Dalším krokem je vytvoření HibernateUtil třídy nutné správnou funkčností knihovny Hibernate. Třída řídí přístup k SessionFactory manipulující s daty.



Obrázek 30 - Hibernate dialog

Pokud vše proběhlo úspěšně, je možné vytvořit POJO třídu a pomocí Hibernate Mapping Wizard provést mapování nad vytvořenou třídou. Nad vytvořenou a namapovanou třídou je možné vytvářet metody pro získání a ukládání dat pomocí SessionFactory.

```
public List vratPisne(int interpretId) {
    List<Pisne> pisne = null;
    try {
        org.hibernate.Transaction transakce = session.beginTransaction();
        Query dotaz = session.createQuery ("from Pisne where pisne.InterpretId '"+
interpretId +"'");
        pisne = (List<Pisne>) q.list();
    } catch (Exception e) {
        e.printStackTrace();
    }
    return pisne;
}
```

Obrázek 31 - manipulace s daty pomocí HQL

6.4 Porovnání implementačních technologií

Programovací jazyk C# i Java patří mezi silně typované objektově orientované s automatickou správou paměti. Problém, který lze vyřešit pomocí jazyka C# lze vyřešit také pomocí jazyka Java. Jazyk C# patří mezi jazyky používající kompilátor, který je rychlejší než jazyk Java používající interpret. Výběr programovacího jazyka může ovlivnit použití navazujících technologií a knihoven, například objektově relační mapování ale i preference vývojáře.

Při porovnání ORM knihoven Entity Framework a Hibernate existují rozdíly větší. Mezi výhody Entity Framework patří přehledná stručná dokumentace a velké množství tutoriálů pro začínajícího vývojáře. Jednoduché nasazení knihovny do projektu, možnost grafického rozhraní, více typů přístupu vytváření objektově relačního mapování. Pro správné použití Entity Framework je základ znát a rozumět programovacímu jazyku C#. Pomocí C# jsou psány třídy, ke kterým jsou vloženy ORM anotace pro správné převedení na databázový model (v případě Code First přístupu). Pro manipulaci s daty je dostačující znalost jazyka LINQ, který je zabudovaný již v základu .NET knihovny a běžně používán. Entity Framework podporuje postupně databázové migrace, které lze provádět manuálně ale také automaticky. Migrace slouží k synchronizaci mapovaných tříd s již vytvořenou databází. Mezi další výhody patří vnořený Identity kontext umožňující správu uživatelů, rolí, práv, přístupů, více faktorovou autentizaci, verifikační algoritmy a mnoho dalšího. Entity Framework umožňuje rychlý start novým projektům a přátelské prostředí novým vývojářům.

Pro správnou funkci Hibernate na rozdíl od Entity Framework je nutné vytvářet velké množství typů tříd které spravují samotný framework (například (HibernateUtil, Hibernate Reverse Engineering File, Mapping Files and POJOs from a Database). Dalším krokem je nastavení velkého množství vlastností pro základní konfigurační soubor (hibernate.cfg.xml), které je zapotřebí vyhledat v dlouhé dokumentaci pro Hibernate. Pro manipulaci s daty pomocí Hibernate je používán jazyk HQL, který nemá mimo Hibernate využití na rozdíl od LINQ u Entity Framework. Mezi nevýhody Hibernate patří velké množství času nutné k osvojení a použití knihovny ale také výkon, který je menší než při použití klasického SQL příkazu z důvodu generování velkého množství SQL příkazů na pozadí.

7 BEZPEČNOST APLIKACE A DATABÁZE

Následující kapitola obsahuje nejčastější typy útoků na webové aplikace a databáze.

7.1 SQL injection

SQL injection patří mezi techniky napadení webové aplikace pomocí databázové vrstvy. Napadení je uskutečněno pomocí podstrčení dat prostřednictvím vyplněného formuláře ale také URL adresy. Úspěšnost provedení útoku je závislá nejčastěji při neošetření uživatelských vstupů. Vyšší programovací jazyky umožňují ošetřit uživatelské vstupy různými způsoby. Programovací jazyk C# disponuje třídou SqlCommand, obsahující vlastnost Parameters. Do vlastnosti Parameters jsou vkládány jednotlivé parametry metody. Pomocí vlastnosti Parameters dochází k oddělenému předávání parametrů.

V případě použití objektově relačního mapování (například Entity Framework) je možné kompletně vynechat ruční psaní SQL příkazů. V tomto případě je pracování s databázovou tabulkou jako s kolekcí objektů. Pro práci s kolekcí objektů je optimální použití dotazovacího jazyka LINQ, umožňující použití jednotné syntaxe pro přístup k datům. Při použití jazyka LINQ vývojář nepřímým způsobem využívá dříve zmíněnou vlastnost Parameters.

Pro jazyk Java existuje velmi podobné řešení pomocí třídy PreparedStatement.

7.2 Cross-site scripting

Cross-site scripting (XSS) patří mezi metody útoku využívající neošetřené vstupy ve scriptu. Na rozdíl od SQL injection který probíhá na serveru webové aplikace, XSS útok je prováděn na klientské části. Útočník disponuje možností vložit a spustit vlastní script. Obrana proti XSS útoku spočívá v ošetření uživatelských vstupů, například odfiltrování nebezpečných vstupů z uživatelského vstupu (převedení HTML elementů). Další možností je vypnutí JavaScriptu, který je ale velmi často použit při vývoji webové aplikace a tím může nastat i kompletní nefunkčnost webové aplikace.

7.3 Odepření služby

Cílem odepření služby (DDoS útok) je zpomalit až znepřístupnit webovou aplikaci uživatelům pomocí přehlcení serveru požadavky. Požadavky nejčastěji pochází z infikovaných koncových zařízení infikované virem (obvykle trojský kůň). Účinná obrana je analýza datového provozu a přesměrování nebo vyloučení nežádoucích požadavků.

7.4 Šifrování přenášených dat

Šifrování obsahu znemožňuje modifikaci obsahu přenášených dat a tím dochází k jejich ochraně. Pro šifrování ve webových aplikacích je použita vrstva SSL (Secure Sockets Layer) nebo modernější verze TLS (Transport Layer Security). Jedná se o šifrovací protokoly spolupracující s protokolem transportní vrstvy TCP (Transport Layer Security). Použití šifrovacích protokolů je možné rozpoznat podle URL adresy, která bude obsahovat protokol HTTPS.

7.5 Nedostatečná autentizace

Jedna z dalších možností napadení webové aplikace může probíhat pomocí uživatelských účtů. Při nedostatečném zabezpečení uživatelských účtů může dojít k zneužití útočníkem. Průběh zneužití účtu je velmi závislý typu webové aplikace a roli uživatele. Nedostatečně silné heslo patří mezi důvody, proč uživatel nemůže mít přiděleny nadbytečná práva. Základní obrana je dostatečně silné heslo. Mezi pravidla pro silné heslo patří:

- dlouhé heslo (alespoň 12 znaků)
- neopakující se znaky
- použití čísel, speciálních znaků
- každé heslo použít pouze jednou

Síla hesla musí být ověřena již při vytváření účtu, ale také i při jeho obnově. Zmíněná pravidla nemají význam, pokud uživatel využije možnost zapamatovaného hesla v prohlížeči.

7.6 Autorizace

Autorizace je ve většině případů vázána na autentizaci. Autentizace slouží k jednoznačnému určení přihlášeného uživatele, zatímco autorizace se zabývá přidělením práv. Pro provedení operace je zapotřebí, aby přihlášený uživatel disponoval potřebnými právy. Větší systémy obsahují role, které jsou množinou práv. Například role administrátor obsahuje veškerá práva základních operací pro čtení, editaci i odebrání.

8 BPMN

Business Process Model and Notation je používán pro modelování podnikových procesů pomocí diagramů. Prvky, ze kterých je model složený lze rozdělit do základních skupin.

8.1 Tokové objekty

Tokové objekty definují chování podnikových procesů. Mezi tokové objekty patří události, které ovlivňují tok procesu. Události rozlišujeme počáteční (spouštěč procesu), průběžná (mezi počátkem a koncem) a konečná (výsledek aktivity). Aktivity zobrazují činnosti vykonané v rámci procesu. Aktivita může být jednoduchá nebo složená z procesu (podproces). Poslední tokový objekt je brána, která umožňuje větvit ale také sloučit tok procesu.^[19]

8.2 Plavecké dráhy

Plavecké dráhy nemají přímý vliv na tok procesu, ale umožňují odlišit účastníky procesu. Bazén ohraničuje hranice procesu. Vzájemná komunikace mezi bazény probíhá pomocí odesílání a přijímání zpráv. Bazén obsahuje jednu nebo více drah. Dráhu lze využít pro organizaci aktivit bazénu. Pro komunikaci mezi dráhami slouží sekvenční tok, patřící mezi spojovací objekty.^[19]

8.3 Spojovací objekty

Využití spojovacích objektů spočívá v propojování jednotlivých objektů modelu. Nejčastěji používaný je sekvenční tok, který určuje pořadí aktivit, ve kterém budou prováděny. Druhý spojovací objekt je tok zpráv. Tok zpráv zobrazuje komunikaci mezi účastníky (například bazény). Poslední spojovací objekt je asociace, která je využívána k propojení artefaktů.^[19]

8.4 Artefakty

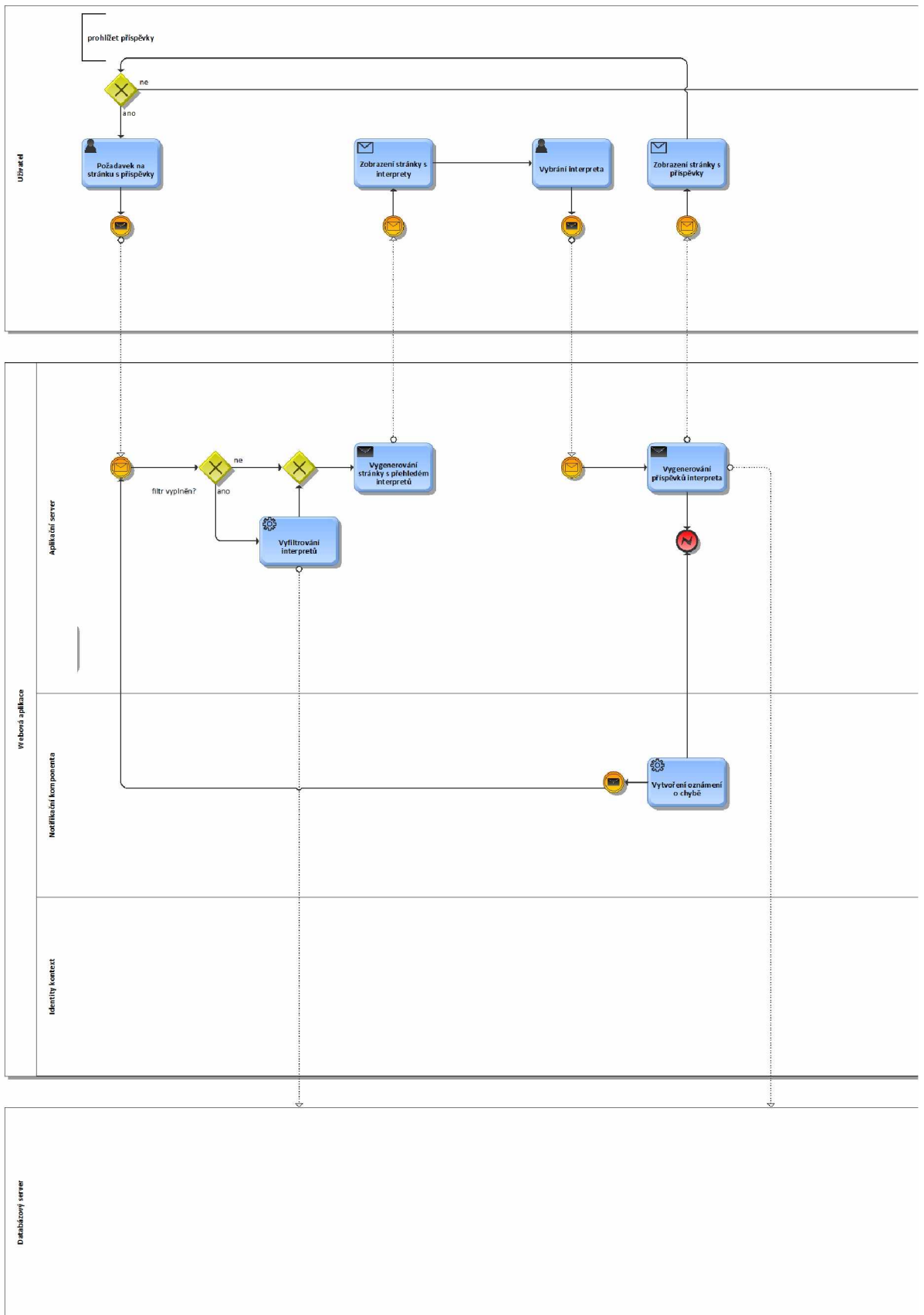
Artefakty upřesňují informace týkající se procesu, žádným způsobem neovlivňují tok procesu. Mezi nejčastěji používané artefakty patří skupina a anotace. Připojení artefaktů k aktivitám probíhá pomocí asociační vazby.^[19]

8.5 Ukázka BPMN

Ukázka BPMN zobrazuje vybrané modelované procesy, které je možné provést ve výsledné webové aplikaci.

8.5.1 Přihlášení uživatele

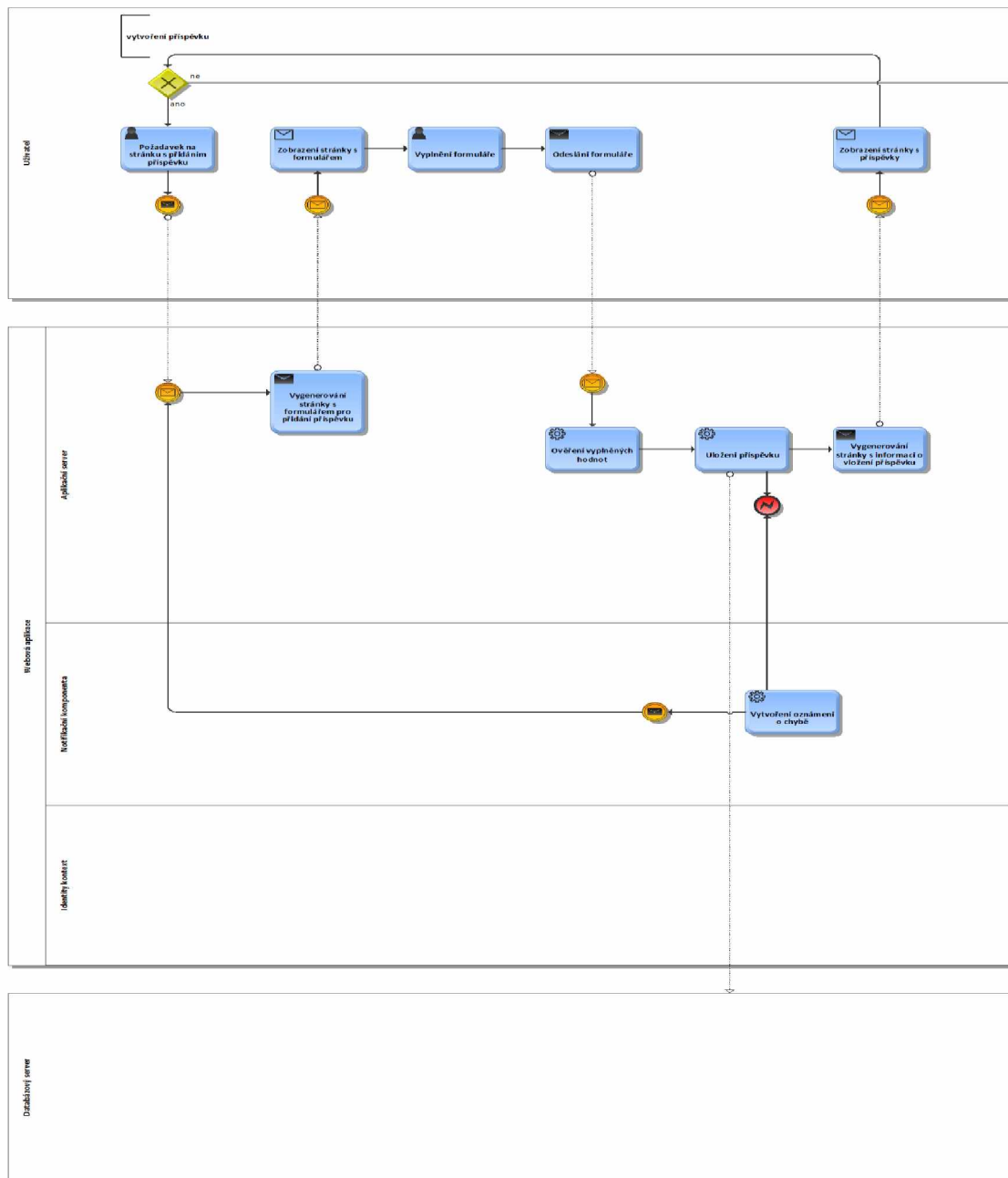
Model zobrazuje sekvenci aktivit při přihlášení uživatele. Pro přihlášení je nezbytná registrace a potvrzení platné emailové adresy. Proces je ukončen v případě nevytvořeného účtu uživatele a odmítnutí registrace. Součástí modelu je ošetření možné chyby při registraci a přihlášení



Obrázek 33 - model prohlížení příspěvků

8.5.3 Vytvoření příspěvku

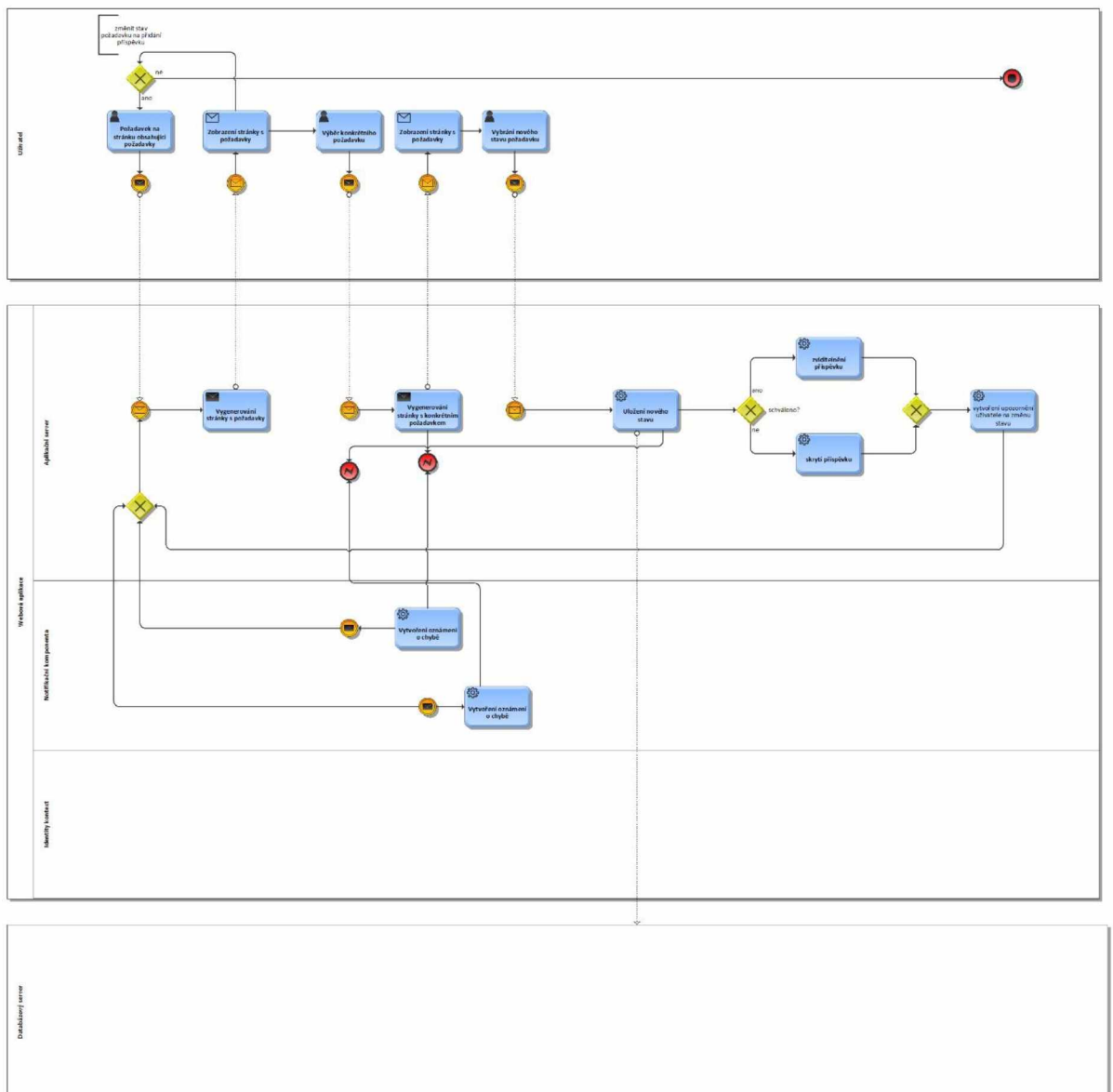
Při požadavku o vytvoření příspěvku je uživateli zobrazen formulář obsahující potřebné údaje pro úspěšné vytvoření příspěvku. Mezi povinné údaje patří název příspěvku, typ a obsah. Po odeslání formuláře nastane vyhodnocení vyplněných hodnot. V případě úspěšného vyhodnocení je příspěvek uložen a správci interpreta odeslána žádost o schválení příspěvku. Pokud při ověřování zadaných hodnot nastane chyba, příspěvek nebude uložen, vyplněný obsah však zůstane ponechán a uživatel je informován o potřebných úpravách.



Obrázek 34 - model vytvoření příspěvku

8.6 Změna stavu požadavku o příspěvek

Změna stavu požadavku o příspěvek zahrnuje zobrazení výpisu přichozících požadavků čekajících na schválení. Z požadavků uživatel zvolí konkrétní, u kterého je zapotřebí změnit stav. Požadavek může být schválen za podmínky splnění veškerých pravidel o přidání příspěvku. V případě schválení je příspěvek zviditelněn ostatním přihlášeným uživatelům. Při změně stavu požadavku je vytvořeno oznámení určené pro autora příspěvku s informací o vyhodnocení žádosti. Důležité je také ošetření možného výskytu chyb při generování formuláře s konkrétní žádostí a při uložení nového stavu.



Obrázek 35 - model změny stavu požadavku

9 ZÁKLADY HUDEBNÍ TEORIE

Následující kapitola obsahuje stručné shrnutí informací z oblasti hudební teorie potřebné pro pochopení obsahu vyskytujícího se ve vytvořené webové aplikaci.

9.1 Tóny

Tón v hudební teorii označuje zvuk o určité frekvenci, hlasitosti a délce trvání. Zdrojem tónu mohou být například lidské hlasivky nebo hudební nástroj. Existuje základních 12 tónů.

Značka	C	C#	D	D#	E	F	F#	G	G#	A	B	H
Čtème	Cé	Cis	Dé	Dis	É	Ef	Fis	Gé	Gis	Á	Bé	Há

Tabulka 2 - výpis tónů

Výše zmíněných 12 tónů tvoří oktávu. Na začátku a konci oktávy je vždy totožný tón, rozdíl je možné slyšet pouze v jejich frekvenci. U kytary je frekvence tónu ovlivněna šířkou a napnutím struny.

Struna	E	A	D	G	H	E
Frekvence [Hz]	82.41	110	146.83	196	246.94	329.63

Tabulka 3 - frekvence základního ladění kytary

Následující tabulka zobrazuje frekvenční rozdíl mezi tóny samotnými a tóny v oktávě. Frekvence tónů se stejným názvem je při porovnání dvojnásobkem frekvence tónů v oktávě předchozí.^[20]

Tón	C	C#	D	D#	E	F	F#	G	G#	A	B	H
První	16,3	17,3	18,3	19,4	20,6	21,8	23,1	24,4	25,9	27,5	29,1	30,8
Druhá	32,7	34,6	36,7	38,9	41,2	43,7	46,2	49	51,9	55	58,3	61,7
Třetí	65,4	69,3	73,4	77,8	82,4	87,3	92,5	98	103,8	110	116,5	123,5

Tabulka 4 - porovnání frekvencí [Hz] mezi tóny a oktávami

9.2 Akordy

Akord je složený z tónů. V evropské hudbě je akord většinou složen právě ze tří tónů. Mezi základní druhy akordů patří dur a moll. Jedna z možností vytvoření akordu jsou masky. Pro zjištění dur akordu postačí aplikovat masku 0-4-7 na naši tabulku tónů. Například potřebuji-li

zjistit akord C-dur pak aplikuji masku pomocí vybrání nultého tónu, který označuje vždy název akordu (v případě akordu C-dur začínáme tónem C), poté odpočítáme tón čtvrtý a sedmý.

Značka	C	C#	D	D#	E	F	F#	G	G#	A	B	H
C dur	0	1	2	3	4	5	6	7	8	9	10	11
D dur	10	11	0	1	2	3	4	5	6	7	8	9
E dur	8	9	10	11	0	1	2	3	4	5	6	7
F dur	7	8	9	10	11	0	1	2	3	4	5	6
G dur	5	6	7	8	9	10	11	0	1	2	3	4
A dur	3	4	5	6	7	8	9	10	11	0	1	2

Tabulka 5 - příklady vyhledání dur akordů pomocí masky 0-4-7

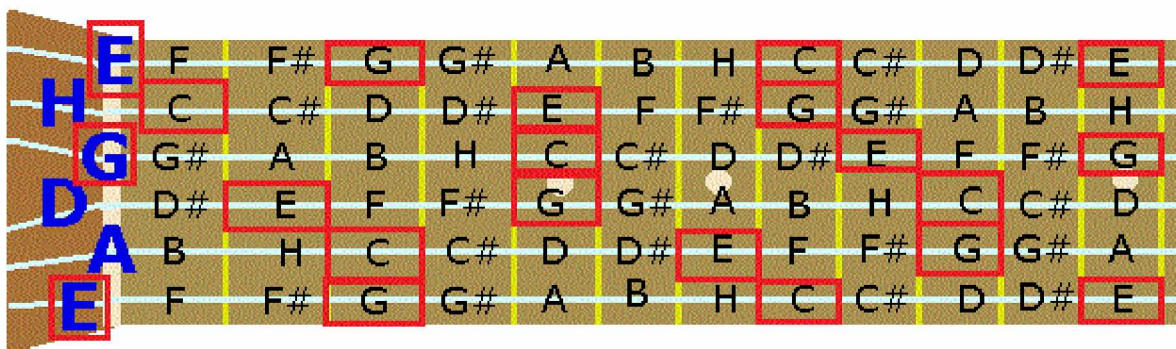
V případě vyhledání moll akordů postačí aplikovat téměř totožný typ masky – 0-3-7.

Značka	C	C#	D	D#	E	F	F#	G	G#	A	B	H
C dur	0	1	2	3	4	5	6	7	8	9	10	11
D dur	10	11	0	1	2	3	4	5	6	7	8	9
E dur	8	9	10	11	0	1	2	3	4	5	6	7
F dur	7	8	9	10	11	0	1	2	3	4	5	6
G dur	5	6	7	8	9	10	11	0	1	2	3	4

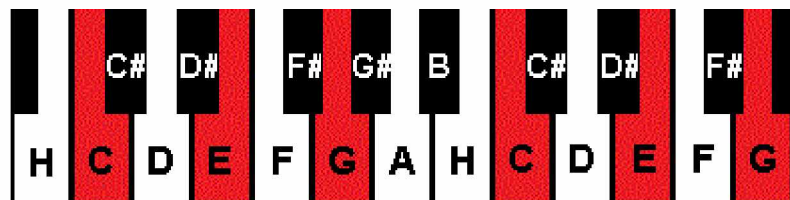
Tabulka 6 - příklady vyhledání moll akordů pomocí masky 0-3-7

Pomocí masky lze určit i jiné tvary akordů, pro tento účel postačí pouze dur a moll.

Úspěšně jsme zjistili jaký akord je složen z kterých tónů, nyní je potřeba znalost tónů aplikovat na hudební nástroj. Pokud opět použijeme akord C dur, zjistili jsme že je složen z tónů C, E a G. Následující krok znamená vyhledat konkrétní tóny na hudebním nástroji. Příklady vyhledání přiložený na následujících obrázcích.



Obrázek 36 - využití znalosti tónů v případě kytary



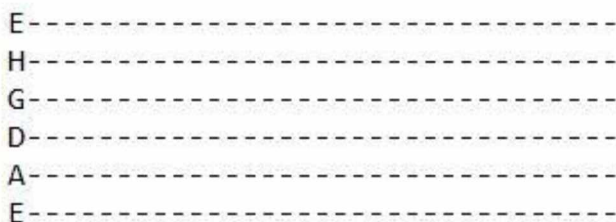
Obrázek 37 - využití znalosti tónů v případě kláves

Obrázky hmatníku kytary a kláves převzaty ze zdroje [21].

Uživatelé, kteří vyhledávají webové stránky s informacemi o tom jak zahrát konkrétní píseň, z velké části vyhledávají právě akordy. V případě akordů je očekávaná základní znalost hudební teorie uživatele. Pro uživatele hudební teorií nepolíbené zůstává možnost dohledávat podobu jednotlivých akordů a doučit se potřebné akordy dle potřeby nebo využít tabulatury u které není znalost hudební teorie vyžadována.

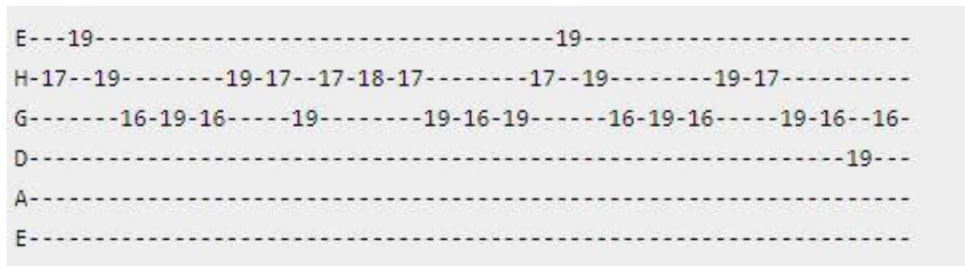
9.3 Tabulatury

Tabulatury jsou určeny pro různé pokročilé hudebníky. Začínajícím pomáhá hrát písně bez znalosti not či akordů, pokročilým k jednoduššímu zápisu tónů (například zapsání kytarového sóla písně). Tabulatura je tedy alternativa notového zápisu pro hráče na strunné nástroje. Na levé straně obsahuje písmena, které značí název struny. Pravá část je určena pro zápis samotné hodnoty.



Obrázek 38 - příklad prázdné tabulatury

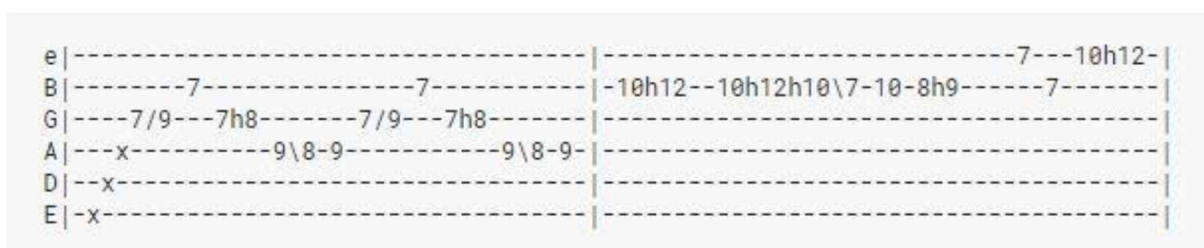
Tabulatury mohou být jednoduché tak i složité. Jednoduché tabulatury obsahují pouze čísla pražců kytary na kterém má být struna stisknuta. Na první pohled je jednoznačně určeno která struna musí být v jakém pořadí zmáčknuta. Tabulatury jsou čteny směrem zleva doprava.



Obrázek 39 - příklad jednoduché tabulatury

Pro začínajícího hudebníka je zápis čitelný, pokročilejší mohou pouze zaznamenat jaké tóny jsou drženy a určit na které struně a jakém pražci stiskne strunu.

Složitější tabulatury již obsahují informací více, například informace o vytažení struny, vibratu, tremolu, tappingu, slidu, tlumení struny a mnoho dalších možností.



Obrázek 40 - příklad složitější tabulatury

10 ANALÝZA PROJEKTU

Cílem práce je vytvoření funkční webové aplikace, která umožní uživateli registraci a přihlášení na vytvořený účet. Po úspěšné registraci uživatel obdrží nová práva, například přidávat příspěvky, možnost zažádat o roli redaktora, komunikovat s ostatními uživateli pomocí zpráv, okomentovat článek, obdržení hodnosti na základě uživatelovi aktivity a mnoho dalších možností, které budou popsány v kapitolách určených jako manuál pro správné používání webové aplikace. Hlavní funkcí webové aplikace je poskytnout začínajícím hudebníkům písně i s akordy nebo tabulaturami případně možnost kontaktovat zkušenější hudebníky, pro pokročilejší muzikanty je zde možnost podělit se o své zkušenosti se začátečníky.

10.1 Požadavky

Požadavky na systém znamenají schopnosti, které musí být implementovány do výsledné webové aplikace. Mohou být považovány za vlastnosti systému umožňující systému zobrazovat, přijímat a ukládat požadovaná data dle pokynů zadavatele systému. Rozdělení požadavků je na funkční a nefunkční. Funkční požadavky obsahují, které funkce systém potřebuje mít implementován (události, akce, aktivity). Nefunkční požadavky jsou požadavky kladoucí omezení, například na výkonnost systému nebo dobu odezvy. Základními entitami projektu je uživatel používající jednotlivé části webové aplikace, interpret obsahující jednotlivé příspěvky, samotné příspěvky obsahující typ zda se jedná o akordy nebo tabulaturu, články které jsou rozděleny do jednotlivých kategorií, zpráva sloužící ke komunikaci mezi uživateli a na závěr žádosti pomocí kterých je funkční systém přidávání příspěvků, článků a přidělování funkcí uživatelům.

ID	Popis	Entita	Typ
1	Systém bude umožňovat vytvoření nového uživatele	Uživatel	Funkční
2	Systém bude umožňovat editaci uživatele	Uživatel	Funkční
3	Systém bude umožňovat odebrání uživatele	Uživatel	Funkční
4	Systém bude umožňovat přidělení role uživateli	Uživatel	Funkční
5	Systém bude umožňovat komunikaci mezi uživateli	Uživatel	Funkční
6	Systém bude umožňovat uživateli přidat komentář pod profil jiného uživatele	Uživatel	Funkční
7	Systém bude umožňovat odebrání role uživateli	Uživatel	Funkční
8	Systém bude umožňovat přidělit uživateli hodnocení jeho aktivity	Uživatel	Funkční

9	System bude umožňovat nahrát profilové foto	Uživatel	Funkční
10	System bude umožňovat zobrazit veškeré příspěvky uživatele	Uživatel	Funkční
11	System bude umožňovat vyhledání uživatele dle jména	Uživatel	Funkční
12	System bude umožňovat zobrazení profilu uživatele	Uživatel	Funkční
13	System bude umožňovat přihlášení uživatele	Uživatel	Funkční
14	System bude umožňovat uživateli kontaktovat podporu	Uživatel	Funkční
15	System bude umožňovat zobrazit uživateli veškeré jeho žádosti	Uživatel	Funkční
16	System bude umožňovat ověření datového typu a omezení velikosti nahrané profilové fotky	Uživatel	Funkční
17	System bude umožňovat obnovení hesla uživateli	Uživatel	Funkční
18	System bude vyžadovat potvrzení emailu před prvním přihlášením	Uživatel	Funkční
19	System bude umožňovat uživateli zobrazit časté otázky (FAQ)	Uživatel	Funkční
20	System bude umožňovat odkazem přesměrovat na skupiny na sociálních sítích	Uživatel	Funkční
21	System bude umožňovat zobrazit veškeré články uživatele	Uživatel	Funkční
22	System bude umožňovat zobrazit na profilu uživatele základní informace uvedené při registraci	Uživatel	Funkční
23	System bude umožňovat zobrazení interpretů jejichž je uživatelem správcem	Uživatel	Funkční
24	System bude umožňovat přidat správci nového správce interpreta	Uživatel	Funkční
25	System bude umožňovat odebrat správci interpreta jiného správce	Uživatel	Funkční
26	System bude umožňovat zobrazení veškerých správce interpreta	Uživatel	Funkční
27	System bude zobrazovat počet příspěvku daného interpreta	Uživatel	Funkční
28	System bude zobrazovat počet schválených příspěvků daného interpreta	Uživatel	Funkční
29	System bude umožňovat vytvoření nového interpreta	Interpret	Funkční

30	System bude umožňovat editaci interpreta	Interpret	Funkční
31	System bude umožňovat odebrání interpreta	Interpret	Funkční
32	System bude uživateli vracet zpětnou vazbu o úspěšném vytvoření interpreta	Interpret	Funkční
33	System bude umožňovat zobrazit informace o konkrétním interpretovi	Interpret	Funkční
34	System bude umožňovat filtrovat interprety dle názvu	Interpret	Funkční
35	System bude umožňovat zobrazit příspěvky daného interpreta	Interpret	Funkční
36	System bude umožňovat zobrazení nejnovějších interpretů	Interpret	Funkční
37	System bude umožňovat zobrazit správce interpreta	Interpret	Funkční
38	System bude umožňovat zobrazit pár příspěvku daného interpreta již při výpisu interpretů	Interpret	Funkční
39	System bude umožňovat vytvoření příspěvku	Příspěvek	Funkční
40	System bude umožňovat editaci příspěvku	Příspěvek	Funkční
41	System bude umožňovat odebrání příspěvku	Příspěvek	Funkční
42	System bude umožňovat ohodnocení příspěvku přihlášeným uživatelem	Příspěvek	Funkční
43	System bude umožňovat vyhledání příspěvku	Příspěvek	Funkční
44	System bude umožňovat filtrování příspěvků	Příspěvek	Funkční
45	System bude umožňovat zobrazení nejnovějších příspěvků	Příspěvek	Funkční
46	System bude umožňovat zobrazení příspěvku	Příspěvek	Funkční
47	System bude umožňovat zobrazení veškerých příspěvků	Příspěvek	Funkční
48	System bude umožňovat vybrat typ vkládaného příspěvku	Příspěvek	Funkční
49	System bude zobrazovat zpětnou vazbu uživateli o úspěšném vložení příspěvku	Příspěvek	Funkční
50	System bude umožňovat zobrazit hodnocení příspěvku	Příspěvek	Funkční
51	System bude umožňovat zobrazit profil autora příspěvku	Příspěvek	Funkční
52	System bude umožňovat zobrazit kdy byl příspěvek přidán	Příspěvek	Funkční
53	System bude umožňovat zobrazení akordů přímo v příspěvku	Příspěvek	Funkční
54	System bude umožňovat vložit nový článek	Článek	Funkční
55	System bude umožňovat editovat vložený článek	Článek	Funkční

56	System bude umožňovat odebrat vložený článek	Článek	Funkční
57	System bude umožňovat vložit komentář pod vybraný článek	Článek	Funkční
58	System bude umožňovat vybrat kategorii článku	Článek	Funkční
59	System bude umožňovat vytvořit kategorii článku	Článek	Funkční
60	System bude umožňovat editovat kategorii článku	Článek	Funkční
61	System bude umožňovat odebrat kategorii článku	Článek	Funkční
62	System bude umožňovat filtrovat články dle názvu	Článek	Funkční
63	System bude umožňovat vyhledávat články	Článek	Funkční
64	System bude umožňovat zobrazit veškeré články	Článek	Funkční
65	System bude umožňovat zobrazit články dle kategorie	Článek	Funkční
66	System bude umožňovat zobrazit veškeré komentáře pod vybraným článkem	Článek	Funkční
67	System bude vracet uživateli zpětnou vazbu o úspěšném vložení článku	Článek	Funkční
68	System bude umožňovat zobrazení nejnovějších článků	Článek	Funkční
69	System bude v komentáři zobrazovat autora komentáře a čas přidání	Článek	Funkční
70	System bude umožňovat stránkování článků	Článek	Funkční
71	System bude umožňovat zobrazit celkový počet komentářů pod článkem	Článek	Funkční
72	System bude umožňovat odeslat žádost o vytvoření interpreta	Žádost	Funkční
73	System bude umožňovat odeslat žádost o vytvoření písňe	Žádost	Funkční
74	System bude umožňovat odeslat žádost na roli redaktora	Žádost	Funkční
75	System bude umožňovat uživateli změnit stav žádosti	Žádost	Funkční
76	System bude umožňovat přidat komentář k žádosti	Žádost	Funkční
77	System bude zobrazovat žádosti které již uživatel zpracoval	Žádost	Funkční
78	System bude zobrazovat žádosti přihlášeného uživatele které čekají na vyřízení	Žádost	Funkční
79	System bude umožňovat zobrazovat žádosti přihlášeného uživatele které již byly vyřízeny	Žádost	Funkční

80	System bude zobrazovat žádosti které čekají na zpracování uživatelem	Žádost	Funkční
81	System nebude zobrazovat příspěvky které nebyly schváleny	Žádost	Funkční
82	System bude zobrazovat upozornění na nevyřízenou žádost	Žádost	Funkční
83	System bude umožňovat zpětně změnit stav požadavku	Žádost	Funkční
84	System bude zobrazovat základní informace o požadavku	Žádost	Funkční
85	System bude umožňovat poslat uživateli zprávu	Zpráva	Funkční
86	System bude umožňovat přidat kontakt	Zpráva	Funkční
87	System bude umožňovat zobrazit profilové foto společné se zprávou	Zpráva	Funkční
88	System bude umožňovat přepínat mezi historií zpráv jednotlivých uživatelů	Zpráva	Funkční
89	System bude zobrazovat upozornění na nepřečtenou zprávu	Zpráva	Funkční
90	System bude rozlišovat odeslané a přijaté zprávy pomocí stran chatu	Zpráva	Funkční
91	System bude umožňovat zobrazení profilových fotek spolu se zprávou	Zpráva	Funkční
92	System bude umožňovat zobrazení času odeslání zprávy	Zpráva	Funkční
93	System bude řadit konverzace dle času poslední zprávy v konverzaci	Zpráva	Funkční
94	S 23 System bude rychle reagovat na požadavky uživatele	System	Nefunkční
95	S 24 System bude chránit uživatelská data	System	Nefunkční
96	S 25 System bude kontrolovat vstupy	System	Nefunkční
97	S 38 System bude zobrazovat možnosti pro uživatele dle přiřazené role	System	Nefunkční

Tabulka 7 - seznam požadavků

10.2 Aktéři

Aktéři patří mezi jednotky užívající výsledný systém. Většinou se jedná o uživatele, kteří jsou rozlišení pomocí rolí, které určují jakým způsobem mohou systém používat. Pro tuto práci byly nalezeny následující aktéři:

10.2.1 Nepřihlášený uživatel

Aktér s rolí nepřihlášeného uživatele má nejvíce omezení. Mezi možnosti nepřihlášeného uživatele patří pouze registrace, přihlášení a obnovení hesla. Po přihlášení uživatel nabude role Přihlášený uživatel.

10.2.2 Přihlášený uživatel

Přihlášený uživatel disponuje možnostmi v různých kategoriích. Následují vybrané příklady z nově dostupných práv:

- odeslání zprávy jinému uživateli
- odeslání žádosti o přidání příspěvku
- možnost žádosti o vytvoření interpreta
- možnost žádosti o roli redaktora
- možnost ohodnotit přidaný příspěvek
- možnost přidat komentář k článku
- kontaktování podpory

10.2.3 Správce interpreta

Správce interpreta patří mezi role nabyté po schválení žádosti o vytvoření interpreta. Správce interpreta schvaluje nebo zamítá příspěvky ostatních uživatelů, kteří mají zájem vložit nový příspěvek pod konkrétního interpreta. Jeden uživatel má možnost být správcem více interpretů. Správce interpreta může přidat nového nebo odebrat stávajícího správce. Jeden interpret tedy může mít více správců. Při dlouhodobé neaktivitě spojené s možností pouze jednoho správce interpreta může dojít k dlouhodobému nevyřízení nových žádostí o vložení příspěvku.

10.2.4 Redaktor

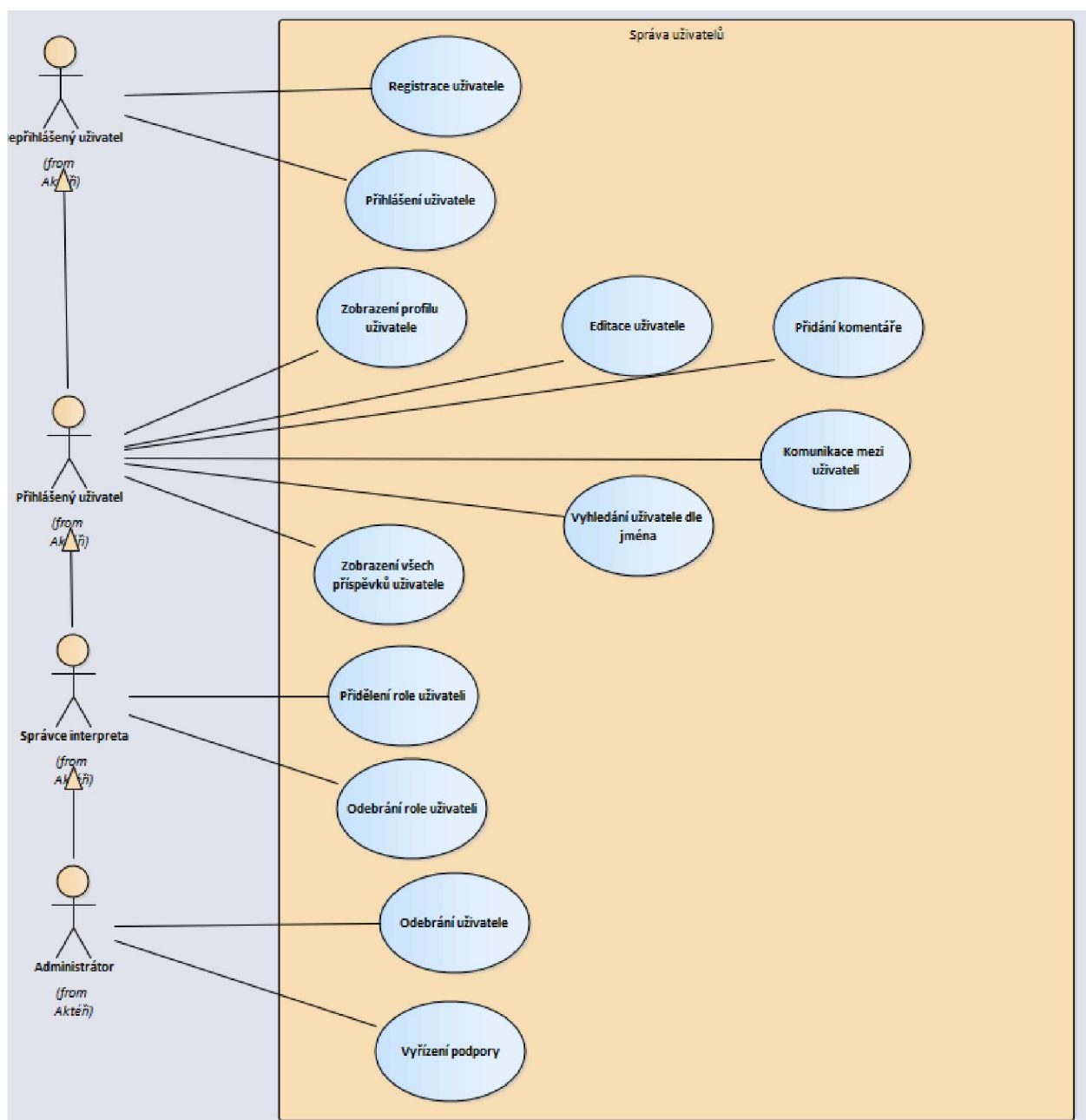
Roli redaktora je možné získat po schválení žádosti o roli redaktora. Role přidává možnost vkládat články které jsou zařazeny do kategorií.

10.2.5 Administrátor

Administrátor je nejvýše postavená role. Mezi povinnosti role patří například vyřizování zpráv odeslaných na podporu nebo schvalování žádostí o roli redaktora.

10.3 Případy užití

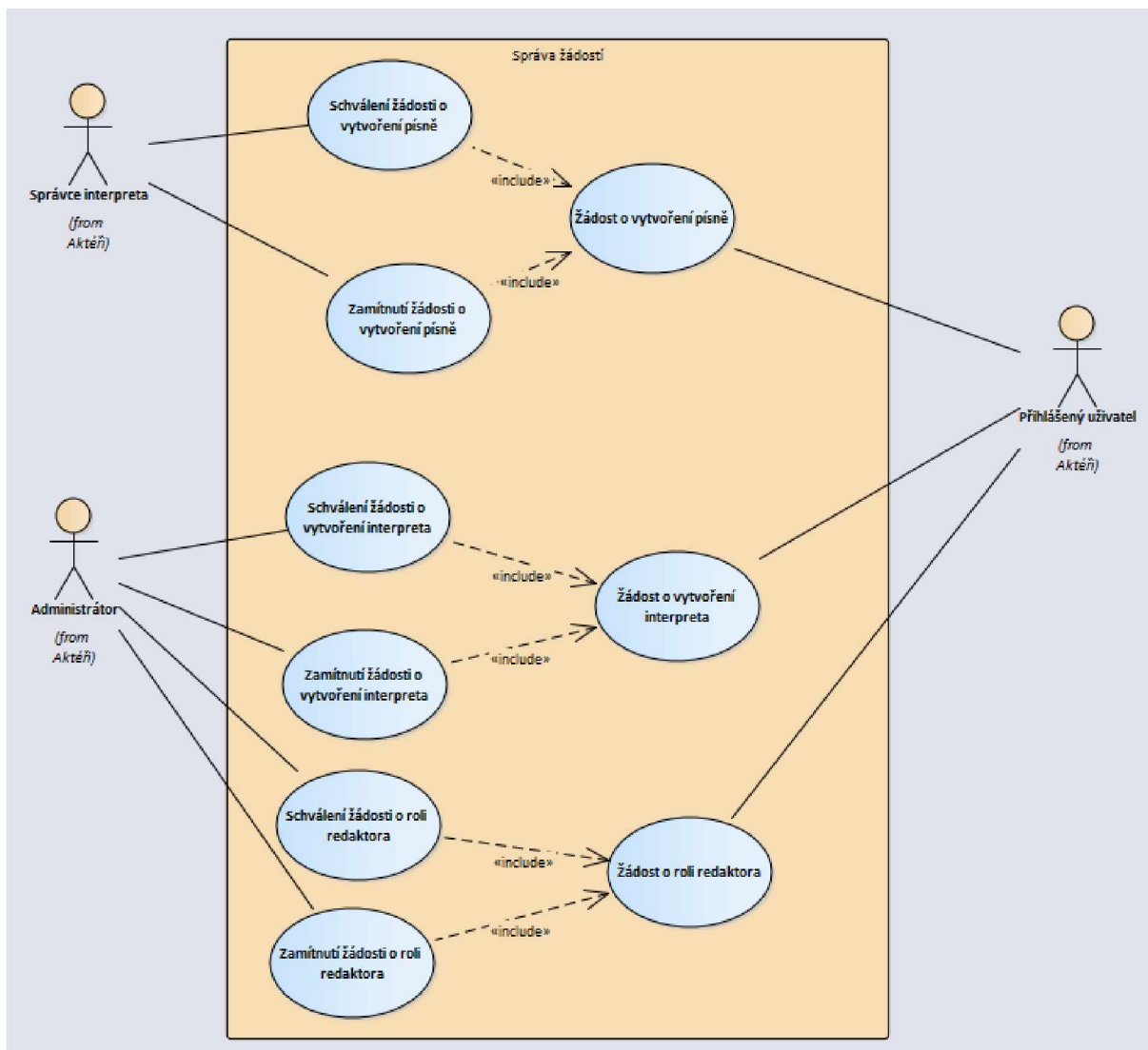
Případy užití představují realizaci požadavků přidělené jednotlivým aktérům. Následující příklady případů užití zobrazují práci se základními entitami projektu. První příklad zobrazuje pravomoci jednotlivých aktérů.



Obrázek 41 - případ užití – uživatelé

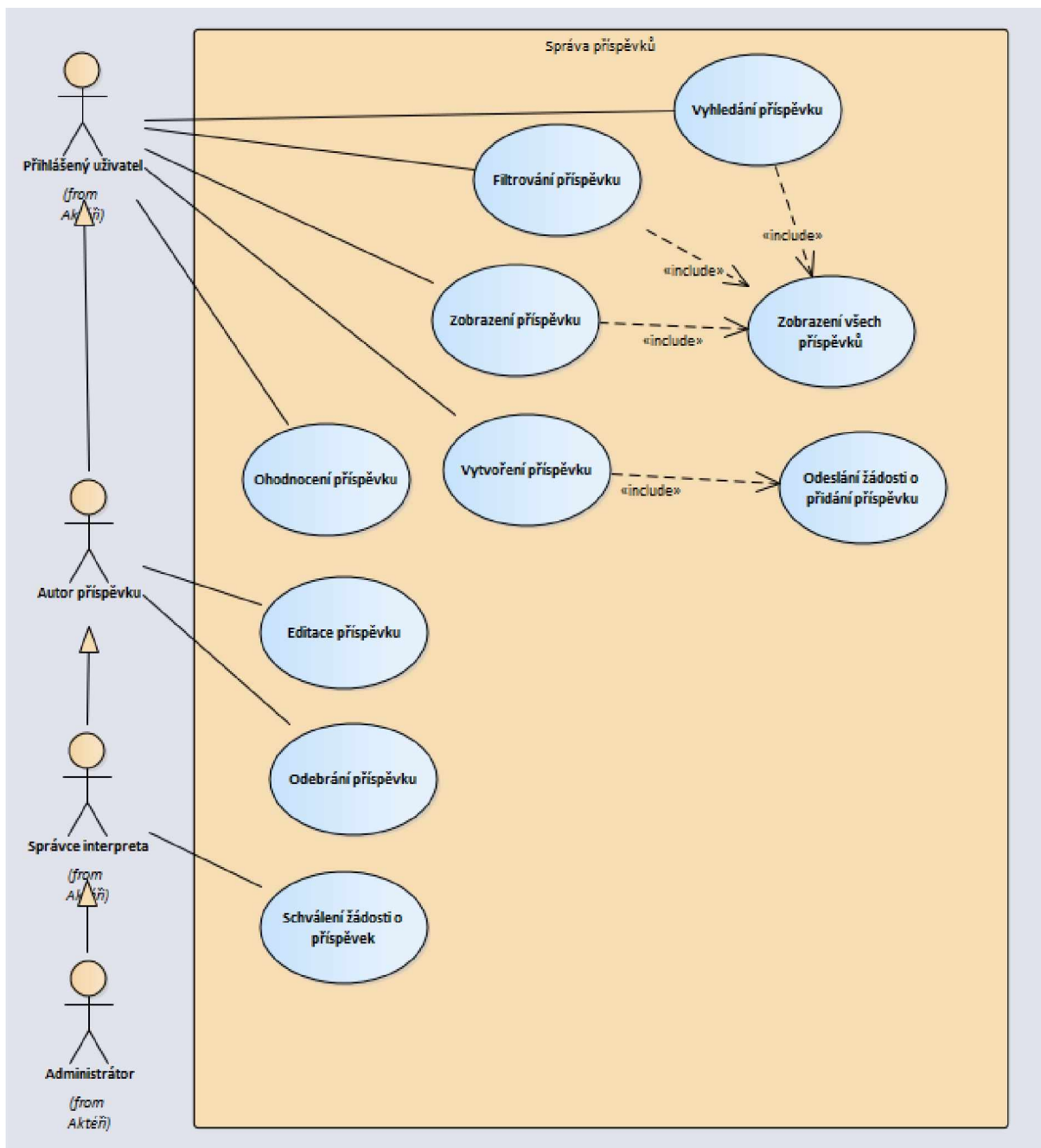
Druhý příklad užití zobrazuje pravomoci aktérů ohledně žádostí a jejich odesílání. Žádosti jsou rozděleny do tří základních typů:

- Žádost o vytvoření interpreta
- Žádost o vytvoření písničky
- Žádost o roli redaktora



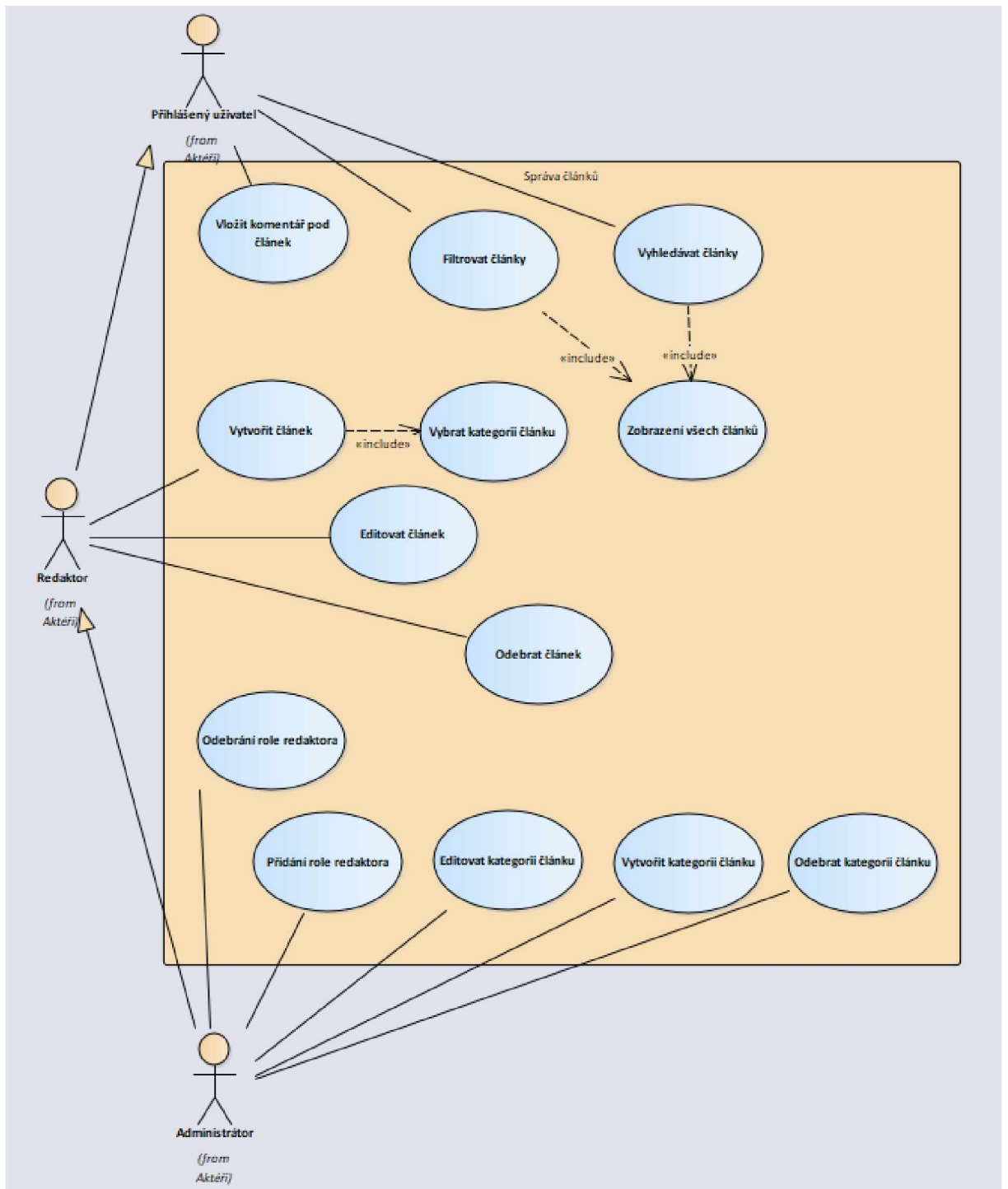
Obrázek 42 - případy užití – žádosti

Třetí příklad případů užití zobrazuje, jaké aktéři mají pravomoci týkající se příspěvků. Přihlášený uživatel disponuje možností vyhledání příspěvku, zobrazení příspěvku nebo jeho přidání. Každý příspěvek musí být schválen správcem daného interpreta. Po schválení příspěvku může přihlášený uživatel (v neoficiální roli autora příspěvku) příspěvek editovat nebo odebrat.



Obrázek 43 - případy užití – příspěvky

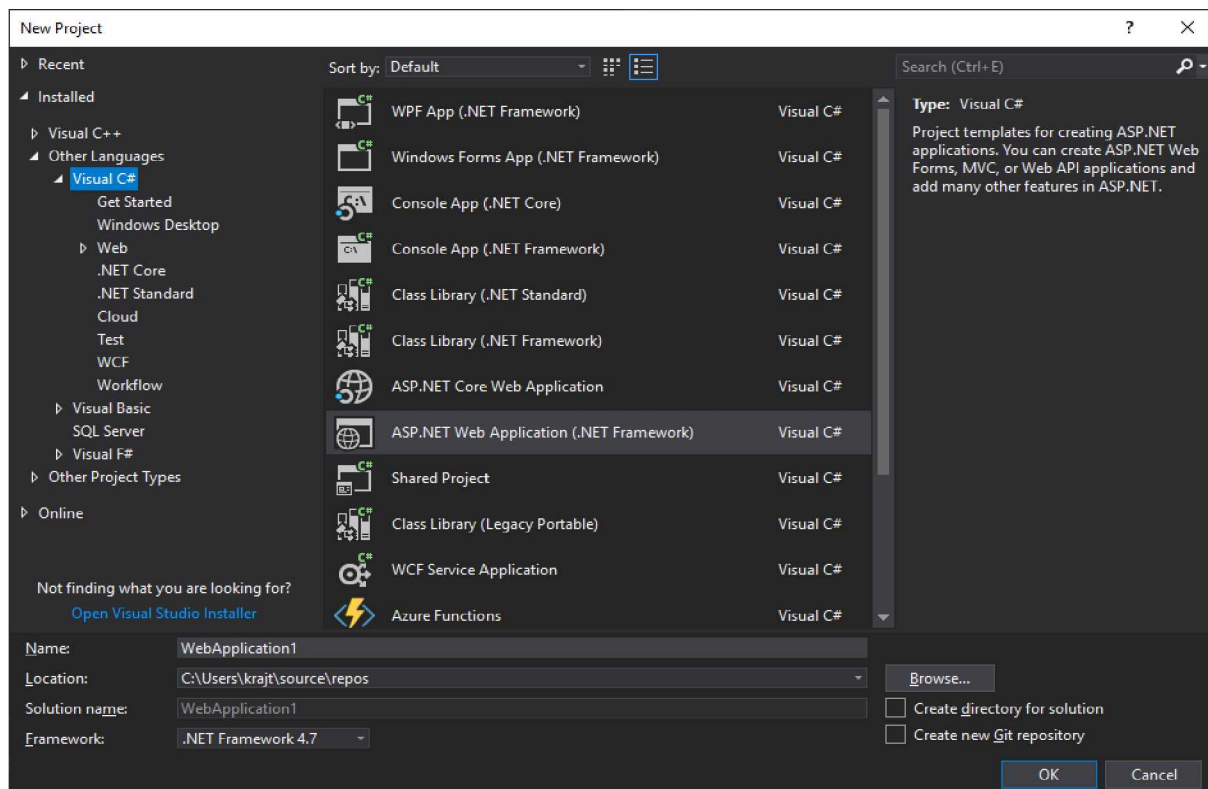
Čtvrtý a zároveň poslední příklad případů užití zobrazuje možnosti a práva uživatelů týkající se článků. Přihlášený uživatel má možnost vložit komentář pod článek, vyhledávat články dle názvu nebo kategorie a odeslat žádost o roli redaktora. Administrátor vyhodnotí žádost o roli redaktora, může roli redaktora také odebrat. Role redaktora obsahuje možnost vkládat nové články, editovat nebo odebrat stávající.



Obrázek 44 - případy užití – články

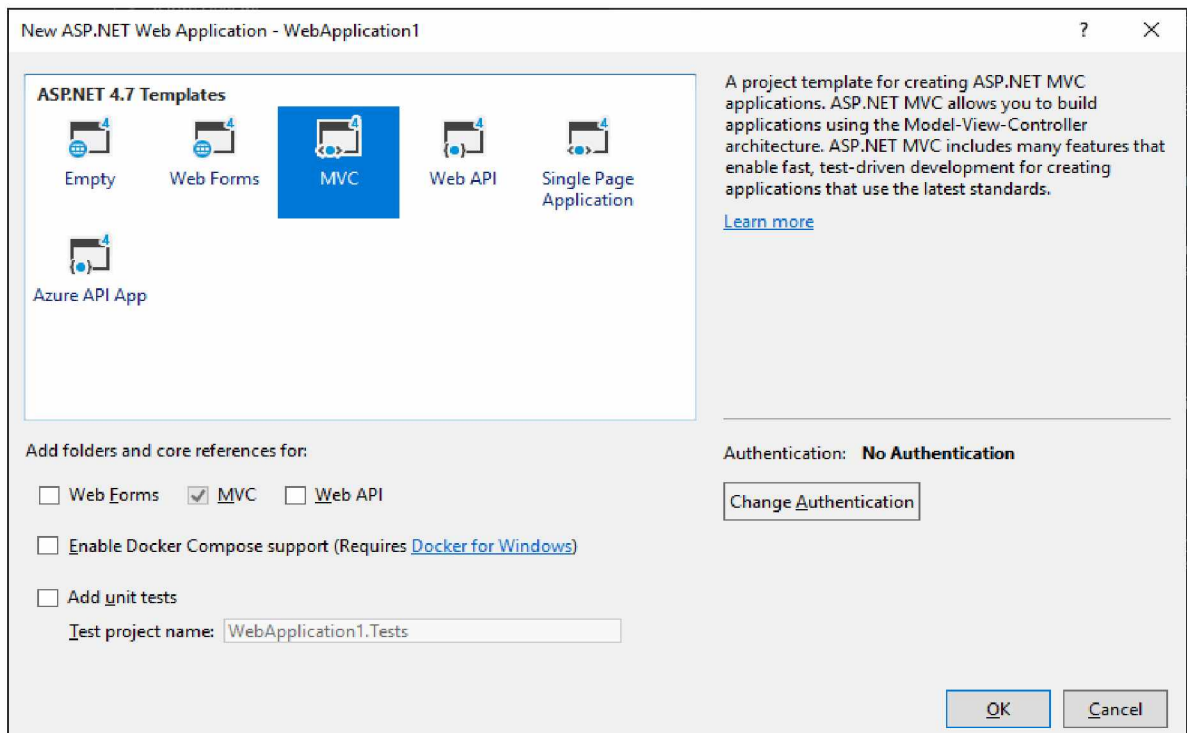
11 VYTVOŘENÍ PROJEKTU

Výsledná webová aplikace je vytvořena pomocí vývojového prostředí Visual Studio. Pomocí možnosti „Nový projekt“ vytvoříme ASP webovou aplikaci.



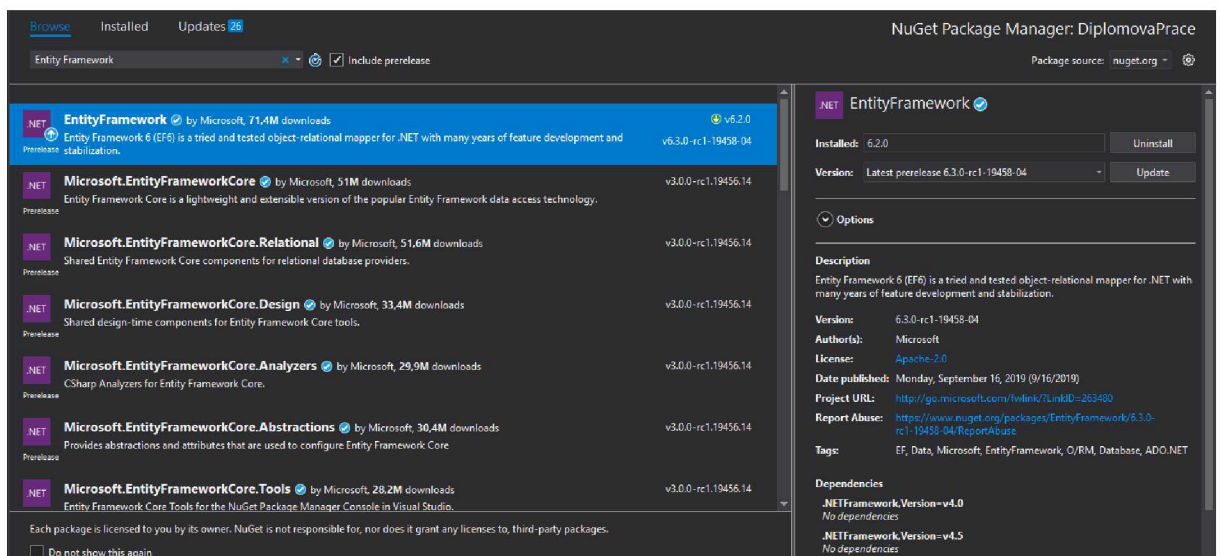
Obrázek 45 - vytvoření projektu

Po potvrzení použité technologie je uživateli zobrazena možnost výběru typu výsledné aplikace. Typy jednotlivých aplikací byly popsány v teoretické části práce. Další možností pro výběr je způsob autentizace jednotlivých uživatelů používajících výslednou webovou aplikaci. Výchozí hodnota neobsahuje autentizaci, změnit lze na možnosti individuální uživatelské účty, pracovní nebo školní účty odlišené pomocí domény nebo Windows autentizaci určenou pro intranetové webové aplikace. Poslední možností uživatele je přidání testovací jednotky pro projekt. Pro zpracování praktické části diplomové práce byl vybrána MVC typ aplikace a možnost autentizace pomocí individuálních uživatelských účtů.



Obrázek 46 - výběr typu aplikace

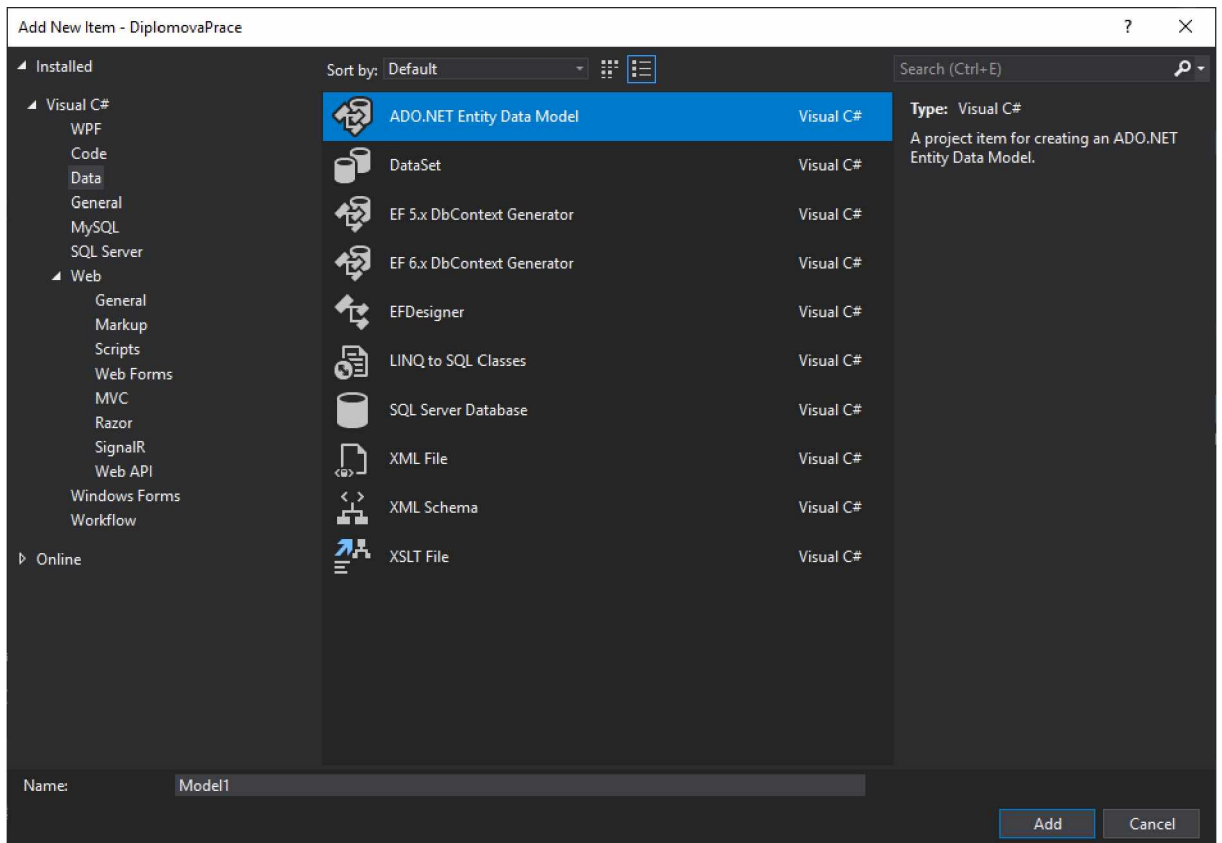
Výsledná webová aplikace používá dříve zmíněný Entity Framework který zajišťuje objektivě relační mapování mezi aplikací a databází. Nejjednodušší způsob nainstalování Entity Frameworku je pomocí NuGet správce balíčků který umožňuje jednoduchým způsobem spravovat závislosti. Balíček obsahuje zkompileovaný kód ve tvaru .dll knihovny.



Obrázek 47 - import Entity Frameworku do projektu

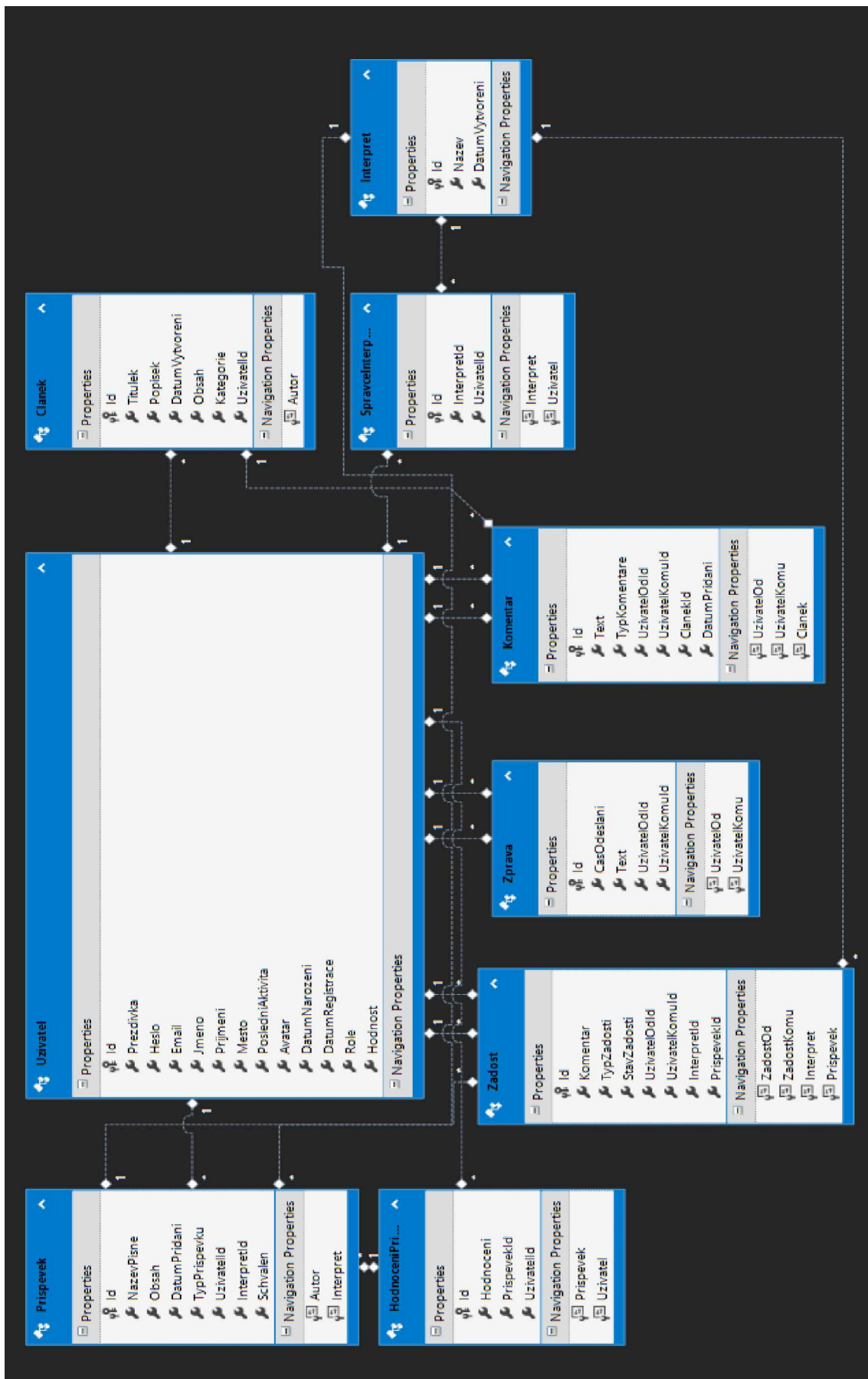
V teoretické části práce byly také zmíněny přístupy, kterými je možné začít vytvářet projekt. Základní možnosti obsahují Code First, Model First nebo Database First. Pro tento projekt byl

zvolen přístup Model First, pro vytvoření modelu je zapotřebí vložit objekt ADO.NET Entity Data Model.



Obrázek 48 - přidání modelu (.edmx)

Vytvořený model poskytuje možnost navrhnout potřebné entity (třídy, výčtové typy) a vztahy mezi entitami které budou využity ve výsledné aplikaci. Návrh modelu je velmi podobný databázovému modelu. Entita Uživatel obsahuje základní informace o přihlášeném uživateli. Každý uživatel vlastní základní atributy (ID, jméno, příjmení, email) ale i rozšířené. Mezi rozšířené atributy patří Hodnost uživatele, která je určena aktivitou uživatele. Dalším rozšířeným atributem je Role. Role slouží k odlišení práv uživatele – redaktor může vkládat nové články. Po vyhledání všech potřebných entit a atributů přichází na řadu relace mezi entitami. Například relace mezi entitami Uživatel a Článek. Uživatel má možnost vytvořit několik článků, ale jeden článek může být napsán pouze jedním uživatelem.



Obrázek 49 - výsledný model webové aplikace

Po nalezení veškerých relací je vygenerován skript, pomocí kterého bude vytvořena databáze obsahující potřebné objekty mezi které patří tabulky, cizí klíče, omezení a indexy.

Pro spuštění skriptu je nutné připojit k Visual Studiu databázi, ve které má být skript spuštěn. Po úspěšném připojení k cílové databázi uživatel vybere, které databázové objekty mají být vytvořeny. Mezi databázové objekty patří tabulky, pohledy, funkce, procedury a další. U každého typu databázové objektu uživatel zvolí které konkrétní potřebuje vytvořit (například konkrétní tabulky ze skriptu).

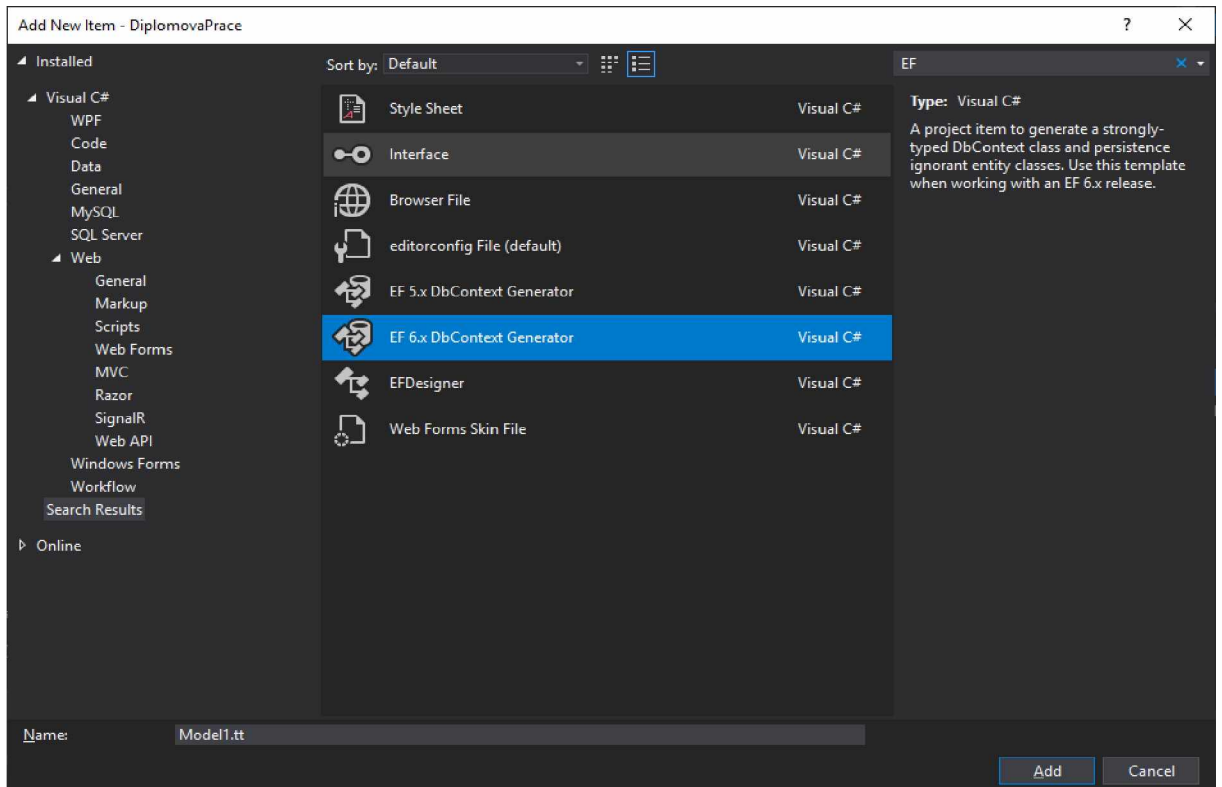
The image shows a screenshot of the 'Add Connection' dialog box in Visual Studio. The dialog is titled 'Add Connection' and contains the following fields and options:

- Data source:** A text box containing 'Microsoft SQL Server (SqlClient)' and a 'Change...' button.
- Server name:** A dropdown menu showing 'DESKTOP-2L6UR2E/SQLEXPRESS' and a 'Refresh' button.
- Log on to the server:** A section containing:
 - Authentication:** A dropdown menu showing 'Windows Authentication'.
 - User name:** A text box.
 - Password:** A text box.
 - Save my password**
- Connect to a database:** A section containing:
 - Select or enter a database name:** A dropdown menu showing 'DefaultConnection'.
 - Attach a database file:** A text box, a 'Browse...' button, and a 'Logical name:' text box.

At the bottom of the dialog, there are four buttons: 'Test Connection', 'OK', 'Cancel', and 'Advanced...'.

Obrázek 50 - připojení k databázi určené pro spuštění SQL skriptu

V této fázi máme vytvořený model a databázi, nyní vytvoříme modelové třídy. Do projektu musí být vložen EF DbContext Generator. Vytvořený objekt je C# třída která zprostředkovává připojení do databáze a nabízí práci s vytvořenými entitami a relacemi v modelu (.edmx). Jednotlivé záznamy konkrétní tabulky databáze jsou uloženy v DbSet konkrétního typu. Například pokud bude potřeba vyhledat konkrétního uživatele pak provedu vytvoření instance třídy ModelContainer dědicí ze třídy DbContext a vybráním konkrétní množiny dat. Způsob funkčnosti bude dále vysvětlen na entitě Článek.



Obrázek 51 - přidání DbContext Generatoru do projektu

```
public partial class ModelContainer : DbContext
{
    public ModelContainer()
        : base("name=ModelContainer")
    {
    }
    public virtual DbSet<Prispevek> Prispevky { get; set; }
    public virtual DbSet<HodnoceniPrispevku> HodnoceniPrispevku { get; set; }
    public virtual DbSet<Komentar> Komentare { get; set; }
    public virtual DbSet<Zprava> Zpravy { get; set; }
    public virtual DbSet<Zadost> Zadosti { get; set; }
    public virtual DbSet<Interpret> Interpreti { get; set; }
    public virtual DbSet<SpravceInterpretu> SpravciInterpretu { get; set; }
    public virtual DbSet<Clanek> Clanky { get; set; }
    public virtual DbSet<Uzivatel> Uzivatele { get; set; }
}
```

Obrázek 52 - třída ModelContainer (DbContext)

DbSet<Clanek> tedy obsahuje informace o veškerých záznamech v databázové tabulce Články. Na obrázku 45 lze vidět, jak vypadá modelová třída ihned po vygenerování.

```
public partial class Clanek
{
    public int Id { get; set; }
    public string Titulek { get; set; }
    public string Popisek { get; set; }
    public System.DateTime DatumVytvoreni { get; set; }
    public string Obsah { get; set; }
    public KategorieClanku Kategorie { get; set; }
    public int UzivatelId { get; set; }
    public virtual Uzivatel Autor { get; set; }
}
```

Obrázek 53 - vygenerována modelová třída

Třída Článek tedy obsahuje atributy:

Atribut	Funkce
ID	Plní funkci jednoznačného identifikátoru záznamu v databázové tabulce.
Titulek	Název konkrétního článku.
Popisek	Stručný popis konkrétního článku.
DatumVytvoreni	Datum, kdy byl článek vytvořen.
Obsah	Obsah konkrétního článku.
Kategorie	Zařazení článku do kategorie.
UzivatelId	ID uživatele který napsal konkrétní článek.
Autor	Instance uživatele který napsal Konkrétní článek.

Tabulka 8 - atributy modelové třídy Článek

Atributy UzivatelId a Autor slouží k odlišnému využití. Atribut Autor obsahuje veškeré údaje uživatele který napsal konkrétní článek, tedy atributy modelové třídy Uzivatel. V případě zjištění informací o konkrétním autorovi článku není zapotřebí vytvářet nový dotaz s vyhledáním uživatele dle jeho ID.

Modelová třída může být doplněna o anotace, které dále rozšiřují možnosti práce s modelovými třídami. Vysvětlení anotací na konkrétním případu:

Anotace	Význam
DisplayName	Název atributu který bude zobrazen uživateli ve View.
Required	Povinný atribut.
ErrorMessage	Text chybové zobrazené zprávy uživateli ve View.

DataType	Rozšířená definice datového typu (MultilineText = víceřádkový text)
AllowHtml	Anotace povolující atributu obsahovat HTML tagy.

Tabulka 9 - anotace modelové třídy Článek

```

public partial class Clanek
{
    [DisplayName("ID")]
    public int Id { get; set; }

    [DisplayName("Titulek")]
    [Required(ErrorMessage = "Titulek musí být vyplněný.")]
    public string Titulek { get; set; }

    [DisplayName("Popisek")]
    [DataType(DataType.MultilineText)]
    [Required(ErrorMessage = "Popisek musí být vyplněný.")]
    [AllowHtml]
    public string Popisek { get; set; }

    [DisplayName("Datum vytvoření")]
    public System.DateTime DatumVytvoreni { get; set; }

    [DisplayName("Obsah")]
    [AllowHtml]
    [Required(ErrorMessage = "Obsah musí být vyplněný.")]
    public string Obsah { get; set; }

    [DisplayName("Kategorie")]
    [Required(ErrorMessage = "Kategorie musí být vyplněný.")]
    public KategorieClanku Kategorie { get; set; }

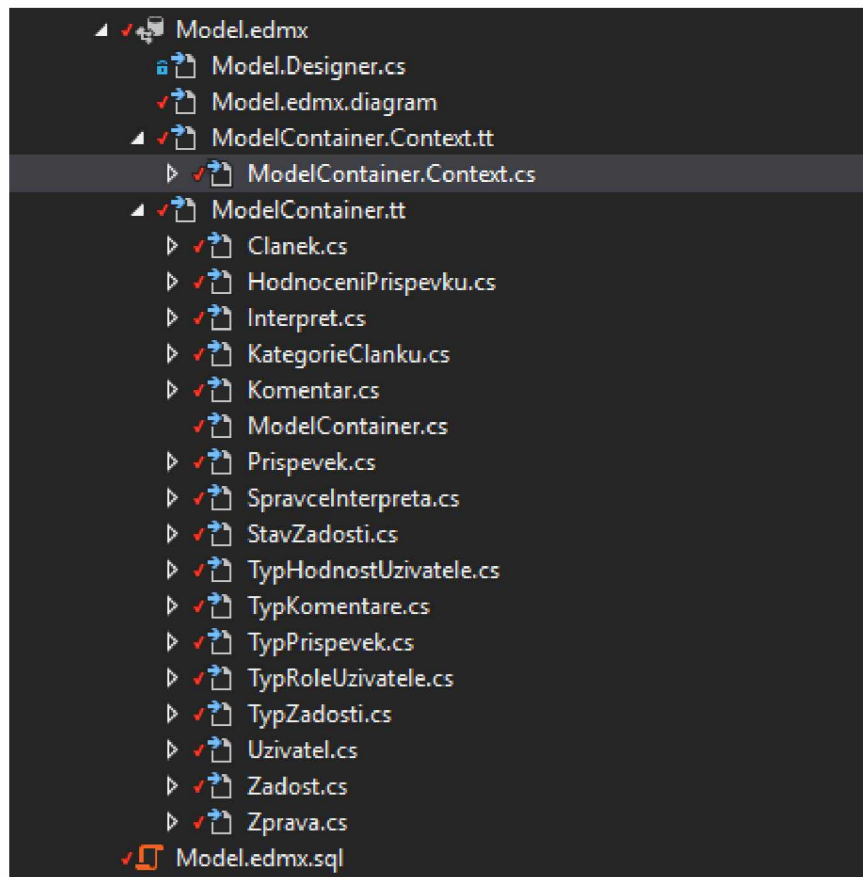
    [DisplayName("Autor")]
    public int UzivatelId { get; set; }

    [DisplayName("Autor")]
    public virtual Uzivatel Autor { get; set; }
}

```

Obrázek 54 - modelová třída Článek doplněna o anotace

Mezi nejdůležitější ze zmíněných anotací patří AllowHTML. Atribut Obsah bude ukládat nebo zobrazovat informace textového editoru a bude tedy obsahovat formátovací znaky (,
, <i>). Bez anotace AllowHtml nebude požadavek o vytvoření článku nikdy odeslán z důvodného podezření na XSS útok (Cross-site scripting). XSS je metoda narušení webové stránky pomocí využití bezpečnostních chyb ve skriptech, které pracují s neošetřenými vstupy. Následky útoku mohou být jak narušení vzhledu webové stránky ale i obcházení zabezpečení a získávání osobních údajů uživatelů. Bez použití anotace AllowHTML nelze tedy odeslat text obsahující HTML tagy bez zrušení kontroly požadavků odeslaných uživatelem. Zrušení kontroly je možné provést nad konkrétní metodou controlleru nebo controllerem samotným, čímž ale dochází k bezpečnostní chybě.



Obrázek 55 – struktura ve Visual Studiu (modelové třídy, skript, model)

12 DATABÁZOVÝ MODEL

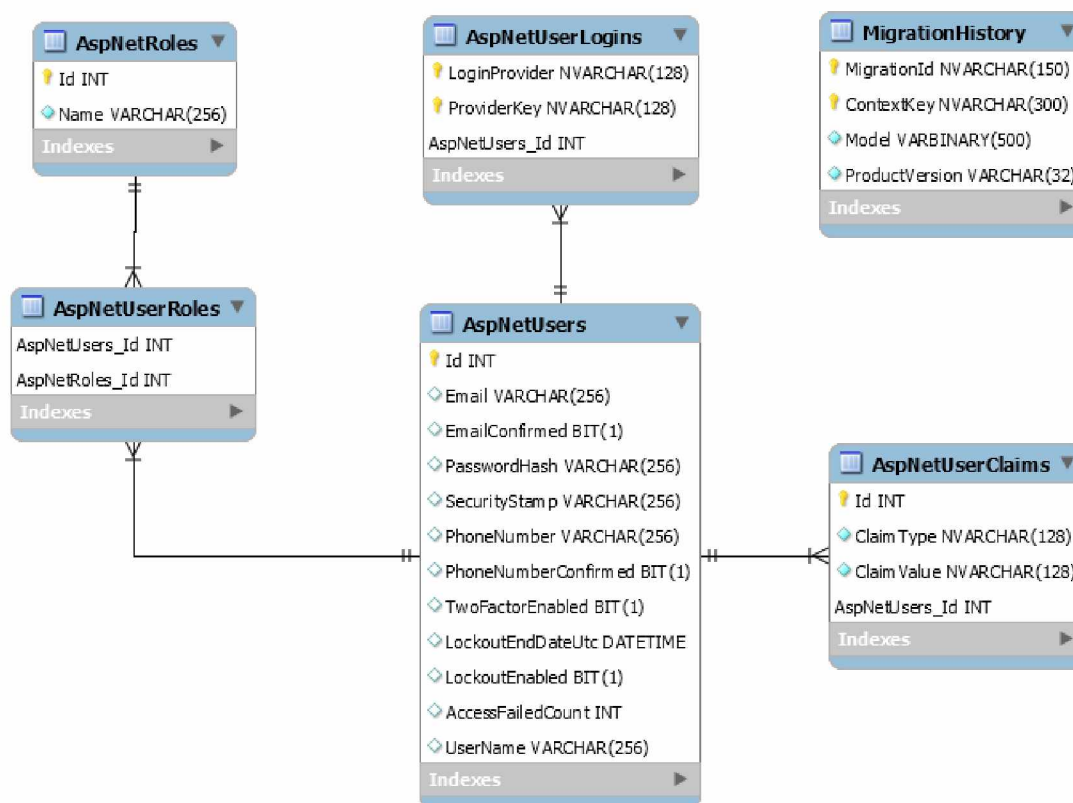
Databázový model obsahuje základní entity, jejich atributy a vztahy mezi entitami (relace). Dle požadavků zadavatele je vytvořen konceptuální model popisující potřebná data pro uchování na abstraktní úrovni. Mezi nejznámější konceptuální model patří entitně-vztahový model (ER). Mezi základní stavební jednotky ER modelu patří entita, relace a atribut. Entita je fyzický existující objekt nebo událost, označena podstatným jménem. Relace zachycuje vztah mezi entitami. Pro jednodušší pochopení mohou být označeny slovesy popisující konkrétní vztah. Pro upřesnění vztahu mezi entitami existují pojmy kardinalita a parcialita. Kardinalita vyjadřuje, kolik řádků jedné entity může být ve vztahu s řádky entity druhé. Parcialita vyjadřuje povinnost účasti ve vztahu. Atribut je konkrétní typ informace uložený o entitě. Každá entita musí mít minimálně tolik atributů, aby byla možná jednoznačná identifikace konkrétní instance (řádek) entity.

12.1 ER model

Entitně-relační model zobrazuje základní modelované objekty, informace, které uchovávají a vztahy mezi nimi. Na obrázku č. 42 lze vidět model vytvořený pro Entity Framework, podle kterého je vygenerována výsledná databáze. Při použití objektově relačního mapování atributy třídy přesně odpovídají atributům entity databáze. Pro propojení webové aplikace a databázové vrstvy je využit databázový kontext, který je složen z přiděleného prostoru (tabulek). Praktická část obsahuje databázové kontexty dva pro odlišení práce se správou uživatelů a uživatelských dat.

12.2 Entity Framework Identity

První zmíněný kontext je Entity Framework Identity. Při vytváření projektu vývojář zvolí, zda bude zapotřebí využít ve výsledném projektu autentifikační algoritmy. V případě využití uživatelských účtů bude společně s vytvořeným modelem zavedeny databázové tabulky uchovávající informace o uživateli (heslo, aktivace účtu) a přidělených rolích. Pomocí Identity lze na každé webové stránce (View) jednoduchým způsobem rozpoznat roli přihlášeného uživatele a přizpůsobit výsledný vzhled webové stránky. Tabulka MigrationHistory obsahuje informace o jednotlivých migracích. Migrace slouží pro upravení databáze do stejné podoby při úpravě modelové třídy. Migrace mohou být spuštěny ručně ale také probíhat automaticky.

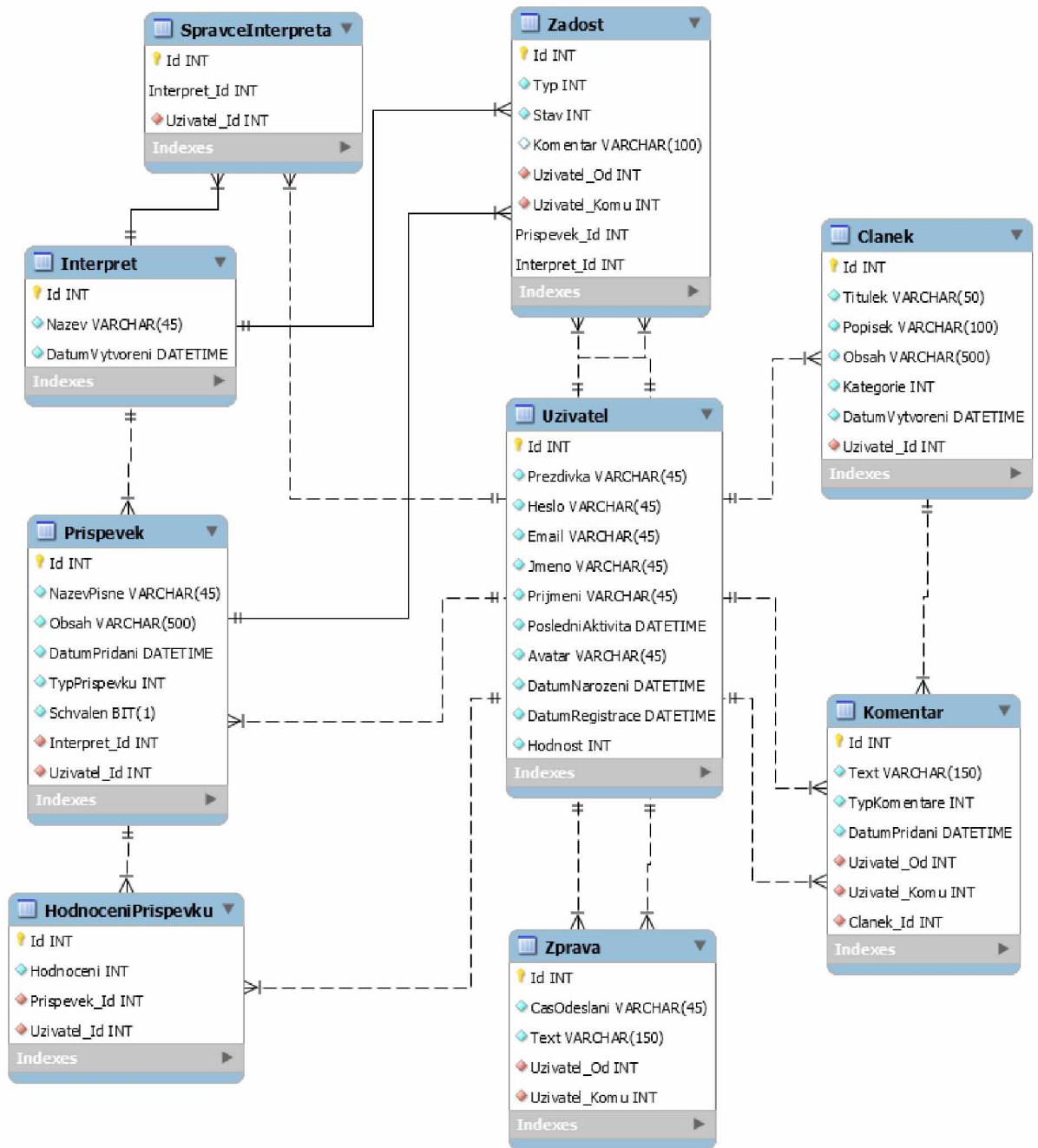


Obrázek 56 - Entity Framework kontext

12.3 Uživatelská část

Uživatelská část obsahuje již namodelované tabulky z .edmx souboru. Základní třídou modelu je třída *Uživatel*, který je přihlášen a různými způsoby může používat webovou aplikaci. Přihlášený uživatel disponuje možností komunikovat s ostatními uživateli. Třída *Zpráva* uchovává informaci o odesílateli, příjemci, obsahu zprávy a času odeslání. Protože je použito objektově relační mapování, atributy třídy jsou přímo převedeny na sloupce databázové tabulky. Jedna z dalších možností uživatele je hodnocení příspěvku vytvořeného jiným uživatelem, které probíhá pomocí vybrání počtu hvězd. Každý uživatel má možnost ohodnotit konkrétní příspěvek pouze jednou. Při opakovaném hodnocení nastane pouze upravení posledního hodnocení. Přihlášený uživatel může napsat příspěvek vlastní. Pro vytvoření příspěvku je zapotřebí vyplnit název písně, typ příspěvku (akordy, tabulatura) a typ příspěvku. Po odeslání příspěvku je vytvořena žádost o schválení příspěvku správcem interpreta. Třída *SprávceInterpreta* plní v databázovém modelu funkci vazební tabulky, která spojuje informace o třídě *Interpret* a *Uživatel*. Správci interpreta je umožněno vyhodnotit přichodící žádosti o vytvoření příspěvku. Existuje situace, kdy interpret nemá přiřazeného správce. V této situaci je

počítání s přidělením správcovství administrátorem webové aplikace. Správcem interpreta je prvotně uživatel, který daného interpreta vytvořil. Uživatel disponuje také možností přidat vlastní článek za podmínky obdržení role redaktora. O roli redaktora je možné požádat odesláním žádosti administrátorovi webové aplikace. Třída Článek je složena z názvu, krátkého popisu obsahu článku (titulek), obsahu samotného a výběrem kategorie. Veškerý obsah webové aplikace je řízen systémem žádostí automaticky odeslaných ostatním uživatelům. Třída Žádost tedy obsahuje informace o typu (žádost o přidání příspěvku, o roli redaktora), aktuálním stavu, komentáři (možné využít pro poznámku uživatele odesílajícího žádost ale také zpětná vazba uživateli od hodnotitele. Stručný příklad využití systému žádostí je přidání nového příspěvku interpreta. Při kliknutí na uložit příspěvek nastane vygenerování žádosti odeslané správci interpreta (případně správcům při větším počtu správců interpreta). Přímo v žádosti je možné zjistit interpreta, název písně, odesílatele žádosti (veškeré hodnoty lze otevřít a zobrazit detail) a komentář který je nepovinný pro odeslání žádosti. Správce interpreta prohlédne příspěvek, vyhodnotí, zda splňuje pravidla pro přidání příspěvku a vyhodnotí žádost. Po vyhodnocení je odesílateli žádosti odesláno upozornění o vyřízení žádosti. Třída Uživatel uchovává informace o přezdívkce, heslu, emailu uvedeném při registraci (potvrzení emailu probíhá pomocí kliknutí na zasláný odkaz, vyhodnocení potvrzení emailu provádí Identity kontext), datum a čas poslední aktivity, informace o profilovém obrázku, datum narození, datum registrace a hodnost. Hodnost je uživateli přidělena podle aktivity (počet přidávaných příspěvků, článků). Hodnost slouží jako motivace uživatelům vkládat nový obsah.



Obrázek 57 - kontext uživatelské části

13 NAsAZENÍ WEBOVÉ APLIKACE

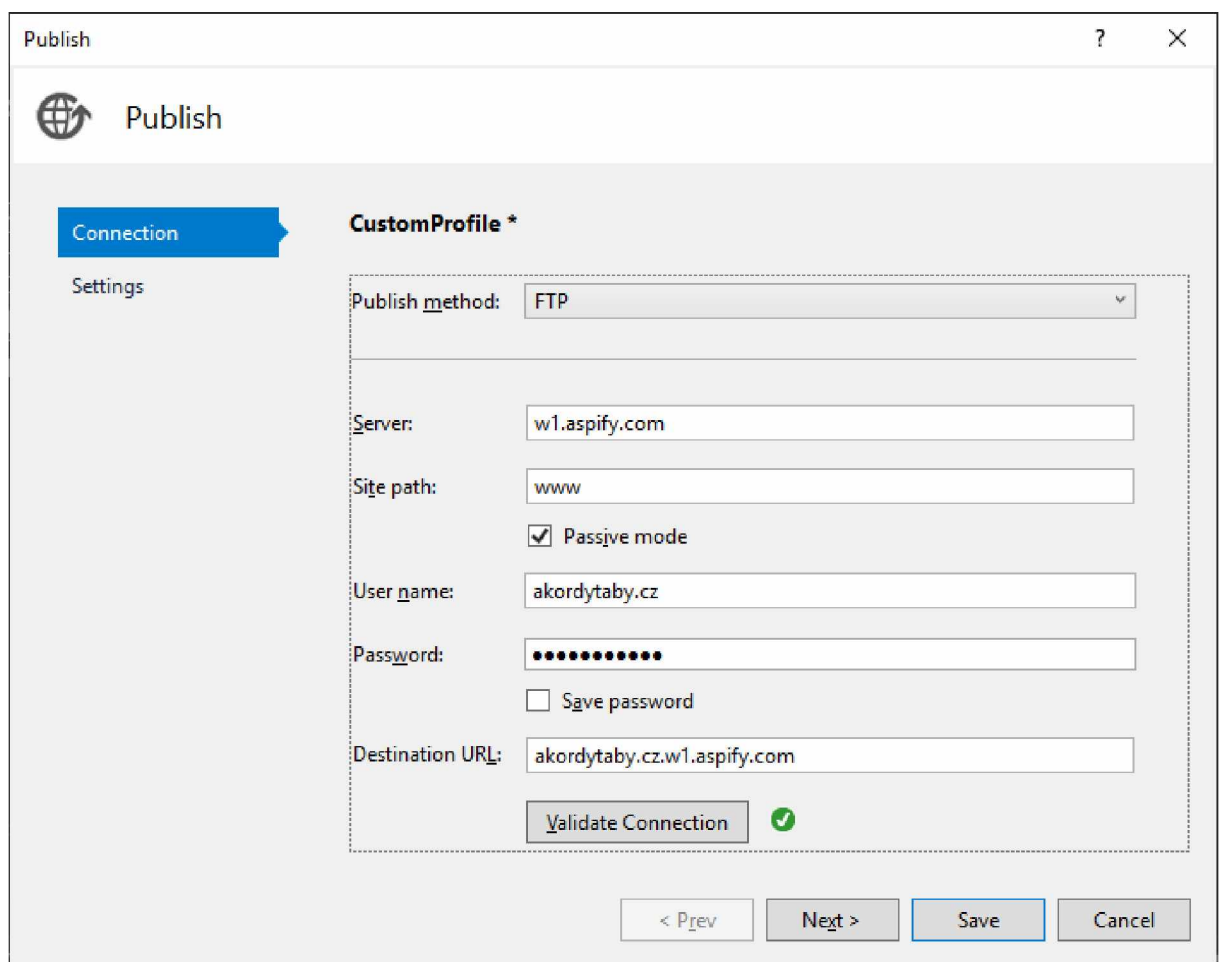
Nasazení neboli publikování webové aplikace je možné provést přímo z prostředí Visual Studia. Při vybrání možnosti „Publikovat“ jsou nabídnuty uživateli základní možnosti publikování webové aplikace.

13.1 Publikování projektu ASP pomocí programu Visual Studio

Následující kapitola stručně popisuje základní způsoby publikace webové aplikace.

13.1.1 FTP

FTP neboli File Transfer Protocol je první možnost publikování webové aplikace používá protokol pro přenos souborů mezi zařízeními. Umožňuje automatické propojení s FTP serverem, který slouží pro uložení souborů a složek webové aplikace.



The image shows a screenshot of the 'Publish' dialog box in Visual Studio. The window title is 'Publish'. On the left, there is a 'Connection' tab selected, with a 'Settings' sub-tab. The main area is titled 'CustomProfile *'. It contains the following fields and options:

- Publish method:** A dropdown menu set to 'FTP'.
- Server:** A text box containing 'w1.aspify.com'.
- Site path:** A text box containing 'www'.
- Passive mode**
- User name:** A text box containing 'akordytaby.cz'.
- Password:** A text box with masked characters (dots).
- Save password**
- Destination URL:** A text box containing 'akordytaby.cz.w1.aspify.com'.
- Validate Connection:** A button with a green checkmark icon, indicating a successful connection.

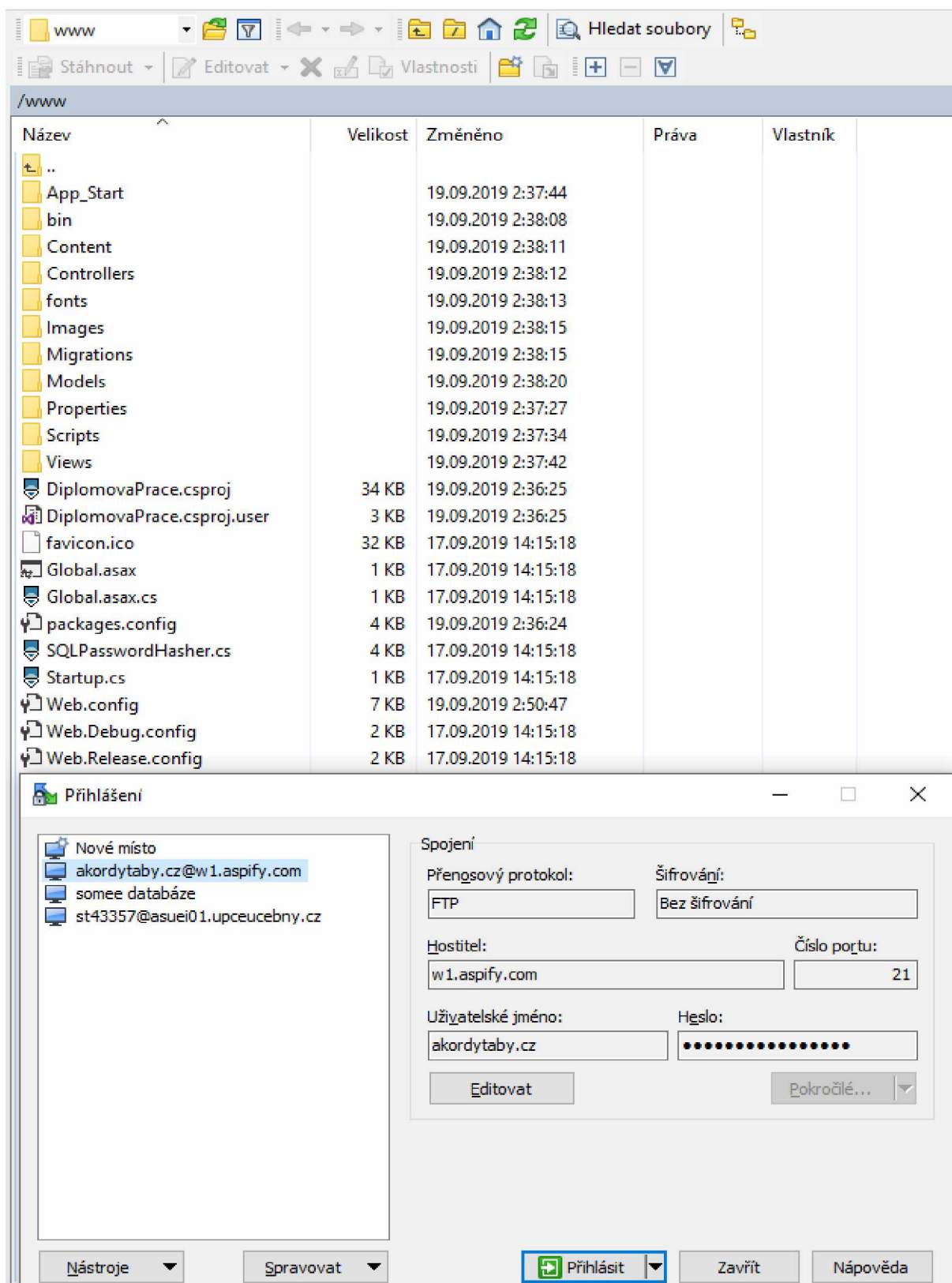
At the bottom of the dialog, there are four buttons: '< Prev', 'Next >', 'Save', and 'Cancel'.

Obrázek 58 - publikace pomocí FTP serveru

13.1.2 Lokální adresář

Druhá možnost publikace webové aplikace je do určené lokální složky. Visual Studio vygeneruje potřebné soubory dle hierarchické struktury projektu. Vygenerovaná data lze poté

pomocí FTP klienta přenést na FTP server. Mezi nejznámější FTP klienty patří WinSCP nebo Total Commander.



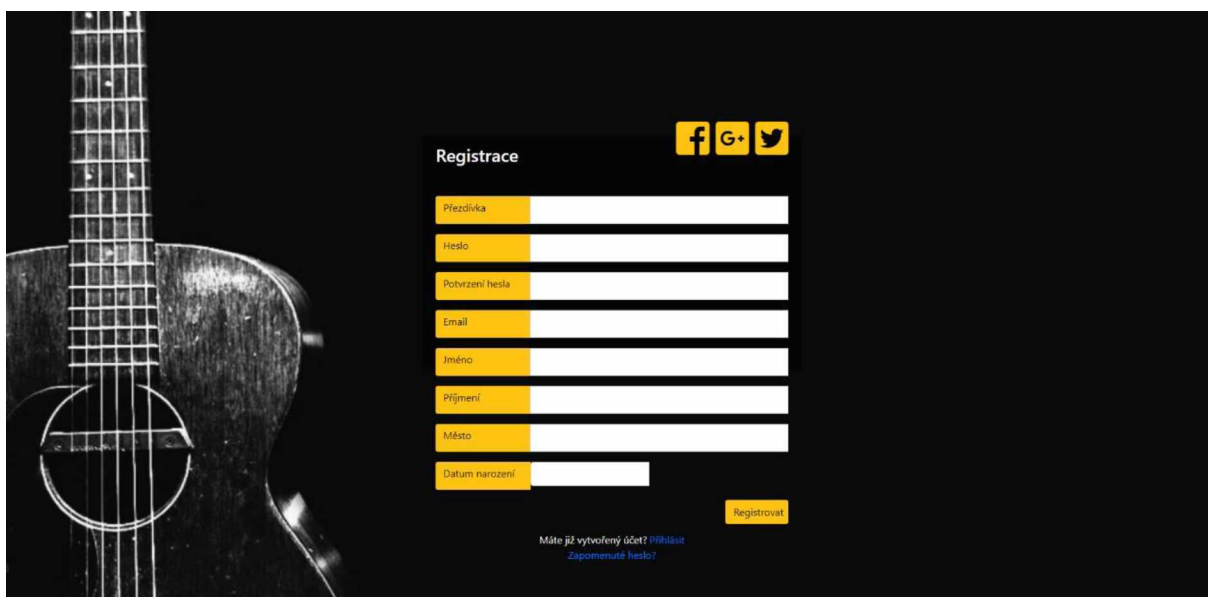
Obrázek 59 - přenos pomocí FTP klienta WinSCP

14 UŽIVATELSKÁ PŘÍRUČKA

Uživatelská příručka popisuje jednotlivé části webové aplikace, které funkce plní nebo jak probíhá zpracování jednotlivých požadavků uživatele.

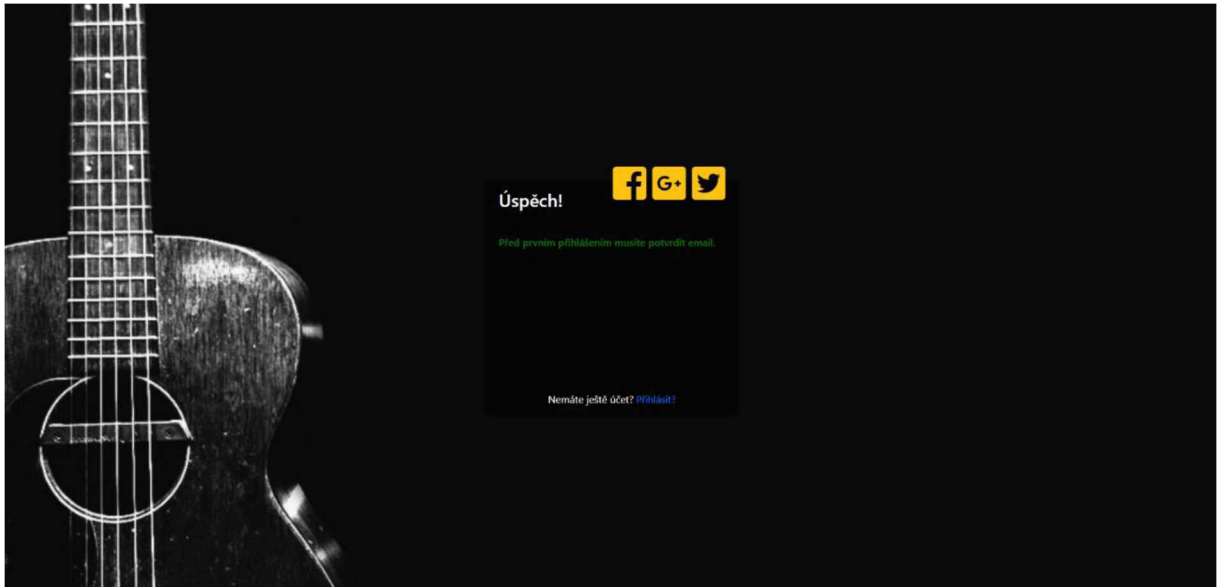
14.1 Registrace

Pro používání webové aplikace je povinná registrace uživatele z důvodu omezení činností bez potřebné autentizace a autorizace. Ve chvíli, kdy uživatel přistoupí na již zapamatovaný URL odkaz vnitřnější částí webové aplikace bez přihlášení, bude automaticky přeměřován na formulář pro přihlášení. Před odesláním formuláře je uživatel informován o bezpečnosti zadaného hesla, dále je kontrolováno zda jsou zadaná hesla shodná a přezdívka uživatele dodržuje podmínku jedinečnosti. Zadané heslo je z bezpečnostních důvodů šifrováno.



Obrázek 60 - registrace uživatele

Po odeslání formuláře je uživatel informován o stavu registrace. První možností je zvýraznění nevalidních hodnot, které potřebují dodatečně upravit. První možnost nastane, pokud uživatel se zadanou přezdívkou v databázi již existuje a není dodržena podmínka jedinečnosti, zadaná hesla nejsou shodná nebo dostatečně silná. Druhá možnost je informování uživatele o úspěšném dokončení registrace.



Obrázek 61 - oznámení úspěšné registrace

Po úspěšné registraci je uživateli odeslán email obsahující vygenerovaný odkaz pro potvrzení emailové adresy a aktivace zaregistrovaného účtu. Bez potvrzení emailové adresy není uživateli umožněno přihlášení a tudíž i používání webové aplikace.

Potvrďte Váš účet.



AkordyTaby <krajta.5@seznam.cz>

13:18

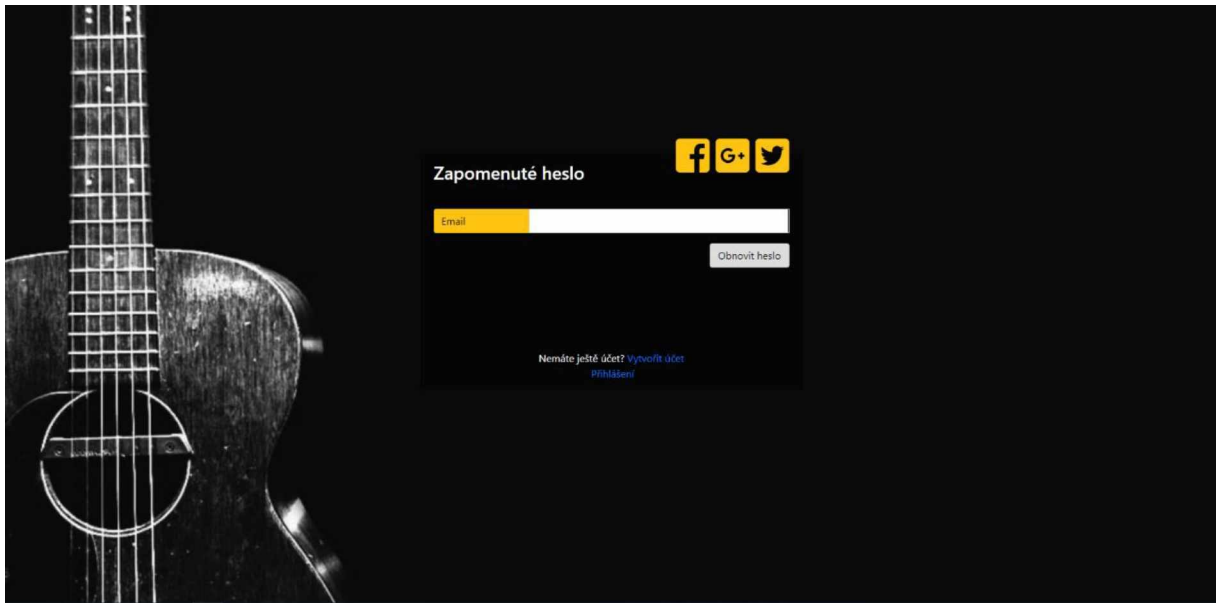
Komu: vanicekdaavid@outlook.cz

Prosím potvrďte email kliknutím [zde](#)

Obrázek 62 - potvrzení emailové adresy

14.2 Zapomenuté heslo

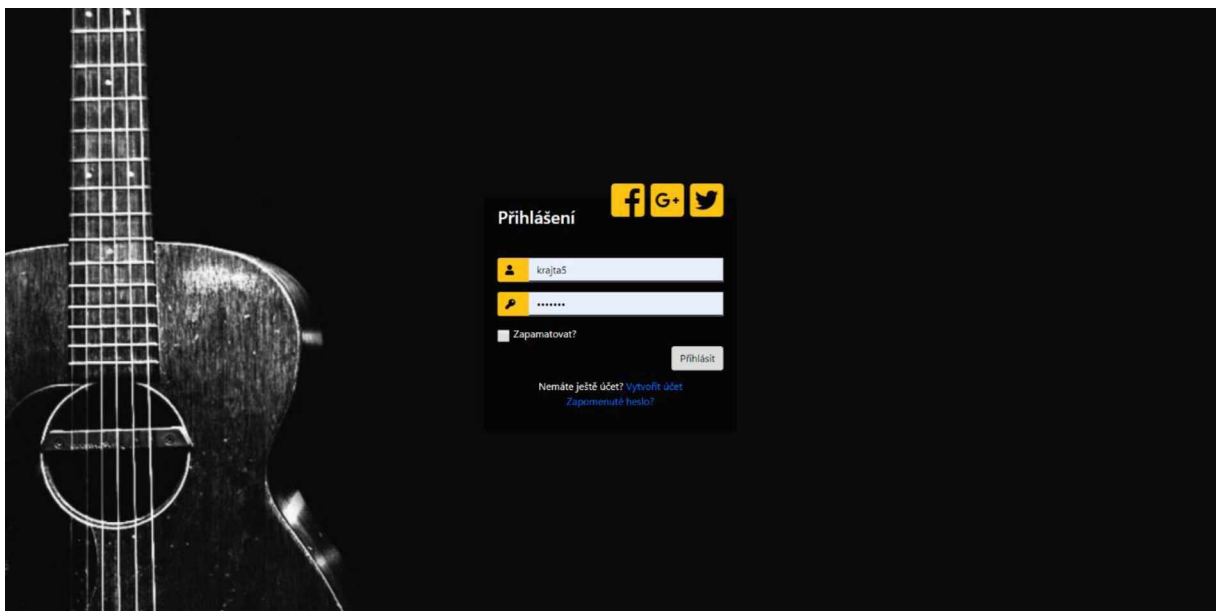
Uživatel, který zapomene vlastní heslo potřebuje možnost vyresetování hesla a poté nastavení hesla nového. Z tohoto důvodu je požadováno potvrzení registračního emailu před prvním přihlášením. Možnost obnovit heslo lze použít pod formulářem pro přihlášení nebo registraci kliknutím na „Zapomenuté heslo?“. Po vyplnění emailu uvedeného při registraci obdrží uživatel email obsahující odkaz na formulář pro nastavení nového hesla.



Obrázek 63 - obnovení hesla

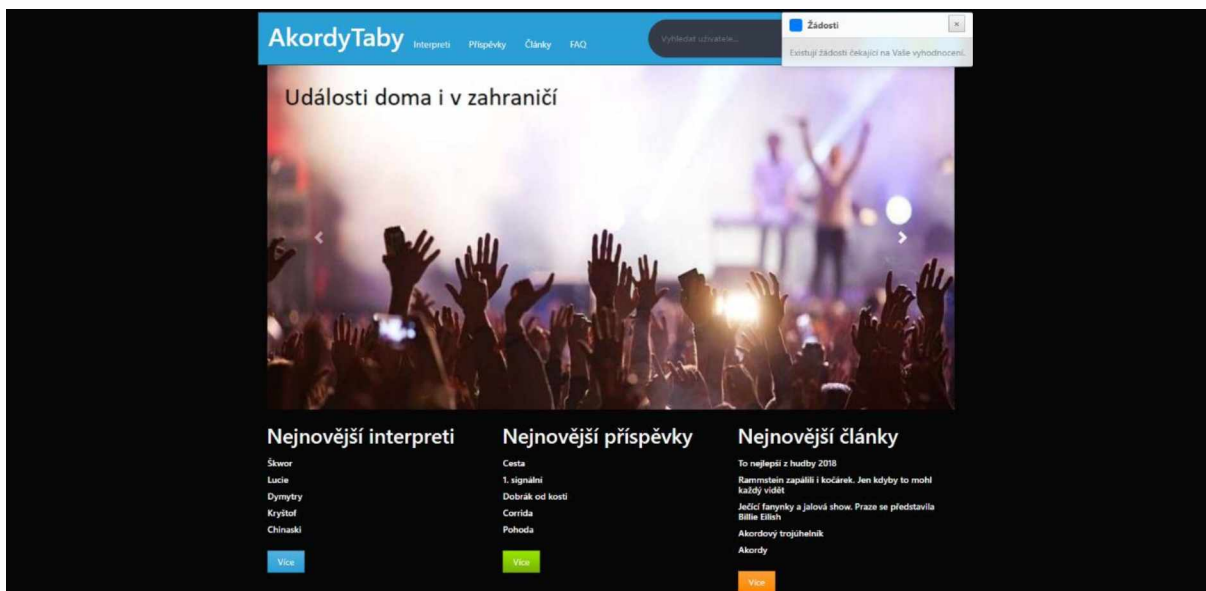
14.3 Přihlášení

Přihlášení probíhá pomocí zadání hodnot do formuláře. Mezi zadané hodnoty patří přezdívka uživatele a heslo vázané k zaregistrovanému účtu. Pokud nastane situace neshodnosti dvojice hodnot, uživatel bude informován pomocí chybového hlášení.



Obrázek 64 - přihlášení uživatele

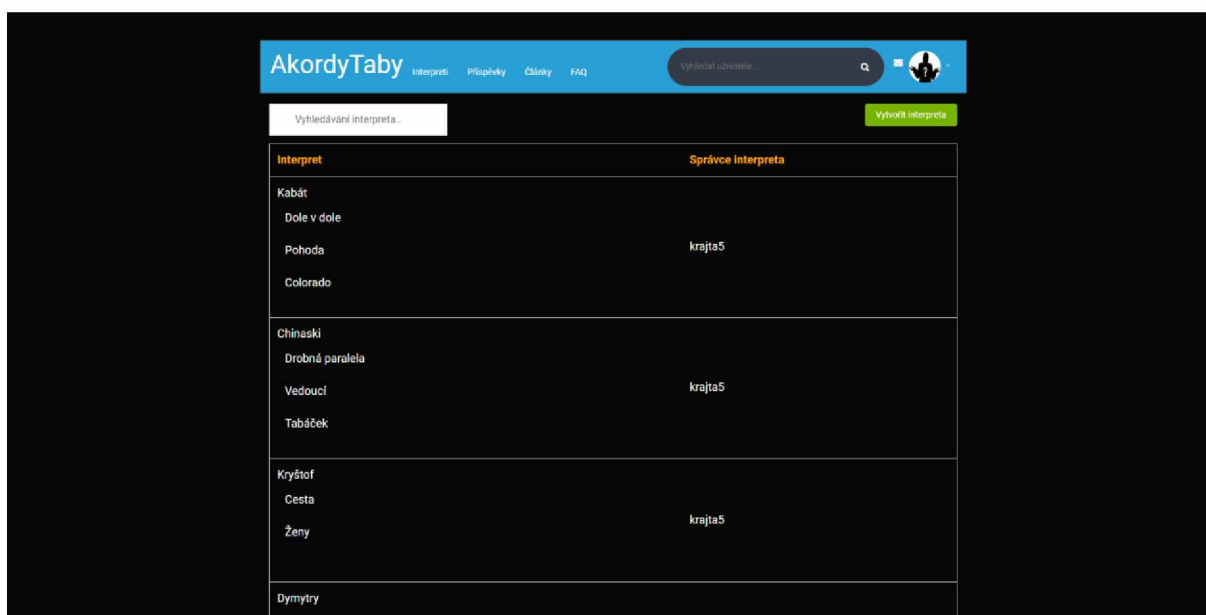
Po úspěšném přihlášení je uživatel přesměrován na domovskou stránku. Domovská stránka slouží jako rozcestník mezi jednotlivými částmi webové aplikace a zároveň upozorňuje přihlášeného uživatele na nevyřízené žádosti.



Obrázek 65 - domovská stránka

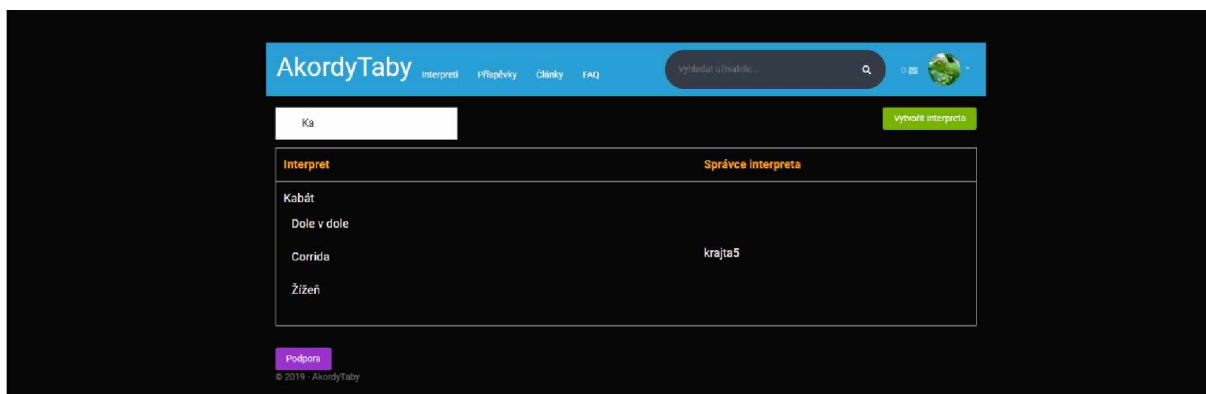
14.4 Interpreti

Interpreti patří mezi první část webové aplikace. Ve výchozím stavu dochází k zobrazení určitého počtu interpretů. Počet není velký kvůli zachování přehlednosti výpisu interpretů ale také neobsahuje pouze pár interpretů pro případnou inspiraci hudebníka začít hrát písně nové skupiny která zaujala například názvem skupiny nebo poznáním názvu písně spojené s interpretem. U každého interpreta jsou vypsány tři náhodné písně které daný interpret obsahuje, které je možné otevřít přímo ze zobrazené tabulky. Společně s písněmi je rovněž vypsán správce daného interpreta.



Obrázek 66 - výchozí výpis interpretů

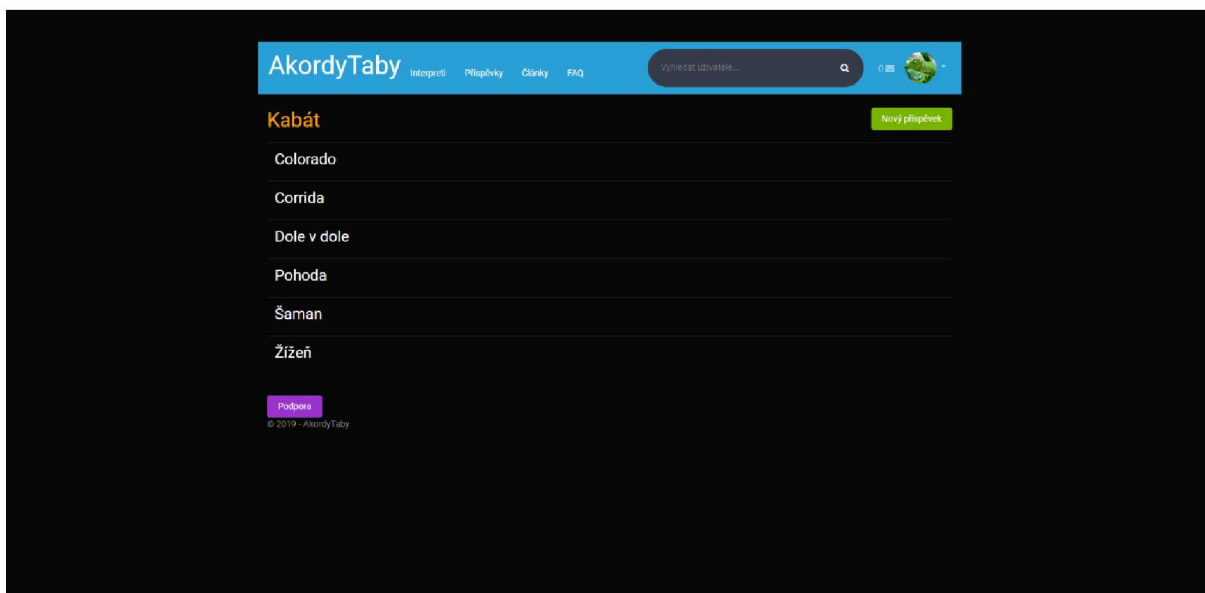
V případě využití vyhledávání interpreta uživatel zadá celý název nebo část názvu interpreta. Při vyhledávání interpreta nenastane pouze našeptání správné hodnoty ale k dynamické změně webové stránky která bude obsahovat interprety obsahující ve svém názvu vyhledávanou část. Pro dodržení přehlednosti je vždy vypsán omezený počet interpretů obsahující shodu s vyhledávanou částí. Pokud hledaný interpret je obsažen v databázi ale nenachází se ve výpisu, postačí přidat další znak z názvu interpreta, který opět zúží počet vyhledaných výsledků.



Obrázek 67 – příklad vyhledání interpreta

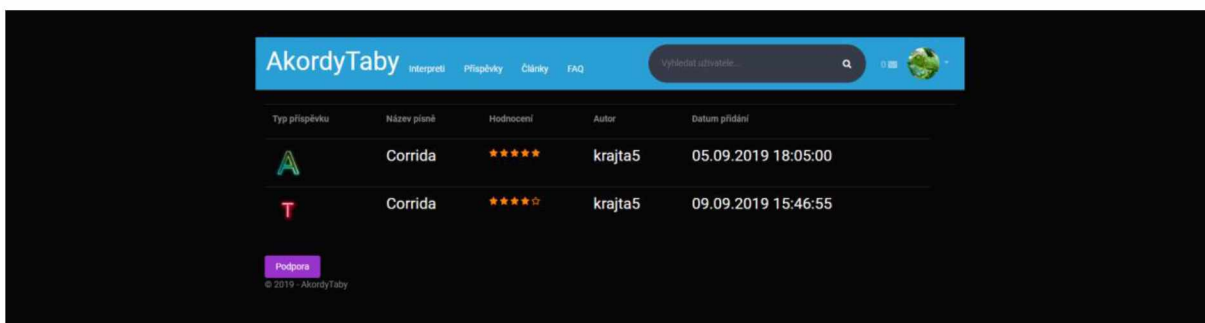
V této části webové aplikace je možné také vytvořit nového interpreta. Pro vytvoření nového interpreta postačí pouze zadat název interpreta. Po vyplnění a potvrzení zadané hodnoty dojde k ověření, zda zadaný interpret již není v databázi. Uživatel je informován o vytvoření interpreta nebo o jeho existenci. V případě vytvoření interpreta je uživatel zároveň i jeho správcem. Role správce interpreta umožňuje uživateli schvalovat příspěvky, které budou přidány pod konkrétního interpreta. Proces schvalování příspěvků bude popsán v kapitola o administraci profilu.

V případě nalezení a otevření požadovaného interpreta uživatel uvidí názvy příspěvků které konkrétní interpret obsahuje. Pro zajištění přehlednosti příspěvků jsou vypsány pouze názvy písní, tím nastane zamezení duplicit a nepřehledně dlouhého výpisu veškerých příspěvků. Dále zde má uživatel možnost přidat vlastní příspěvek pomocí tlačítka v pravé části obrazovky pod navigační lištou.



Obrázek 68 – názvy příspěvků interpreta

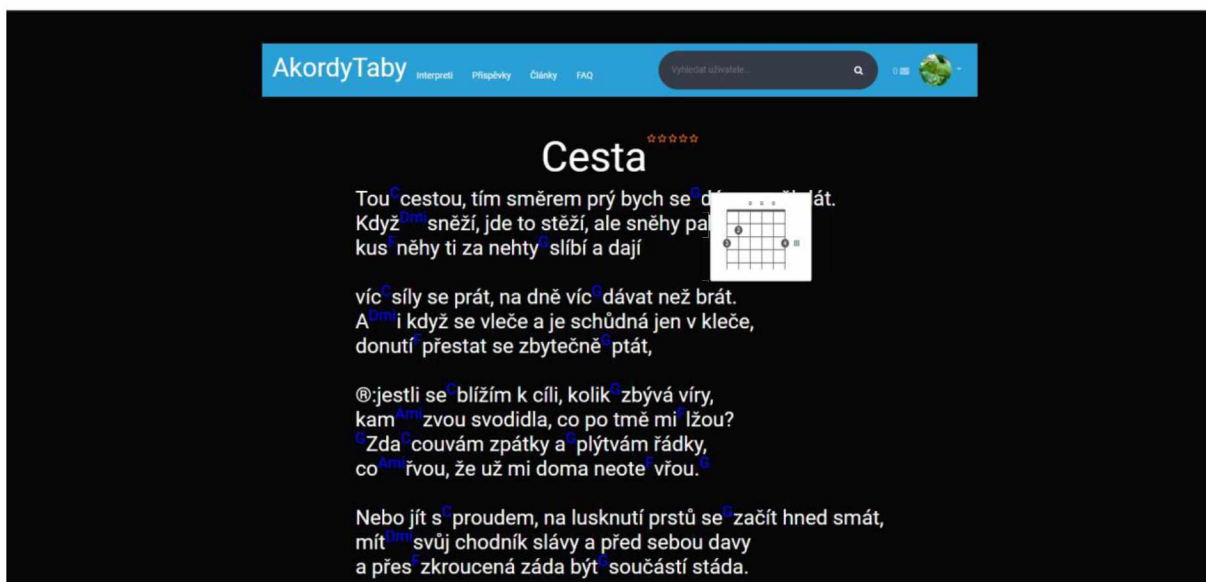
Po otevření názvu písně jsou zobrazeny příspěvky pro vybranou píseň. U příspěvků je odlišen typ příspěvku (akordy a tabulatura), dále je zobrazeno hodnocení příspěvku pomocí počtu vyplněných hvězd, autor příspěvku, datum a čas přidání příspěvku, dále lze otevřít profil autora příspěvku.



Obrázek 69 - příspěvky interpreta s vybraným názvem písně

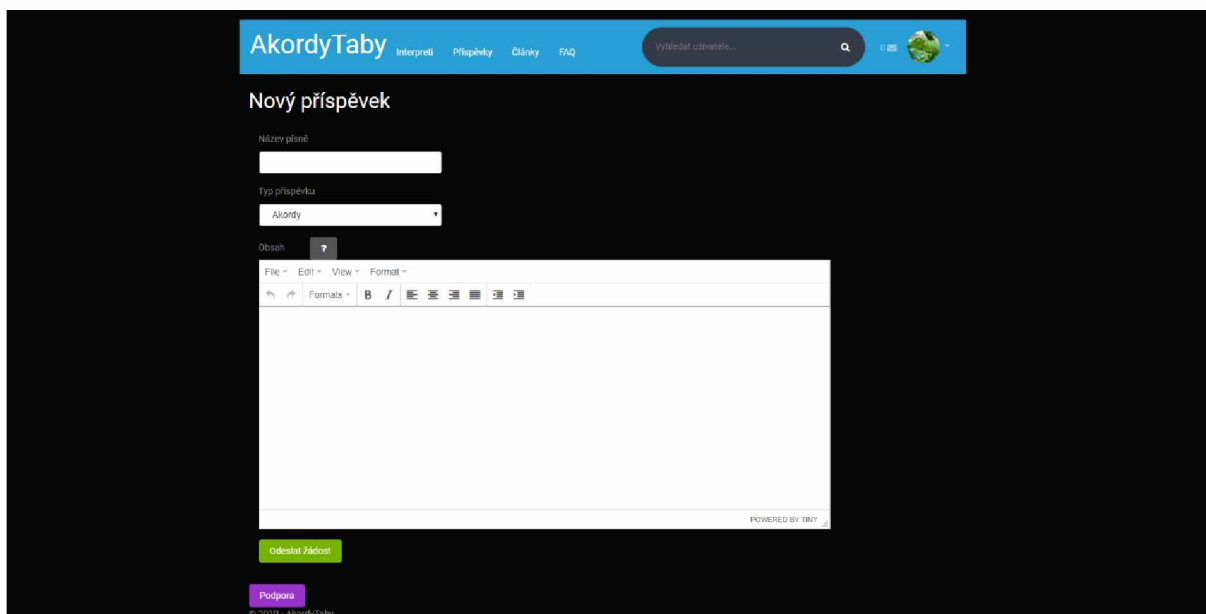
Po vybrání názvu písně a konkrétního příspěvku nastane zobrazení obsahu příspěvku. V horní centrální části je zobrazen název písně, z pravé strany názvu písně je zobrazeno pět hvězd pomocí kterých uživatel disponuje možností ohodnotit zobrazený příspěvek. Každý uživatel může ohodnotit příspěvek pouze jednou, z tohoto důvodu jsou možné dva scénáře. První scénář ve kterém uživatel konkrétní příspěvek dosud neohodnotil, nastane uložení hodnocení. Druhý scénář již počítá s předchozím hodnocením uživatele a po kliknutí na počet hvězd nastane editace hodnoty v databázi (přepsání novou hodnotou hodnocení).

Jedna z dalších možností je přímé zobrazení tvaru akordu v příspěvku. Pokud uživatel nastaví kurzor myši nad konkrétní akord, tvar akordu bude zobrazen pomocí částečného zobrazení.



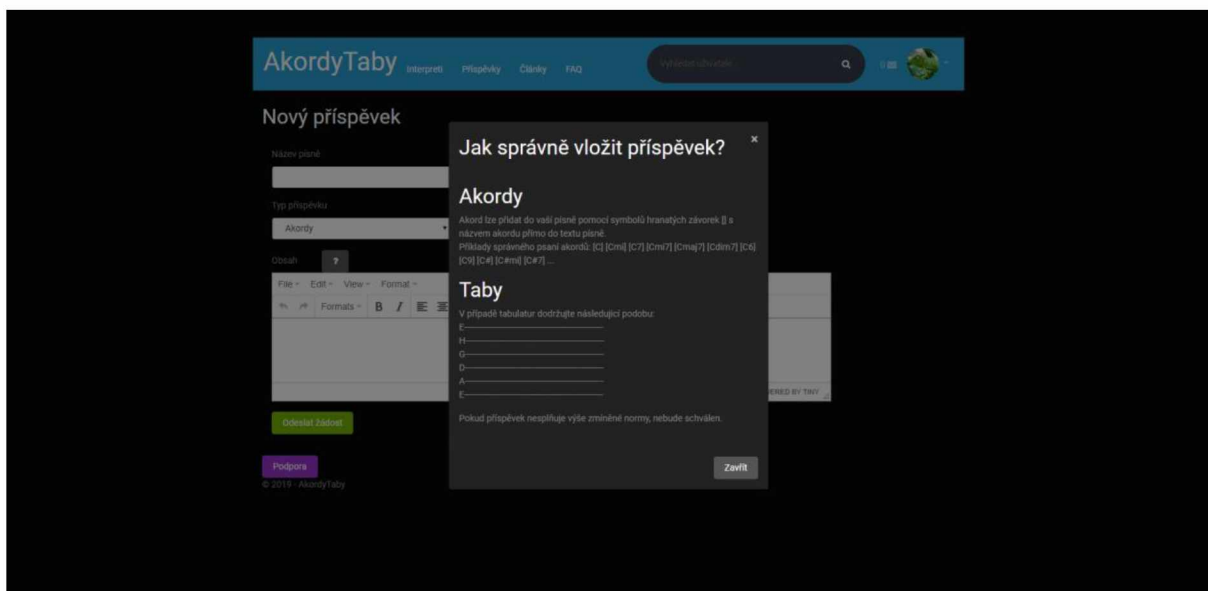
Obrázek 70 - obsah příspěvku

Jak bylo zmíněno dříve, uživatel disponuje možností přidat vybranému interpretovi příspěvek. Přidání je provedeno pomocí formuláře obsahující pole název písně, typ příspěvku a samotný obsah. Rozpoznání akordu v textu dochází pomocí zapsání názvu akordu do hranatých závorek.



Obrázek 71 - vložení příspěvku

Vkládání příspěvků samozřejmě potřebuje pravidla, která budou stručná a jednoznačná. Pro zobrazení pravidel vložení příspěvku slouží ikona otazníku. Po kliknutí uživatele na ikonu nastane zobrazení modálního okna s pravidly přidání příspěvku. Pravidla obsahují správný zápis tvaru akordů do hranatých závorek pro typ příspěvku akordy a pořadí strun pro typ příspěvku tabulatura.

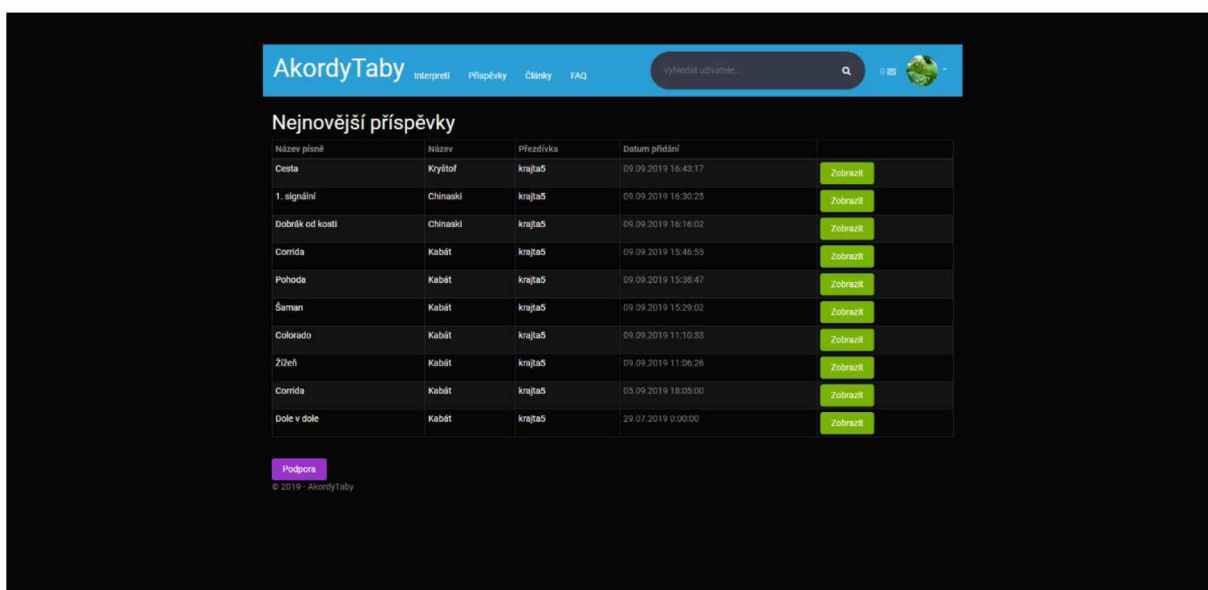


Obrázek 72 - pravidla pro přidání příspěvku

Po dopsání příspěvku a jeho odeslání je vytvořena žádost, která je odeslána uživateli s rolí správce interpreta pro konkrétního interpreta.

14.5 Příspěvky

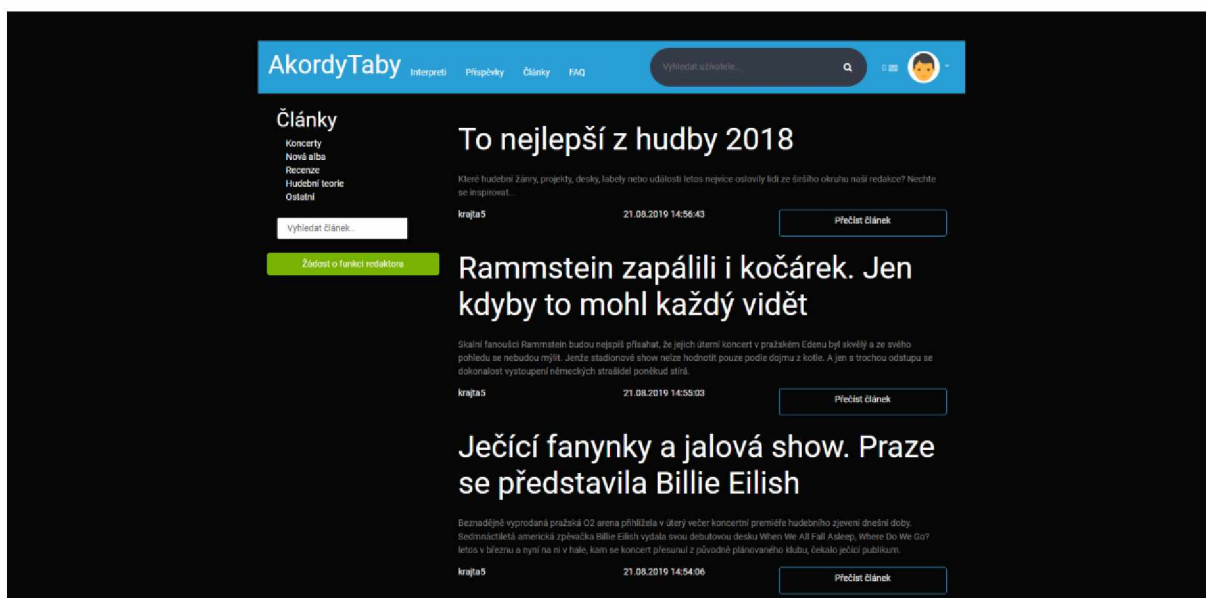
Druhá položka navigační lišty jsou příspěvky. Příspěvky obsahují určitý počet nejnovějších schválených příspěvků. U každého záznamu lze přímo zobrazit vybraný příspěvek, interpreta nebo profil uživatele, který daný příspěvek vytvořil. Dále je zde zobrazený i datum a čas přidání příspěvku.



Obrázek 73 - nejnovější příspěvky

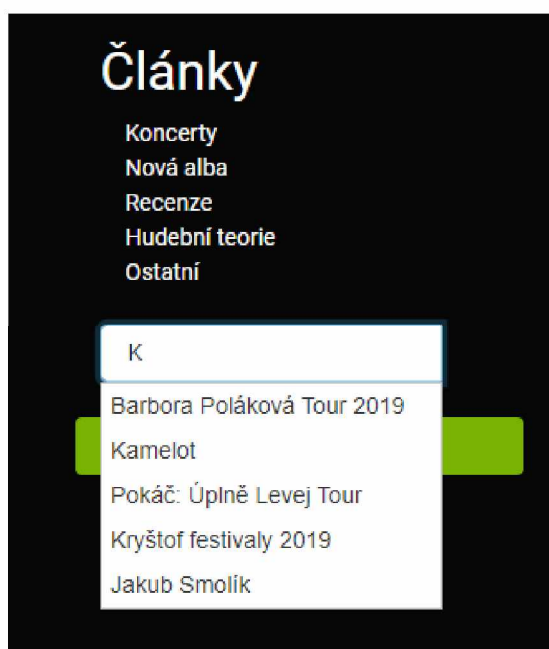
14.6 Články

Třetí položkou navigační lišty jsou články. Výchozí řazení článku je sestupně dle času přidání. Uživatel má možnost zažádat o roli redaktora (v případě že již roli redaktora obdržel, nastane nahrazení tlačítka žádosti o roli redaktora tlačítkem nový článek). Články jsou rozděleny do jednotlivých kategorií (novinky, nová alba, recenze, hudební teorie a ostatní).

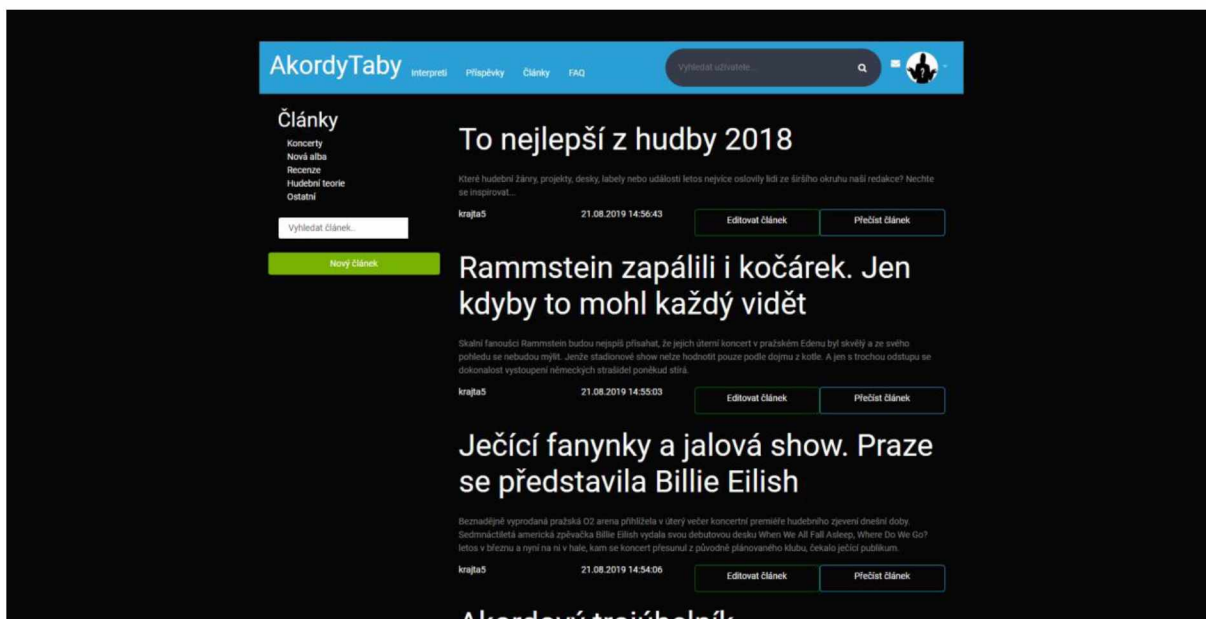


Obrázek 74 - výpis článků

Články je také možné vyhledávat dle názvu. Našeptávač zobrazí určitý počet článků obsahující v názvu článku vyhledávaný řetězec.

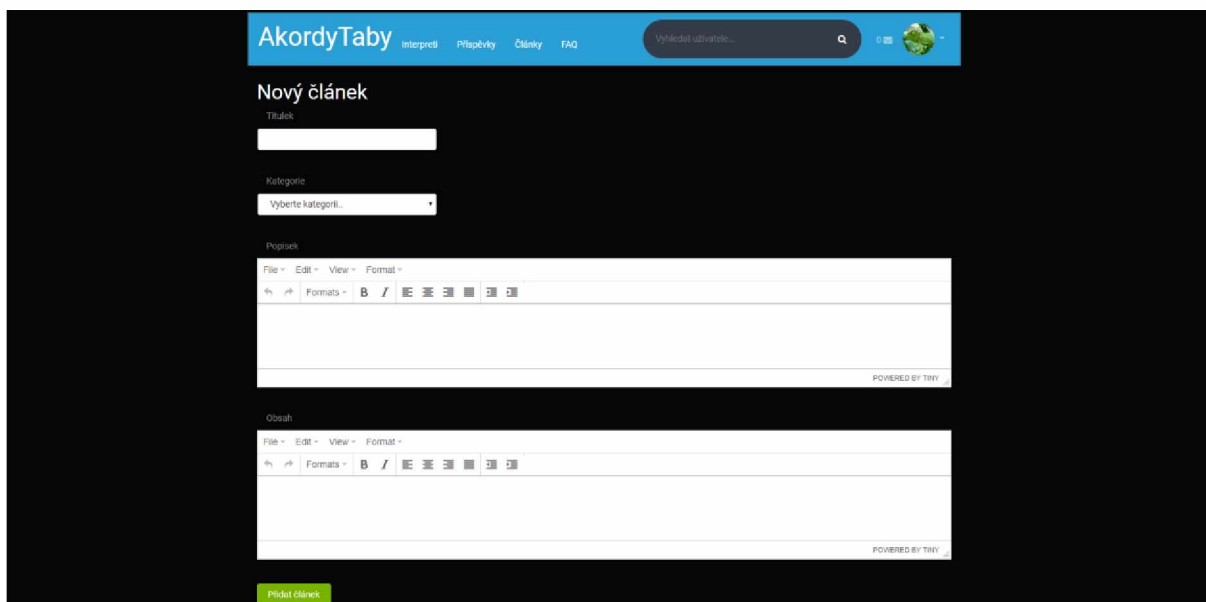


Pokud uživatel získal roli redaktora a již má vytvořené vlastní články, bude uživateli zobrazena možnost přesměrování na editaci vlastního článku přímo z výpisu článků.



Obrázek 75 - články z pohledu redaktora

Vytváření nového článku probíhá pomocí vyplnění formuláře. Titulek a popis článku je zobrazen ve výpisu článků, kategorie slouží pro potřeby filtrování článků dle kategorie.

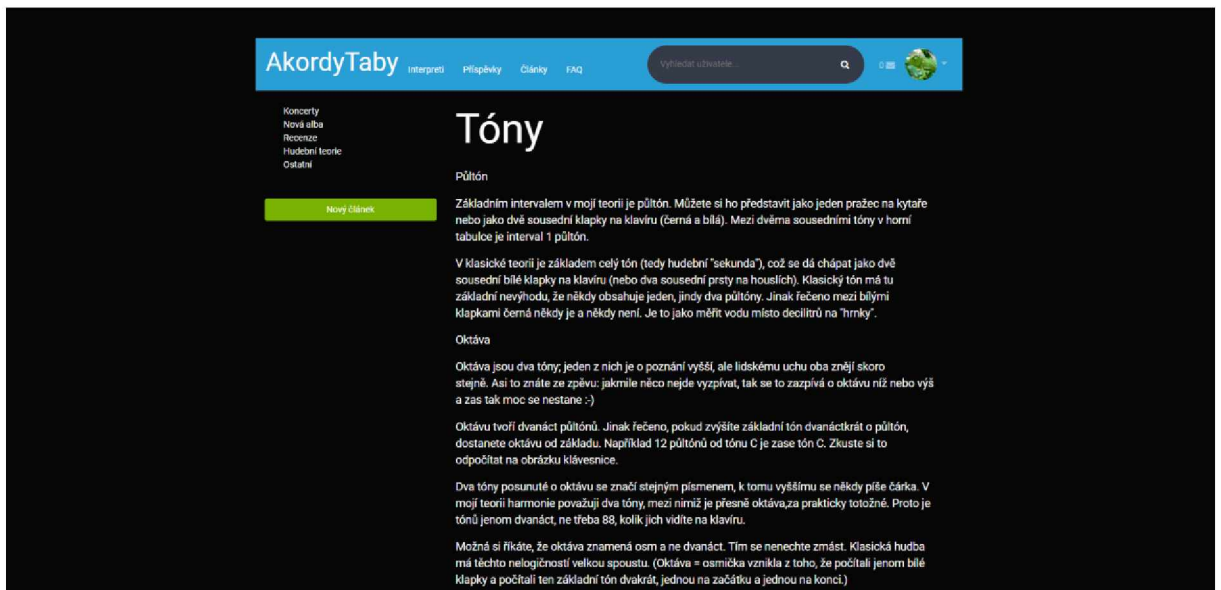


Obrázek 76 - vytvoření nového článku

Položka obsah a popis jsou vyplňovány pomocí textového editoru. Textový editor umožňuje formátování textu které je při uložení textu zobrazeno jako HTML značky. HTML není ve výchozím stavu povoleno ukládat (řešení a z jakého důvodu není povoleno je rozepsáno u

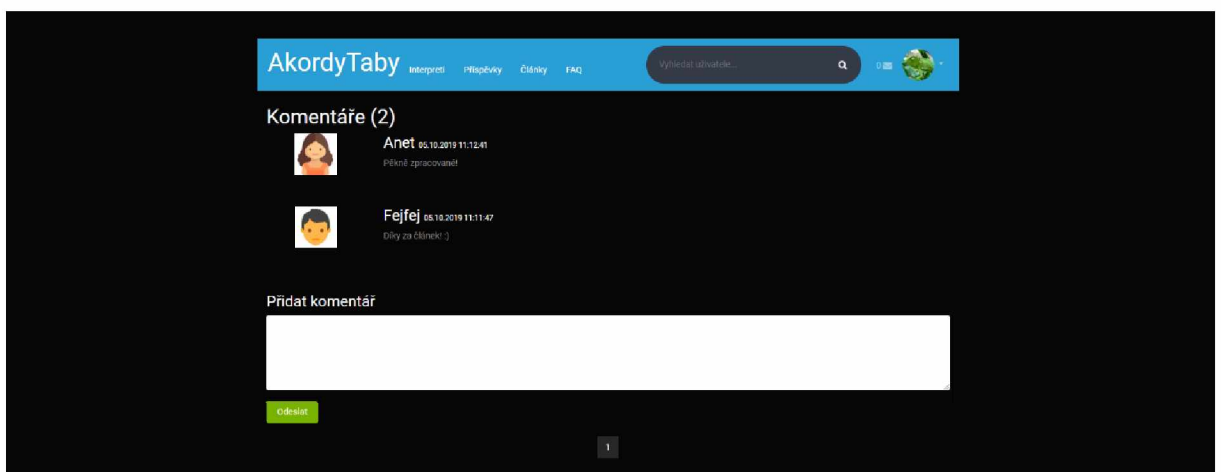
předchozí kapitoly s XSS útokem). Pro dvojí bezpečnost zadaných vstupních hodnot jsou HTML tagy před uložením zakódovány a opětovně dekodovány při zobrazení hodnot.

Zobrazit obsah článku je možné kliknutím na název článku nebo na tlačítko s obsahem přečíst článek.



Obrázek 77 - zobrazení konkrétního článku

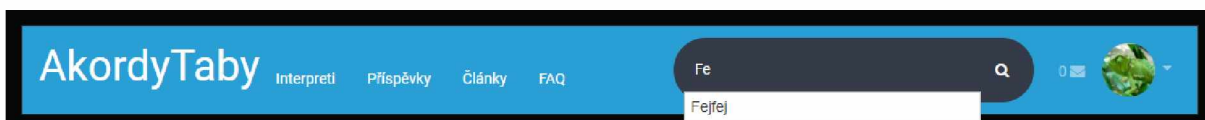
V dolní části obrazovky pod obsahem článku je možnost zobrazení komentářů konkrétního článku a zároveň přidání komentáře vlastního. Uživatelé nemají možnost přímo zareagovat na komentáře ostatních z důvodu prevence spamu. Pokud uživatel má zájem zareagovat, může využít implementovaný systém zpráv.



Obrázek 78 - diskuze u článku

14.7 Vyhledání uživatele

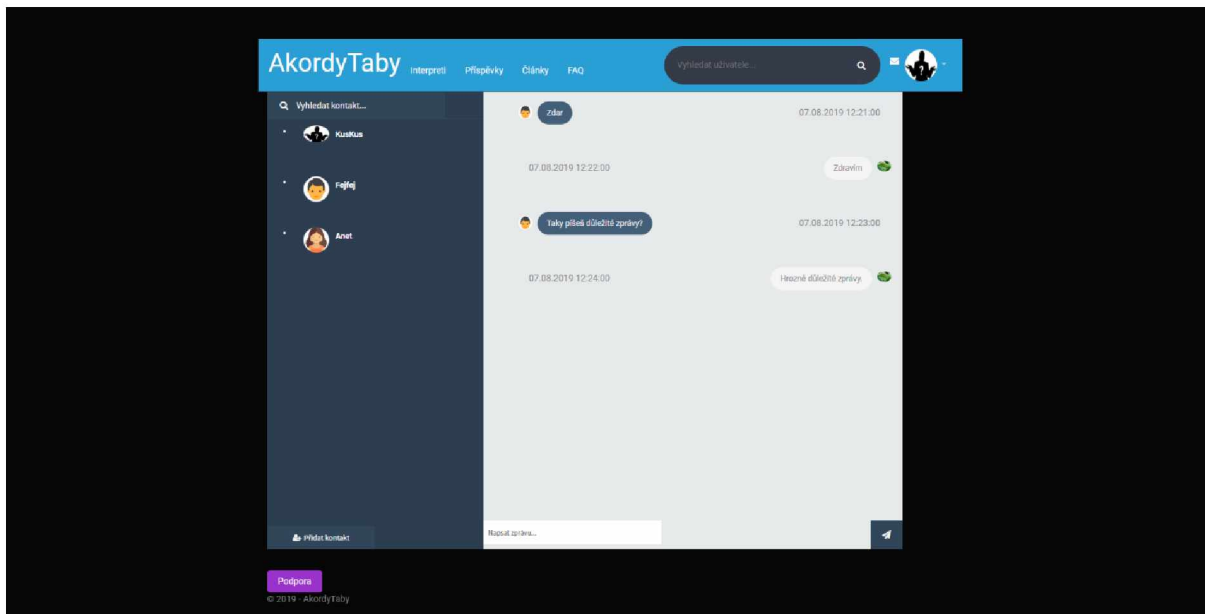
Vyhledání uživatele je umožněno přímo v navigační liště pomocí zadání celé nebo alespoň části přezdívky. Po vybrání kliknutím na přezdívku nastane přesměrování na profil vyhledaného uživatele.



Obrázek 79 - vyhledání uživatele

14.8 Implementovaný systém zpráv

Pomocí kliknutí na ikonu obálky v navigační liště nastane přesměrování na zprávy, které uživatel obdržel nebo napsal. Pořadí kontaktů je řazeno dle časů poslední zprávy s daným uživatelem. Odlišení přijatých a odeslaných zpráv je odlišeno pomocí rozřazení do různých stran chatu (přijaté zprávy na levé straně, odeslané zprávy na straně pravé). Přidat kontakt je umožněno pomocí tlačítka v levém dolním rohu. Po kliknutí na tlačítko bude zobrazeno modální dialog určené pro vyplnění přezdívky uživatele, kterého potřebuje přihlášený uživatel kontaktovat. Přepínání mezi konverzacemi dochází pomocí kliknutí na přezdívku uživatele.

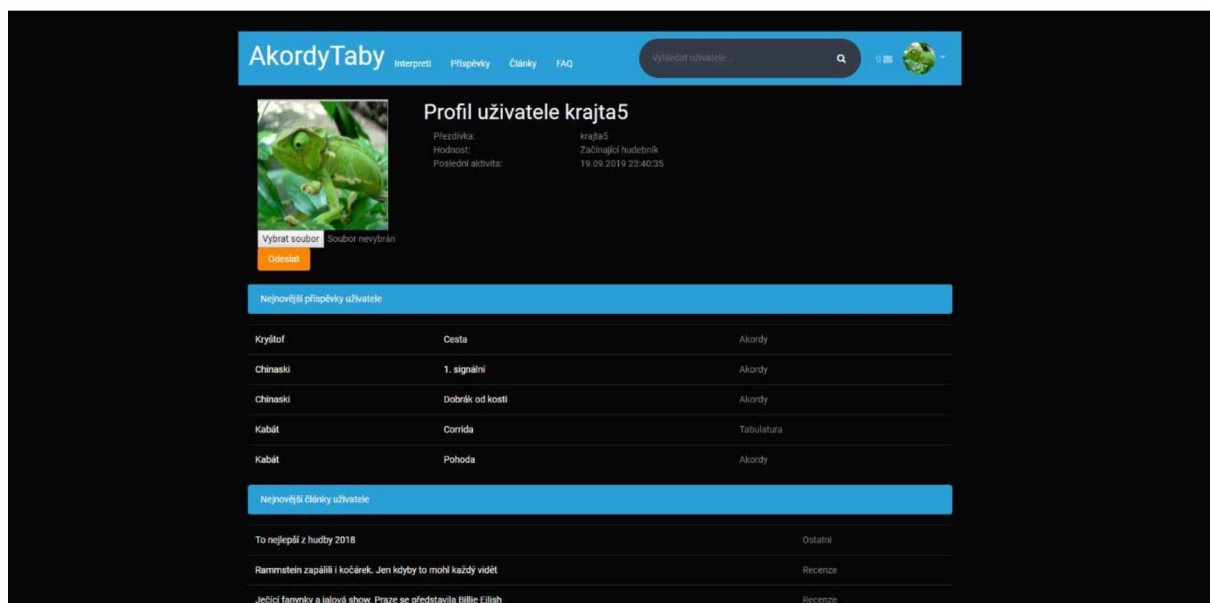


Obrázek 80 - implementovaný systém zpráv

14.9 Profil uživatele

Profil uživatele zobrazuje základní informace o zaregistrovaném uživateli. Na profilu je zobrazena hodnota uživatele, která je závislá na množství schválených příspěvků přidanych uživatelem. Uživatel má možnost změnit vlastní profilové foto, které musí splňovat určitá

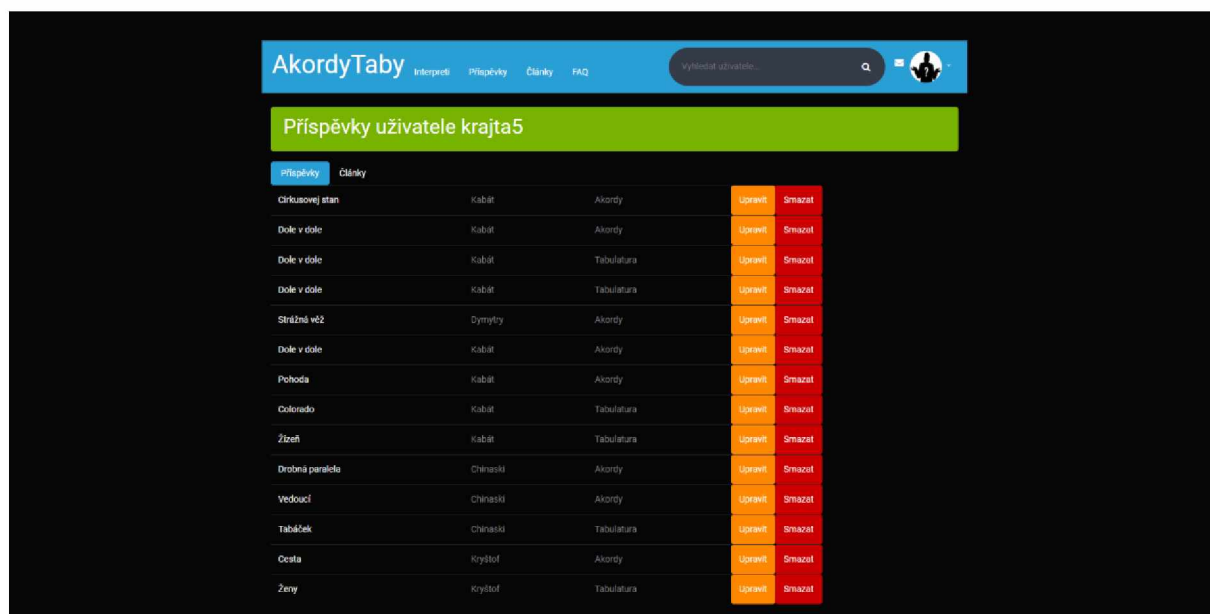
pravidla. Prvním pravidlem je typ souboru png nebo jpg, druhé pravidlo je spojené s maximální povolenou velikostí souboru. Při porušení jednoho z pravidel bude uživatel informován o chybovém stavu a jeho příčině. Následující částí profilu jsou nejnovější příspěvky a články uživatele, které je možné přímo otevřít pomocí kliknutí na název písně, interpreta nebo titulku článku.



Obrázek 81 - profil uživatele

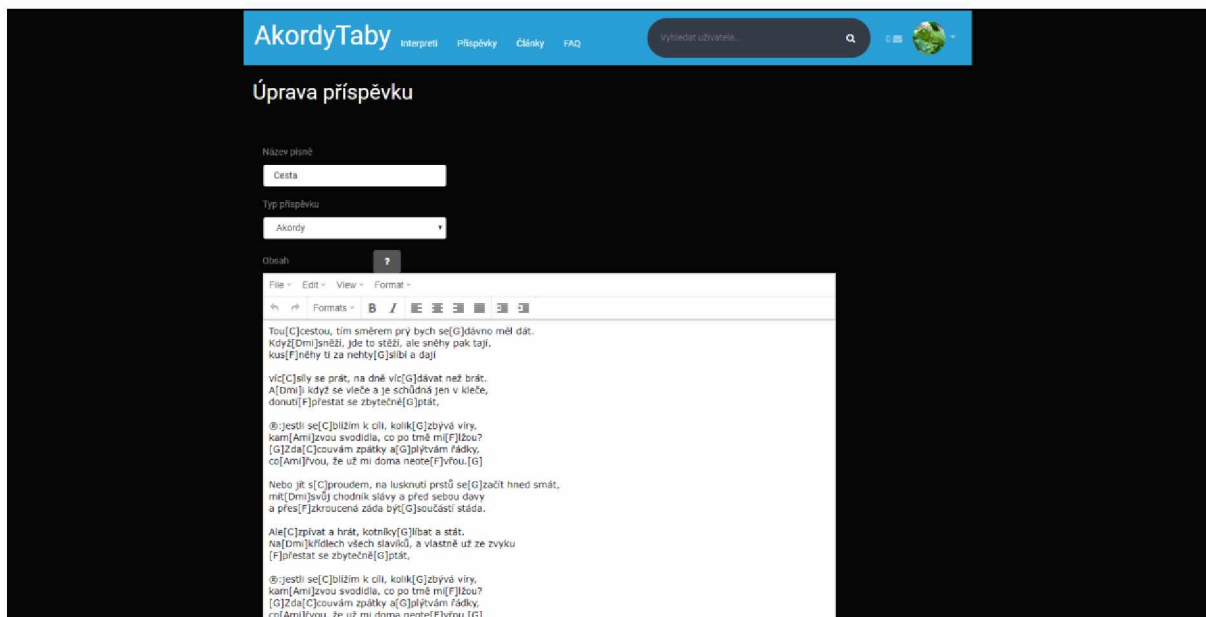
14.10 Vlastní příspěvky

Vlastní příspěvky zobrazují písně a články přihlášeného uživatele. Mezi jednotlivými



Obrázek 82 - vlastní příspěvky

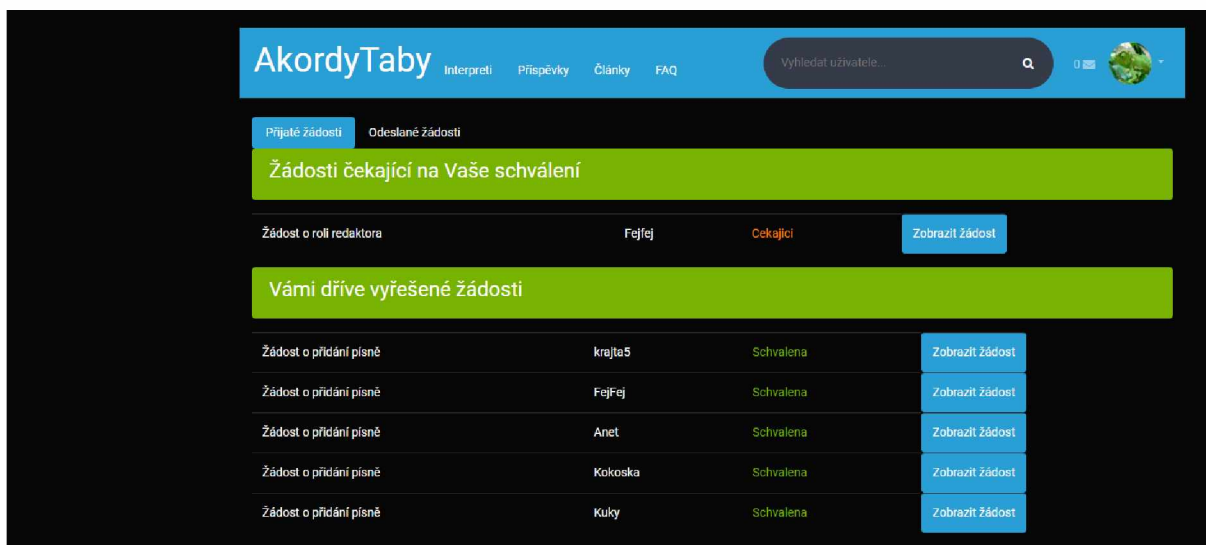
kategoriemi lze přepínat pomocí tlačítka plnicí funkci přepínače. Veškeré písně i články je možné zobrazit, editovat nebo trvale smazat. Obsahy tabulek jsou řazeny sestupně dle data vložení.



Obrázek 83 – editace písně

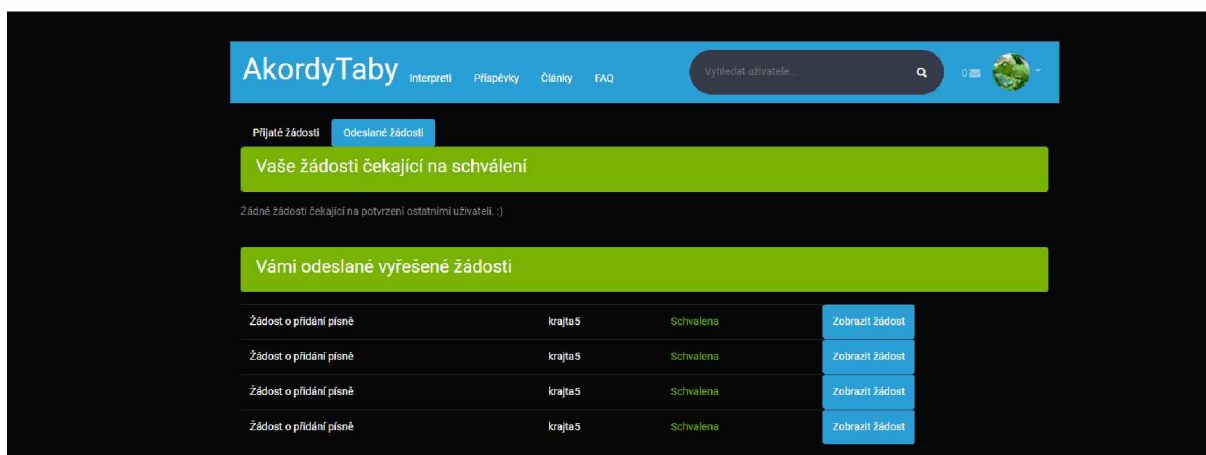
14.11 Správa žádosti

Správa žádostí obsahuje výpis žádostí, rozdělený do čtyř částí. Mezi jednotlivé části patří přijaté nevyřízené, přijaté vyřízené, odeslané nevyřízené a odeslané vyřízené žádosti. Existují dva typy žádostí, žádost o roli redaktora nebo žádost o přidání písně.



Obrázek 84 - přijaté žádosti

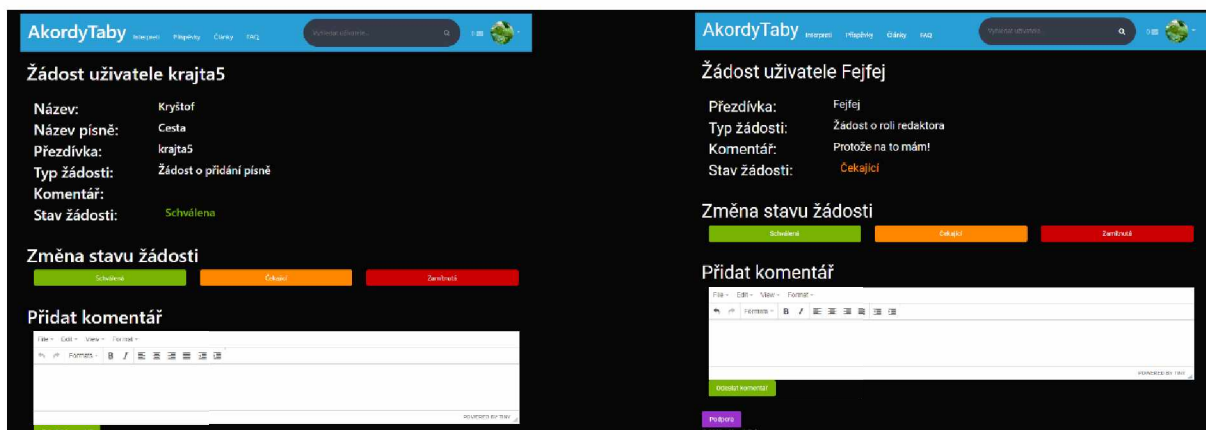
Přijaté žádosti byly odeslány přihlášenému uživateli ke zpracování. Pokud přihlášený uživatel patří mezi uživatele s rolí administrátora, disponuje možností vyhodnotit žádost o roli redaktora. Uživatel s rolí správce interpreta má možnost vyhodnotit zda píseň bude přidána konkrétnímu interpretovi a zviditelněna ostatním uživatelům. Pro zviditelnění písně je zapotřebí označit žádost stavem schválena.



Obrázek 85 - odeslané žádosti

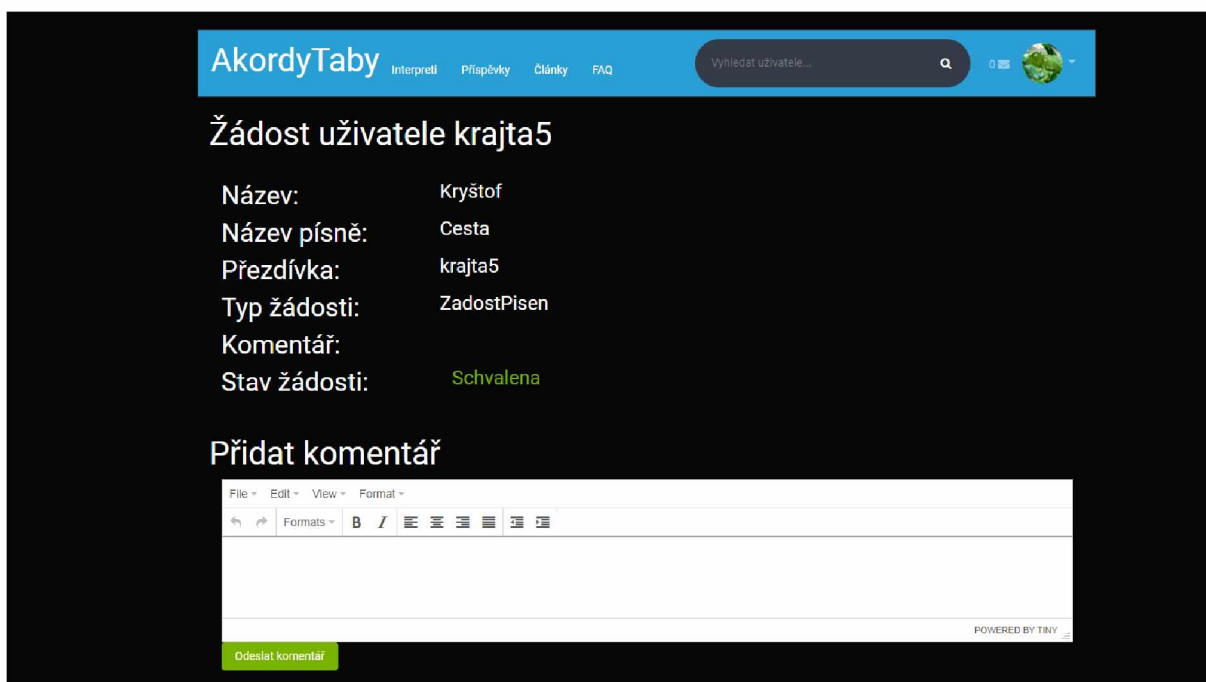
Odeslané žádosti byly vytvořeny přihlášeným uživatelem. Odeslaná žádost čeká na zpracování uživatelem s potřebnou konkrétní rolí. Zobrazit podrobnosti o konkrétní žádosti lze pomocí kliknutí na název žádosti nebo na tlačítko zobrazit žádost.

Zpracovat doručenou žádost je možné pomocí tří tlačítek označující nový stav žádosti, který bude přidělen. Uživatel posuzující žádost potřebuje zobrazit obsah konkrétní žádosti (profil uživatele odesílajícího žádost, obsah konkrétní písně), z tohoto důvodu lze jednotlivé položky žádosti otevřít kliknutím na její hodnotu. Například uživatel s rolí správce interpreta bude zpracovávat žádost o přidání písně, první možnost kterou správce interpreta disponuje je zobrazení konkrétního interpreta pomocí kliknutí na název interpreta. Tímto způsobem lze zkontrolovat zda interpret již vkládanou píseň neobsahuje. Druhá možnost je rozkliknutí názvu písně pro zobrazení obsahu písně. V tomto kroku je možné zkontrolovat dodržení pravidel pro přidání příspěvku nebo jestli schválení žádosti bude užitečné pro ostatní uživatele.



Obrázek 86 - zobrazení konkrétních žádostí

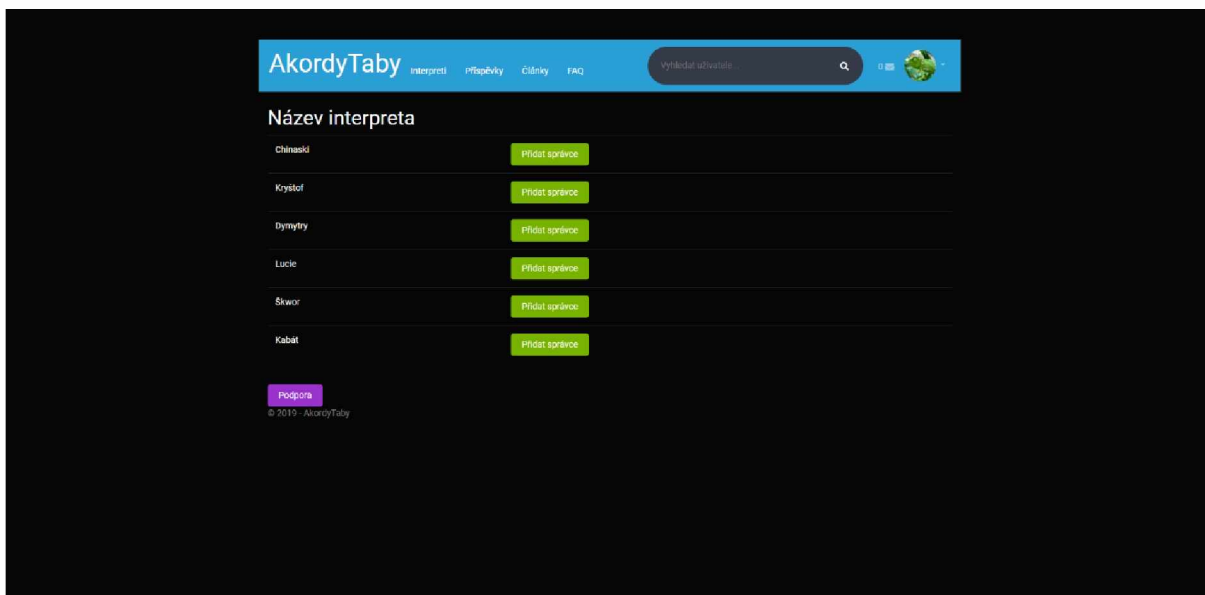
Zobrazení konkrétní odeslané žádosti přihlášeným uživatelem je téměř totožné, pouze neumožňuje změnit stav žádosti.



Obrázek 87 - zobrazení konkrétní odeslané žádosti

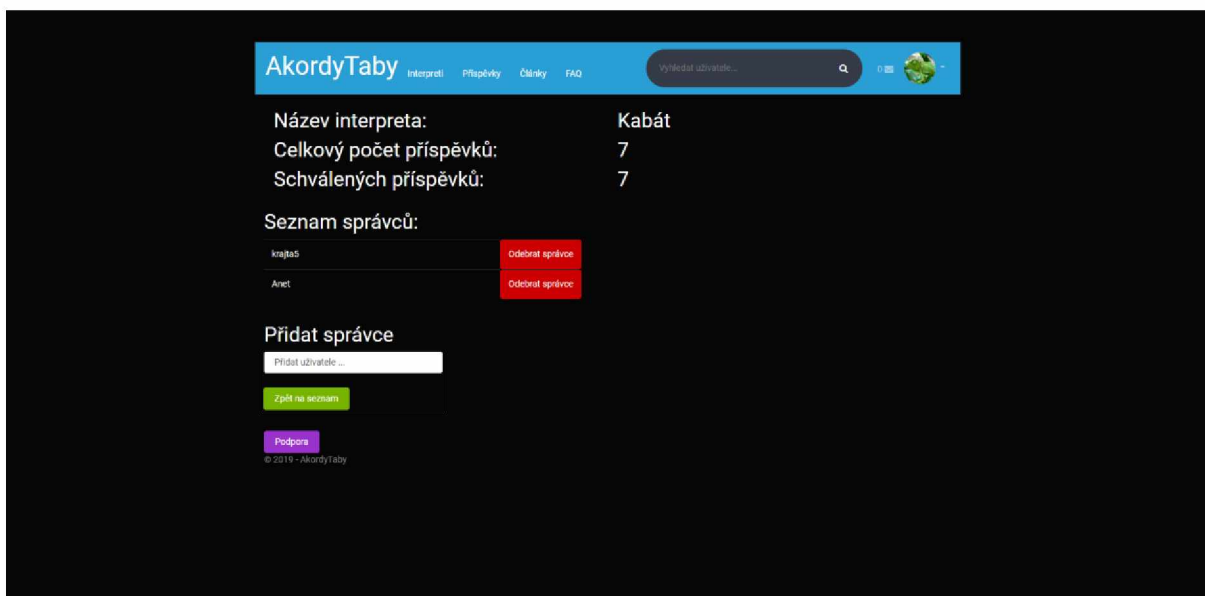
14.12 Správa interpretů

Správa interpretů obsahuje výpis interpretů, které spravuje přihlášený uživatel. To znamená, výpis interpretů, pro které uživatel získal roli správce interpreta. Přihlášený uživatel disponuje možností přidat nového správce interpreta nebo stávajícího správce odebrat.



Obrázek 88 - správa interpretů

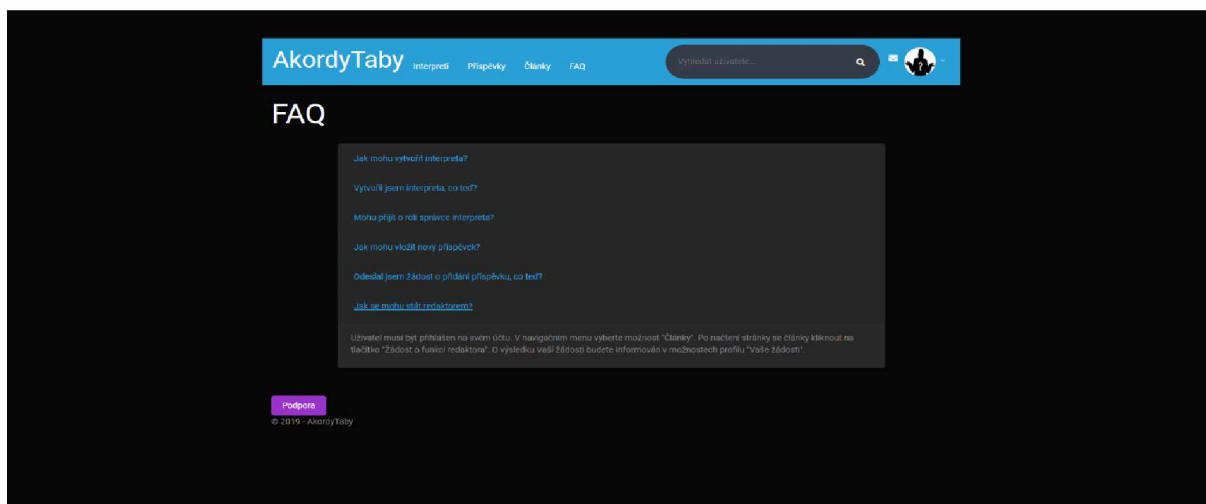
Editace správců konkrétního interpreta zobrazuje informace o názvu interpreta, celkovém počtu písní a počtu schválených písní. Statistiky umožňují sledovat množství přidávaných příspěvků které lze porovnat s hodnotou schválených příspěvků podle které lze určit, zda je aktivita nebo počet správců interpreta dostatečná.



Obrázek 89 - aktualizace správců interpreta

14.13 Často kladené otázky

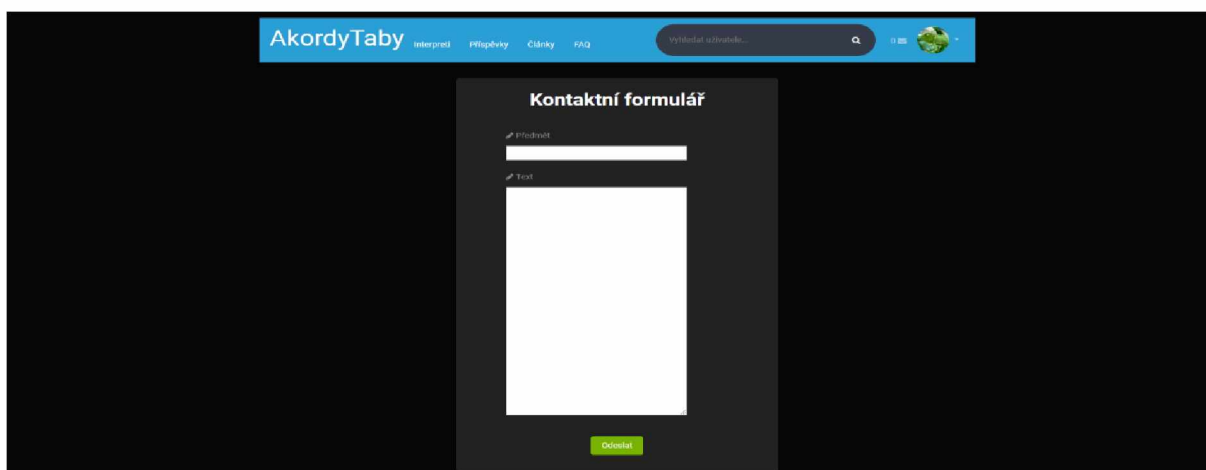
Často kladené otázky je důležitá součást každé webové stránky. Umožňuje nově registrovaným uživatelům získat odpovědi na základní otázky spojené s používáním webové aplikace. Zveřejnění často kladených otázek je výhodné pro administrátora webové aplikace, ale i pro uživatele samotného. Přihlášený uživatel získá odpovědi bez vyplňování kontaktního formuláře nebo čekání na odpověď podpory a administrátor webové aplikace je ušetřen odpovídání na dotazy, které se často opakují.



Obrázek 90 - často kladené otázky (FAQ)

14.14 Podpora

Podporu lze kontaktovat pomocí formuláře, jehož obsahem je pouze předmět a popis problému nebo stížnosti. V případě potřeby zpětné odpovědi uživateli kontaktujícího podporu je součástí odeslaného textu i kontaktní email, který uživatel vyplnil při registraci a potvrdil před prvním přihlášením.



Obrázek 91 - kontaktní formulář

ZÁVĚR

Při vypracování diplomové práce byly využity zkušenosti získané během magisterského studia, spojené s aktuálně používanými technologiemi pro vývoj webových aplikací (ASP.NET, ORM mapování, Entity Framework).

Webová aplikace pracuje s nedostatky již existujících webových aplikací popsaných v rešerši. Spojuje také jejich silné stránky a přidává nové možnosti i funkcionality popsané v uživatelské příručce.

Výsledná webová aplikace je plně funkční, existuje ale velké množství možností, o které lze rozšířit. Například transpozice akordů, interaktivní tabulatury, přidání písní do oblíbených, vytvoření seznamů písní, možnost sdílení příspěvků a další možnosti.

Vývojem webové aplikace jsem získal nové zkušenosti s použitými technologiemi, které využiji i v pracovním prostředí.

SEZNAM POUŽITÉ LITERATURY

- [1] Webová aplikace (Web Application). In: ManagementMania.com [online]. Wilmington (DE) 2011-2019, 18.10.2018 [cit. 27.08.2019]. Dostupné z: <https://managementmania.com/cs/webova-aplikace-web-application>
- [2] Aplikační server (Application server). In: ManagementMania.com [online]. Wilmington (DE) 2011-2019, 20.01.2017 [cit. 27.08.2019]. Dostupné z: <https://managementmania.com/cs/aplikacni-server-aps>
- [3] HTTP (Hypertext Transfer Protocol). In: ManagementMania.com [online]. Wilmington (DE) 2011-2019, 07.12.2016 [cit. 27.08.2019]. Dostupné z: <https://managementmania.com/cs/http-hypertext-transfer-protocol>
- [4] HTTPS (Hypertext Transfer Protocol Secure). In: ManagementMania.com [online]. Wilmington (DE) 2011-2019, 19.12.2016 [cit. 27.08.2019]. Dostupné z: <https://managementmania.com/cs/https-hypertext-transfer-protocol-secure>
- [5] SSL protokol. <https://www.ssl-certifikaty.cz> [online]. [cit. 2019-08-27]. Dostupné z: <https://www.ssl-certifikaty.cz/o-certifikatech/ssl-protokol/>
- [6] ČÁPKA, David. Úvod do C# a .NET frameworku. *Itnetwork.cz* [online]. itnetwork.cz [cit. 2019-08-30]. Dostupné z: <https://www.itnetwork.cz/csharp/zaklady/c-sharp-tutorial-uvod-do-jazyka-a-dot-net-framework>
- [7] GALLOWAY, Jon, Brad WILSON, K. Scott ALLEN a David MATSON. *Professional ASP.NET MVC 5*. Indianapolis, IN: Wrox, a Wiley brand, [2014]. Wrox professional guides. ISBN 11-187-9475-3
- [8] ČÁPKA, David. Úvod do Single Page Application v ASP.NET. *Itnetwork.cz* [online]. itnetwork.cz [cit. 2019-08-30]. Dostupné z: <https://www.itnetwork.cz/csharp/asp-net/mvc/single-page-application/tutorial-uvod-do-asp-net-single-page-application>
- [9] LERMAN, Julia. *Programming Entity framework*. 2nd ed. Sebastopol: O'Reilly, c2010. ISBN 05-968-0726-0
- [10] *Introduction to LINQ Queries (C#)* [online]. 2015 [cit. 2019-09-02]. Dostupné z: <https://docs.microsoft.com/es-cz/dotnet/csharp/programming-guide/concepts/linq/introduction-to-linq-queries>

- [11] ČÁPKA, David. *Migrace a uživatelské role v ASP.NET MVC* [online]. [cit. 2019-09-02]. Dostupné z: <https://www.itnetwork.cz/csharp/asp-net/mvc/zaklady/asp-net-mvc-tutorial-migrace-a-uzivatelske-role>
- [12] DUCKETT, Jon, Gilles RUPPERT a Jack MOORE. *JavaScript & jQuery: interactive front-end web development*. Indianapolis, IN: Wiley, [2014]. ISBN 11-185-3164-7
- [13] *Lekce 1 - Úvod do jQuery* [online]. [cit. 2019-09-02]. Dostupné z: <https://www.itnetwork.cz/javascript/jquery-zaklady/javascript-tutorial-funkcionalni-programovani-a-jquery-webova-kalkulacka>
- [14] *Lekce 1 - Úvod do AJAXu* [online]. [cit. 2019-09-02]. Dostupné z: <https://www.itnetwork.cz/javascript/ajax/uvod-do-ajaxu>
- [15] GROFF, James R. a Paul N. WEINBERG. *SQL: kompletní průvodce*. Brno: CP Books, 2005. Programování (CP Books). ISBN 80-251-0369-2
- [16] ČÁPKA, David. *Lekce 1 - Úvod do databází v C#* [online]. [cit. 2019-09-03]. Dostupné z: <https://www.itnetwork.cz/csharp/databaze/c-sharp-tutorial-uvod-do-databazi>
- [17] SendGrid–odesílání a přijímání e-mailů v Azure Web Apps. <https://blogs.msdn.microsoft.com> [online]. 2015 [cit. 2019-09-03]. Dostupné z: <https://blogs.msdn.microsoft.com/vyvojari/2015/05/06/sendgridodesln-a-pijmn-e-mail-v-azure-web-apps/>
- [18] ASP.NET MVC Folder Structure. *Tutorialsteacher* [online]. [cit. 2019-09-03]. Dostupné z: <https://www.tutorialsteacher.com/mvc/mvc-folder-structure>
- [19] BPMN. *Documents Associated with BPMN* [online]. 1997 - 2016 [cit. 2019-09-07]. Dostupné z: <http://www.omg.org/spec/BPMN/2.0.2>
- [20] *Frequencies for equal-tempered scale* [online]. [cit. 2019-09-06]. Dostupné z: <http://pages.mtu.edu/~suits/notefreqs.html>
- [21] Tóny. <https://dusan.pc-slany.cz/> [online]. [cit. 2019-09-07]. Dostupné z: <https://dusan.pc-slany.cz/hudba/tony.htm>