

Návrh aplikace pro hudební portál

Pavel Knotek



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav počítačových a komunikačních systémů
Akademický rok: 2020/2021

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení:	Pavel Knotek
Osobní číslo:	A17722
Studijní program:	B3902 Inženýrská informatika
Studijní obor:	Informační technologie v administrativě
Forma studia:	Prezenční
Téma práce:	Návrh aplikace pro hudební portál
Téma práce anglicky:	A Design Application for Music Portal

Zásady pro vypracování

1. Provedte rešerši existujících řešení.
2. Vypracujte stručný rozbor technologií, které budou použity k návrhu.
3. Provedte rozbor a analýzu požadavků na zvolené řešení.
4. Zpracujte model navržené aplikace.
5. Věnujte pozornost zabezpečení aplikace.

Forma zpracování bakalářské práce: **Tištěná/elektronická**

Seznam doporučené literatury:

1. NEUSTADT, Ila; ARLOW, Jim. *UML 2 a unifikovaný proces vývoje aplikací*. Computer Press, Albatros Media as, 2016.
2. JOHNSON, Glenn. *Programming in HTML5 with JavaScript and CSS3: training guide*. Redmond, Wash.: Microsoft, 2013. ISBN 978-0735674387.
3. UNHELKAR, Bhuvan. *Software engineering with uml*. Auerbach Publications, 2017.
4. LETT, Jacob. *Bootstrap 4 Quick Start: A Beginners Guide to Building Responsive Layouts with Bootstrap 4*. Bootstrap Creative, 2018.
5. JAKOBUS, Benjamin. *Mastering Bootstrap 4: Master the latest version of Bootstrap 4 to build highly customized responsive web apps*. Packt Publishing Ltd, 2018.
6. BEN-GAN, Itzik; DAVIDSON, Louis; VARGA, Stacia. *MCSA SQL Server 2016 Database Development Exam Ref 2-pack: Exam Refs 70-761 and 70-762*. Microsoft Press, 2017.

Vedoucí bakalářské práce: **doc. Ing. Petr Šilhavý, Ph.D.**
Ústav počítačových a komunikačních systémů

Datum zadání bakalářské práce: **15. ledna 2021**

Termín odevzdání bakalářské práce: **19. května 2021**

doc. Mgr. Milan Adámek, Ph.D. v.r.
děkan



doc. Ing. Martin Sysel, Ph.D. v.r.
garant oboru

Ve Zlíně dne 15. ledna 2021

Jméno, příjmení: Pavel Knotek

Název bakalářské: Návrh aplikace pro hudební portál

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové/bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová/bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou/bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování diplomové/bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové/bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové/bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne

Pavel Knotek v.r.
podpis diplomanta

ABSTRAKT

Tato bakalářská práce se věnuje návrhu aplikace pro hudební portál. V teoretické části práce naleznete popis již existujících aplikací napříč celým internetem. Dále obsahuje rozbor technologií, které jsou použity pro tento návrh. V této práci se věnuji také zabezpečení aplikace před možnými hrozbami. V praktické části je zhotovena analýza požadavků aplikace, kde jsou detailněji zpracovány funkcionality aplikace. Vytvořený a popsáný je zde model databáze, která může být následně použita pro aplikaci. V poslední fázi praktické části je zhotoven model navržené aplikace. Při vytváření praktické části bakalářské práce byly použity jazyky HTML, CSS, JS a SQL, které jsou doplněny o Bootstrap.

Klíčová slova: webová aplikace, analýza požadavků, databáze, HTML, CSS, JS, SQL, Bootstrap.

ABSTRACT

This bachelor thesis focuses on the design of an application for a music portal. In the theoretical part of the thesis you will find a description of already existing applications across the internet. It also contains an analysis of the technologies that are used for this design. In this thesis I also discuss the security of the application from possible threats. In the practical part, an analysis of the application requirements is made where the functionalities of the application are elaborated in detail. A database model is created and described which can be used for the application. In the last part of the practical part, a model of the proposed application is made. In the creation of the practical part of the bachelor thesis, the languages HTML, CSS, JS and SQL were used, which are complemented by Bootstrap.

Keywords: web application, requirements analysis, database, HTML, CSS, JS, SQL, Bootstrap.

Rád bych chtěl poděkovat svému vedoucímu práce doc. Ing. Petr Šilhavému, Ph.D. za jeho pomoc a trpělivost při vypracovávání bakalářské práce. Mé poděkování patří také mé rodině a blízkým přátelům za pomoc a podporu při studiu.

Motto:

„Je těžké čekat na něco, co víte, že se možná nikdy nestane. Ale je ještě těžší vzdát se, když víte, že to je to jediné, co opravdu chcete.“

– neznámý autor

Prohlašuji, že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

OBSAH

ÚVOD.....	9
I TEORETICKÁ ČÁST.....	10
1 REŠERŠE EXISTUJÍCÍCH HUDEBNÍCH APLIKACÍ	11
1.1 SPOTIFY.....	11
1.1.1 Verze zdarma	11
1.1.2 Placená verze.....	12
1.1.3 Výhody placené verze	12
1.1.4 Aplikace	13
1.2 SOUNDCLOUD	14
1.2.1 Dostupnost.....	14
1.2.2 Aplikace	15
1.3 YOUTUBE MUSIC	16
1.3.1 Dostupnost.....	17
1.3.2 Aplikace	18
2 ROZBOR TECHNOLOGIÍ.....	20
2.1 HTML.....	20
2.1.1 HTML5	20
2.1.2 Základní struktura kódu	21
2.1.2.1 Html	21
2.1.2.2 Head	21
2.1.2.3 Title.....	21
2.1.2.4 Body.....	21
2.1.3 Úprava textu	22
2.1.3.1 Nadpisy	22
2.1.3.2 Odstavce.....	22
2.1.3.3 Manipulace s textem	22
2.1.4 Seznamy	22
2.1.4.1 Číslované seznamy	23
2.1.4.2 Odrážkové seznamy	23
2.1.5 Odkazy	23
2.1.6 Obrázky	23
2.1.7 Tabulky	23
2.2 CSS.....	24
2.2.1 Výhody využívání CSS	24
2.2.2 Syntaxe	24
2.2.3 Způsoby použití.....	24
2.2.3.1 Inline CSS	25
2.2.3.2 Vestavěné CSS.....	25
2.2.3.3 Externí CSS.....	25
2.3 SQL	25
2.3.1 Základní pojmy	26
2.3.2 Datové typy v SQL.....	26
2.3.3 Nejdůležitější příkazy SQL	27

2.4	UML	28
2.4.1	Use Case diagram (případ užití).....	29
2.4.1.1	Prvky diagramu.....	29
2.4.1.2	Use case Specifikace (specifikace případu užití).....	32
2.4.2	Class diagram (diagram tříd).....	33
2.4.2.1	Třídy.....	33
2.4.2.2	Atributy	33
2.4.2.3	Operace	34
2.4.2.4	Vztahy mezi třídami.....	34
2.4.2.5	Asociace.....	34
2.4.2.6	Agregace	34
2.4.2.7	Kompozice	34
2.4.2.8	Generalizace.....	35
2.4.2.9	Násobnost.....	35
2.5	JAVASCRIPT	36
2.5.1	Výhody a nevýhody	37
2.5.2	Využití.....	38
2.6	BOOTSTRAP.....	38
3	ZEBEZPEČENÍ A HROZBY ÚTOKŮ NA WEBOVOU APLIKACI	39
3.1	HTTPS.....	39
3.1.1	Jak se bránit.....	39
3.2	SQL INJECTION	40
3.2.1	Zabezpečení proti SQL injection	40
3.2.2	Ukázka útoku	40
3.3	CROSS SITE SCRIPTING (XSS).....	42
3.3.1	Jak chránit uživatele	43
3.4	ÚTOK HRUBOU SILOU NEBOLI DDoS.....	44
3.4.1	Jak se bránit proti DDoS útoku	44
II	PRAKTICKÁ ČÁST	45
4.1	FUNKČNÍ POŽADAVKY	46
4.1.1	Obecné funkční požadavky	47
4.1.2	Uživatelské požadavky.....	48
4.1.3	Správa aplikace	50
4.2	NEFUNKČNÍ POŽADAVKY	51
4.2.1	Bezpečnost systému	51
4.2.2	Jednoduchost systému	52
4.2.3	Požadavky pro provoz.....	53
4.3	USE CASE MODEL.....	54
4.3.1	Scénář UC01: Registrace uživatele	54
4.3.2	Scénář UC02: Zobrazení registračního formuláře	55
4.3.3	Scénář UC03: Přihlášení	56
4.3.4	Scénář UC04: Zobrazení přihlašovacího formuláře.....	56
4.3.5	Scénář UC05: Změna uživatelských údajů	57
4.3.6	Scénář UC06: Změna hesla	58
4.3.7	Scénář UC07: Obnova zapomenutého hesla	59
4.3.8	Scénář UC08: Přehrávání hudby	60

4.3.9	Scénář UC09: Vyhledávání hudby	61
4.3.10	Scénář UC10: Zobrazení základních informací o skladbách	62
4.3.11	Scénář UC11: Zobrazení základních informací o interpretech	62
4.3.12	Scénář UC12: Zobrazení základních informací o hudebních žánrech	63
4.3.13	Scénář UC13: Přidávání skladeb	64
4.3.14	Scénář UC14: Vyhledávání obsahu podle potřeb uživatele	65
4.3.15	Scénář UC15: Podle skladeb	66
4.3.16	Scénář UC16: Podle interpretů	66
4.3.17	Scénář UC17: Podle žánrů	67
4.3.18	Scénář UC18: Editace obsahu	68
4.3.19	Scénář UC19: Správa uživatelů	69
4.3.20	Scénář UC20: Přehled statistik	69
4.4	DATABÁZE	70
4.4.1	Diagram databáze	70
4.4.2	Vytvoření databáze	71
4.4.2.1	Tabulka Uživatel	71
4.4.2.2	Tabulka Interpret	72
4.4.2.3	Tabulka Album	72
4.4.2.4	Tabulka Žánry	73
4.4.2.5	Tabulka Skladba	73
4.4.2.6	Tabulka Skladba_zanr	74
4.4.2.7	Tabulka Role_uzivatele	74
4.4.2.8	Tabulka Hodnoceni_skladby	75
5	MODEL NAVRŽENÉ APLIKACE	76
5.1	HLAVNÍ STRÁNKA	76
5.2	PŘIHLÁŠENÍ A REGISTRACE	77
5.3	STRÁNKA SKLADBY	78
5.4	STRÁNKA INTERPRETI	79
5.5	STRÁNKA ŽÁNRY	80
5.6	PROFIL UŽIVATELE	80
	ZÁVĚR	81
	SEZNAM POUŽITÉ LITERATURY	82
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	87
	SEZNAM OBRÁZKŮ	88
	SEZNAM TABULEK	89
	SEZNAM PŘÍLOH	90

ÚVOD

Tato bakalářská práce se věnuje návrhu aplikace pro hudební portál. Téma této práce jsem si vybral z důvodu mého zájmu o hudbu a taky z důvodu nespokojenosti s nabízenými aplikacemi. Velkou nevýhodou dnešních hudebních aplikací je, že jsou placené a většině případů se po každé odehrané skladbě přehraje reklama, což značně narušuje uživatelský prožitek. V hudebních aplikacích je pro začínajícího umělce téměř nemožné se protlačit, protože aplikace vyzdvihují jen známá jména a ostatní umělci zůstávají v pozadí. Já sám hudbu vytvářím a jsem členem hudební skupiny, se kterou pořádáme koncerty, na kterých se vystřídala spousta světových producentů hudby. Hudební aplikace využívám každý den, proto mě napadlo že bych se mohl pokusit sám nějakou vytvořit.

V první části se budu zabývat rešerší existujících řešení. Zaměřil jsem se na ty největší z největších jako je například Spotify nebo SoundCloud. U každé aplikace bude popsána jejich dostupnost pro běžného uživatele a funkce.

V následující kapitole naleznete technologie, které byly použity pro tento návrh. Jedná se o HTML, CSS, SQL, JS a Bootstrap. U každé technologie nebude chybět její popis a způsob, jak se s ní pracuje.

Ve třetí části se budu zabývat zabezpečením aplikace a hrozbami spojenými s touto problematikou. Představím základní způsoby zabezpečení aplikace a možné hrozby napadnutí webových aplikací. U hrozeb vysvětlím, jak fungují a jak se proti nim dá bránit.

Čtvrtá kapitola se zaměřuje na předběžnou analýzu požadavků na aplikaci. V tomto bodě popíšu funkční i nefunkční požadavky a případ užití aplikace. Dále se budu zabývat návrhem a vypracováním databáze pro tuto aplikaci.

V poslední části navrhnu samotnou aplikaci za pomoci již zmíněných technologií. Bude se jednat o předběžnou designovou verzi aplikace, která pak může posloužit jako inspirace pro programátora, který si bude chtít vytvořit hudební aplikaci.

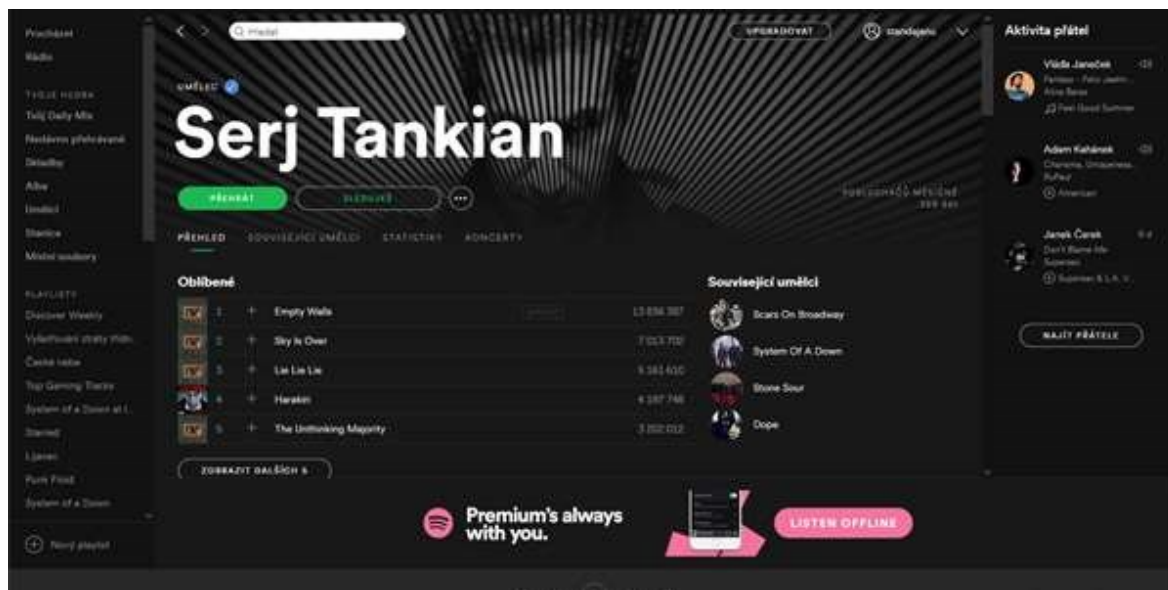
I. TEORETICKÁ ČÁST

1 REŠERŠE EXISTUJÍCÍCH HUDEBNÍCH APLIKACÍ

V dnešní době existuje velká řada hudebních aplikací. Ty nejlepší a největší pocházejí ze zahraničí, ale disponují i českou podporou a překladem. Ve většině případů fungují podobně jako klasické přehrávače například ve Windows. Uživatelé mají k dispozici téměř nekonečnou databázi hudby. Bohužel ve většině případů se po přehrání určitého počtu skladeb spustí reklama, to do velké míry snižuje uživatelské prožitky.

1.1 Spotify

Spotify je jedna z nejoblíbenějších streamovacích aplikací určených pro poslech hudby. Služba je k dispozici jak pro počítač, tak pro mobilní telefony, tablety a ostatní elektronická zařízení s připojením k internetu. Na Spotify nalezneme nespočet hudebních skladeb od interpretů z celého světa. Všechny tyto skladby si můžeme poslechnout zcela zdarma. Spotify nabízí placenou a neplacenou verzi. [1]



Obrázek 1 – Spotify aplikace [2]

1.1.1 Verze zdarma

Pro základní funkce stačí pouhá registrace přes email. Po registraci můžeme začít poslouchat hudbu. V této verzi můžeme poslouchat pouze online skladby, které se nachází v playlistech od Spotify. Tyto playlisty jsou obvykle seskládány podle žánrů hudby a jejich počet je omezený. Výhodou je vložený algoritmus umělé inteligence, který na základě našeho hudebního vkusu, tedy toho, co rádi posloucháme, sestavuje náš osobní playlist, do kterého vkládá naše oblíbené skladby. Nevýhodou je nemožnost přeskakování skladeb v daném playlistu.

Můžeme pouze zvolit playlist zaměřený na určitou hudbu a poslouchat. Tato služba je sice zcela zdarma, ale musíme počítat s tím, že po přehrání několika skladeb se přehraje audio reklama, která je vložena mezi sklady. Bohužel velkou nevýhodou je snížená kvalita hudby. Ten, kdo má citlivější vnímání hudby, bude zklamaný, protože maximální tok dat je 160 kbps/s. [3]

1.1.2 Placená verze

Spotify Premium si můžeme pořídit v jednom ze tří balíčků. První s názvem Individual stojí 5,99 EUR měsíčně. V tomto balíčku máme pouze jeden účet, tudíž se můžeme v jeden okamžik připojit pouze na jednom zařízení. Dalším typem je Family balíček. Tento balíček stojí 8,99 EUR měsíčně. Za tuto částku získáme přístup až k šesti účtům, což je výhodné zejména pro rodiny nebo skupinu kamarádů, kteří si chtějí pořídit tuto aplikaci. Poslední je balíček pro studenty s názvem Students, který si můžeme pořídit za 2,99 EUR měsíčně. Na tento balíček máme nárok pouze tehdy, pokud ještě studujeme. Po ověření, že jsme studenti nám Spotify přidělí jeden uživatelský účet. Všemi těmito balíky se zavazujeme smlouvou na dobu neurčitou, tudíž můžeme službu kdykoliv zrušit. [3] [4]

1.1.3 Výhody placené verze

Spotify Premium má stejnou databázi hudby jako free verze, ale hudba se přehrává v lepší kvalitě, a to minimálně v 320kbps/s. Toto je určitě velké plus, protože při lepším audio vybavení jde velmi poznat rozdíl mezi kvalitní a nekvalitní nahrávkou. U menší kvality zvuku může dokonce dojít i ke ztrátě některých zvukových tónů. Na první pohled většina posluchačů rozdíl nepozná, ale pro hudebně znalé jedince je kvalita zvuku zásadní. Velkou výhodou je i poslech offline. Poslechnout offline si můžeme všechny skladby, které jsme už přehráli na svém účtu. Skladby si lze i stáhnout, což beru jako velké plus, protože je to mnohem levnější než si skladby kupovat jednotlivě. Pokud si ale zrušíme prémiový účet, všechny skladby, které jsme si uložili, nám již nepůjdou přehrát. Oproti free verzi můžeme přehrávat a přeskakovat jakékoliv skladby, a to samozřejmě bez reklam. Spotify nabízí i funkci „connect“. Tato funkce vám umožní přepínat mezi jednotlivými zařízeními, které používáme. Znamená to, že se vám přehraje všechna historie a po přihlášení do nového zařízení budete pokračovat tam, kde jste skončili. Velmi oblíbenou funkcí je propojení s přáteli, díky které se můžete propojit i s vašimi sociálními sítěmi. Samotná aplikace funguje jako sociální síť. Můžete zde například komunikovat s přáteli. Po propojení uvidíte, co právě poslouchají vaši přátelé a vy s nimi můžete sdílet playlisty, které jste vytvořili. [4]

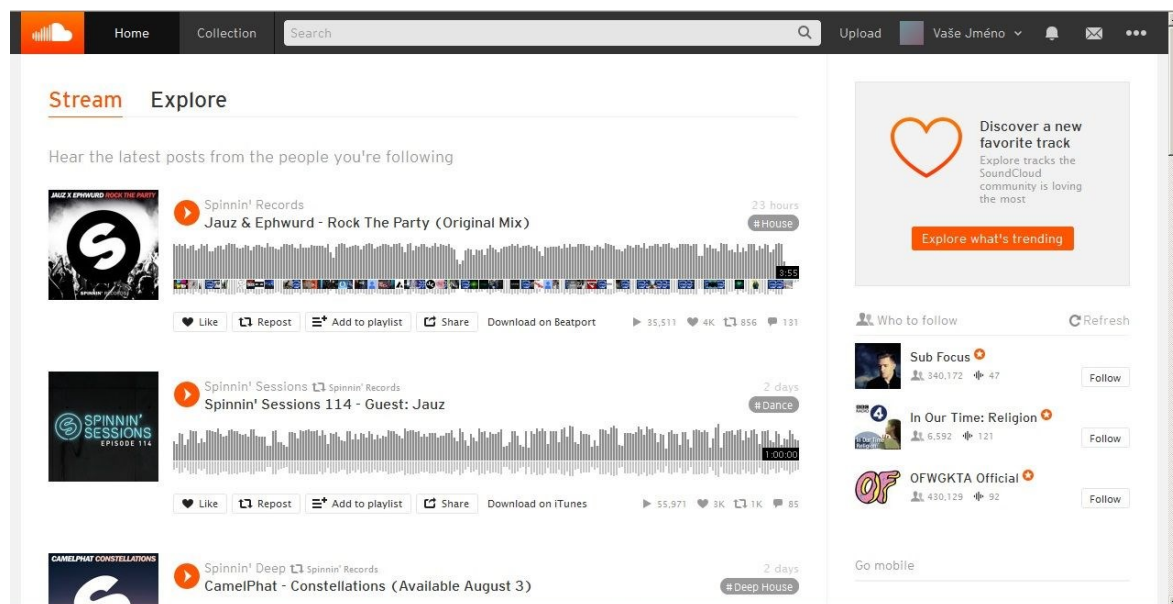
1.1.4 Aplikace

Aplikace je dostupná jak pro Android, tak i pro iOS. K poslechu můžeme využít i počítač nebo chytrou televizi. Aplikaci si jednoduše stáhneme a nainstalujeme. Po instalaci po nás bude vyžadována registrace. Jakmile se registrujeme můžeme se ihned přihlásit a začít poslouchat hudbu. Tu nalezneme buď v přednastavených playlistech, nebo si ji můžeme sami vyhledat. Hledání v této aplikaci je na velké úrovni a každý si jednoduše nalezne přesně to, co potřebuje. Vyhledávat můžeme umělce, alba, playlisty, podcasty, žánry nebo hudbu pro určité příležitosti, jako je například hudba pro cvičení, odpočinek nebo party. Spotify nabízí i žebříček těch nejlépe hodnocených skladeb, nebo sekci čerstvých novinek, kde nalezneme ty nejnovější skladby. Jestliže máme propojený účet s nějakou sociální sítí můžeme vyhledat naše přátele a začít poslouchat to, co právě poslouchají oni, nebo si pustit jeden z jejich playlistů. At už posloucháme jakoukoliv skladbu, vždy si ji můžeme přidat do svých oblíbených skladeb, nebo do svých playlistů. Ke každé skladbě je připojen krátký text s informacemi o dané skladbě, jako například kdy byla písnička vydána, kdo ji složil nebo stručný popis umělce s odkazem na jeho profil. Design aplikace je pojat v moderním provedení. Aplikace je jednoduchá a přehledná, a díky tomu hned naleznete přesně to, co hledáte. Přes nastavení účtu si můžeme aplikaci přizpůsobit svým požadavkům. Je zde spousta nastavení týkající se přehrávání hudby. Například si můžeme nastavit nízkou kvalitu přehrávané hudby, to se hodí například když nejsme připojeni na wifi síť, ale používáme mobilní data. Tato funkce nám ušetří spoustu dat na úkor kvality zvuku, což spousta uživatelů ani nepocítí. Další užitečnou funkcí je normalizace hlasitosti, ta má za úkol automaticky upravit hlasitost přehrávaných písniček na stejnou, protože každá skladba je do databáze vložena s jinou hlasitostí. Jednou z nejlepších funkcí je možnost využití mobilního telefonu jako ovladače pro ostatní zařízení, na kterých máme staženou tuto aplikaci. K tomu stačí být připojený na všech zařízeních přes stejný účet. Tuto funkci si můžeme vysvětlit na jednoduchém příkladu. Představme si, že na televizi máme puštěné písničky a bohužel zrovna hraje skladba, kterou nemáme rádi a chceme ji přepnout. My se ale nacházíme ve vedlejší místnosti. Nemusíme však chodit k televizoru, ale stačí vytáhnout mobilní telefon a přepnout skladbu na něm. V tom okamžiku se přepne skladba na všech zařízeních, na kterých jsme daný okamžik připojeni. Toto beru jako velkou výhodu, protože když se přihlásím na nějakém zařízení, a zapnu hudbu, tak pokračuji tam, kde jsem minule skončil. Spotify lze mimo jiné využít i k poslechu rádia. V této aplikaci máme na výběr mnoho světových i českých radiových stanic. Rádiové stanice můžeme vytvořit i my, a to velmi jednoduše. Stačí si vybrat skladby, které se budou v našem rádiu

pouštět a je hotovo. Každý si tak může vytvořit své vlastní rádio a sdílet ho s ostatními uživateli Spotify nebo na internetu. [5] [6]

1.2 SoundCloud

SoundCloud je služba pro distribuci a přehrávání online hudby. Tato aplikace slouží ke sdílení hudby přímo od interpretů, proto na SoundCloudu nalezneme unikátní skladby, které nikde jinde nenajdeme. SoundCloud se stal velmi populární, protože je zcela zdarma. Díky své popularitě disponuje obrovskou databází hudby. Tuto aplikaci nalezneme především ve webovém provedení. Existují i aplikace do telefonů, ale ty většinou slouží jen k poslechu skladeb. [7]



Obrázek 2 – SoundCloud aplikace [8]

1.2.1 Dostupnost

Pro běžného uživatele je služba dostupná zcela zdarma, dokonce se zde nesetkáme ani s žádnou formou reklamy. K tomu, abychom mohli poslouchat hudbu na této stránce, není potřeba registrace. Pokud se zaregistrujeme, získáme mnoho funkcí navíc. Po registraci každý uživatel získá svůj osobní profil, který funguje jako profil na sociálních sítích. Přes tento profil můžeme komunikovat s ostatními uživateli, sdílet a hodnotit hudbu, a taky sledovat jednotlivé umělce, aby nám neunikly žádné nové písničky. I pro umělce je tato služba zdarma, ale má spoustu omezení. Ve verzi zdarma může každý umělec nahrát maximálně 3 hodiny jakéhokoliv zvuku. Dále se mu zobrazují jen základní statistiky, které se týkají jeho skladeb. Stahování hudby u tohoto typu účtu není nijak omezeno. Stahovat může kdokoliv, a to

v nejlepší dostupné kvalitě. Bohužel není možné stáhnout úplně vše, co bychom chtěli. Stáhnout lze pouze tu skladbu, u které nám to její autor dovolí. Prémiový účet si lze momentálně pořídit už od 8.25 USD ročně nebo za 11 USD měsíčně. Tento účet běžnému posluchači, který jen poslouchá hudbu, nic víc nepřinese. SoundCloud je především orientovaný na hudební umělce, kteří zde vydávají svoji hudbu. Umělec získá mnoho výhod, které mu pomohou k propagaci jeho výtvorů. Každý umělec, který vlastní prémiový účet, může nahrát neomezené množství jakéhokoliv typu zvuku. V tomto účtu lze naplánovat, kdy se vaše skladba vydá. Stačí nahrát skladbu a vybrat si datum a čas vydání. Umělec získá přístup k úplným statistikám, které se týkají hudby. Tuto hudbu mu bude SoundCloud propagovat a tím osloví nové fanoušky. Na co si musíme dát pozor je, že taková propagace může přilákat i lidi, kterým se jeho hudba nebude líbit. My proto můžeme vypnout hodnocení a komentáře u svých nahrávek. Svoje písničky může umělec i zpeněžit. Skladby lze běžně přehrát, ale pokud si ji chceme stáhnout, musíme za ni zaplatit. [9]

1.2.2 Aplikace

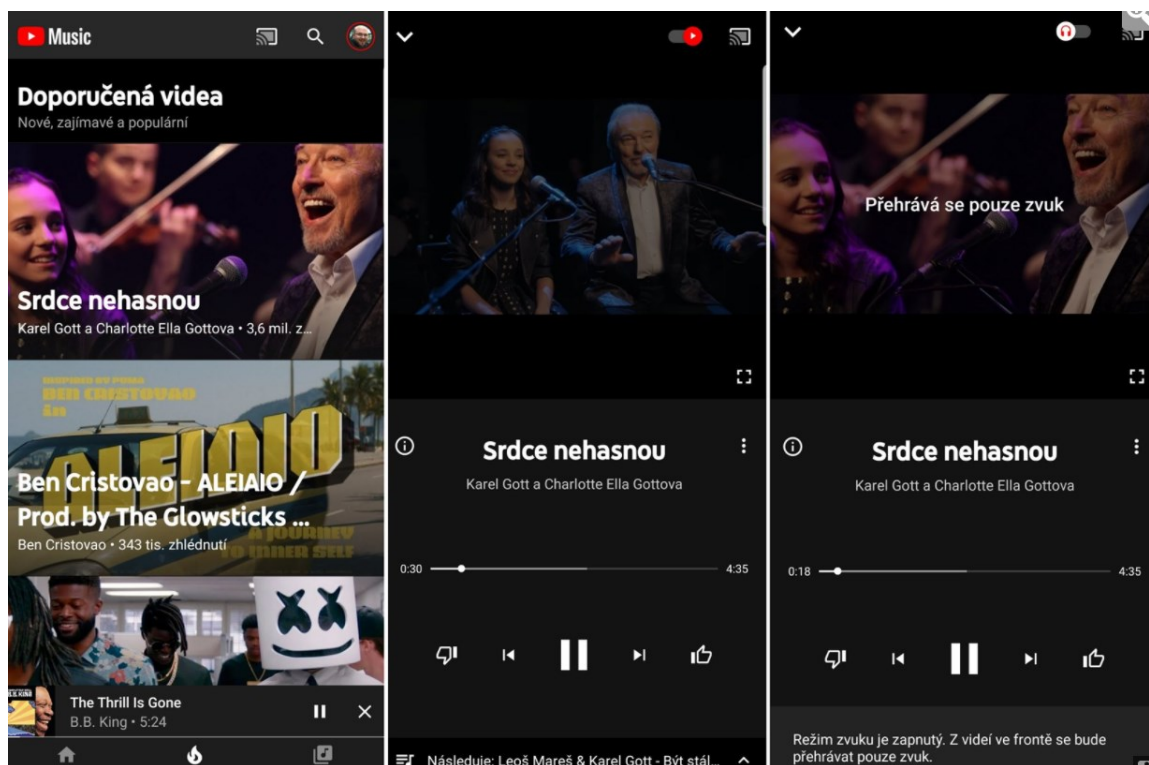
SoundCloud funguje převážně ve webovém provedení, ale lze stáhnout i aplikaci do telefonu. Aplikace je pouze zaměřená na poslech hudby a nenabízí všechny funkce, co jejich webové stránky. Po stažení aplikace je po nás vyžadováno buď přihlášení do našeho účtu, nebo registrace do této služby, což je rozdíl oproti webové verzi. Hudbu v této aplikaci můžeme vyhledat pomocí umělce, žánru, alba nebo playlistu. Skladby nalezneme na úvodní stránce aplikace. Zde jsou zařazeny do rubrik podle různého zaměření. Nalezneme zde například hudbu pro relaxování, odpočinek, posilování nebo hudbu ke spánku. Samozřejmě nechybí ani různé žebříčky. Do těchto žebříčků patří například nejnovější písničky, nejlépe hodnocené skladby, nebo výběr toho, co sám SoundCloud doporučuje. Každý uživatel má svůj osobní profil, na kterém nalezneme informace spojené s tímto uživatelem. Když si uživatel zvolí takové jméno, jaké ještě nikdo nemá, může si přizpůsobit url ve tvaru „soundcloud.com/jehojmeno“. Této funkce využívají především ti interprety, kteří si sem vkládají svůj pseudonym. V této aplikaci funguje i služba „connect“, která nás propojí s ostatními sociálními sítěmi. Tato funkce funguje především pro pohodlnější vyhledávání přátel a sdílení obsahu. SoundCloud funguje podobně jako sociální síť. Každého uživatele můžeme vyhledat a začít sledovat. Sledování volíme z důvodu, aby nám neunikly žádné novinky z jeho tvorby. Uživatelé mezi sebou mohou komunikovat, psát si komentáře, sdílet hudbu a hodnotit ji. Pokud nějakou hudbu ohodnotíme takzvaným „likem“, automaticky se nám přidá do oblíbených skladeb. Takto lze vytvořit knihovna naší oblíbené hudby, kterou

si můžeme kdykoliv otevřít a poslechnout. Také si můžeme vytvářet vlastní playlisty. Do těchto playlistů můžeme vložit jakoukoliv skladbu, kterou nalezneme v této aplikaci. Playlistů můžeme vytvářet neomezené množství. Velkou výhodou je možnost přehrávání offline. Ta funguje tak, že skladby, které jsme pouštěli, se nám uloží do paměti telefonu. Tyto písničky lze přehrát i po odpojení z internetu. Samozřejmostí je i synchronizace zařízení. Ta probíhá automaticky po přihlášení do našeho účtu. Stahovat můžeme zdarma, záleží jenom na umělci, jestli tuto funkci povolí, či nikoli. Pokud ne, většinou je možnost si danou skladbu zakoupit. Cena jednotlivých skladeb není nijak vysoká, obvykle lze jednu písničku pořídít okolo jednoho dolaru. Hudbu stahujeme samozřejmě v té nejlepší dostupné kvalitě, protože SoundCloud si zakládá na dobré kvalitě zvuku. Z toho důvodu lze nahrávat hudbu ve formátech té nejlepší kvality. Co se týče nahrávání skladeb, každý uživatel nahrává skladby na svůj profil. Proces nahrávání skladeb má SoundCloud velmi dobře zpracovaný. Bohužel nahrávat skladby lze pouze přes webovou verzi. V aplikaci není tato funkce povolena. Pro nahrávání stačí jednoduše vybrat soubor, který chceme nahrát a spustí se jeho nahrávání. Mezitím můžeme vyplnit informace o skladbě. Mezi tyto informace patří například jméno skladby, krátký popis a tagy. Zde také nastavujeme viditelnost naší hudby. Buď může být viditelná nebo neviditelná. Když nastavíme, že je viditelná, má k této nahrávce přístup jakýkoliv uživatel. U neviditelné se k této skladbě dostaneme pouze přes odkaz, který získá pouze autor tohoto díla a je jen na něm, jak s odkazem naloží. Dále můžeme přidat obrázek, kterým se bude skladba označovat. Ve většině případů to bývá fotka alba, ze kterého tato písnička pochází. Následně vyplňujeme metadata. Do nich můžeme vložit odkaz na nákup této písničky, na nahrávací společnost a datum vydání. V oprávnění nastavujeme, co mohou uživatelé se skladbou dělat. Například jestli mají oprávnění přidávat komentáře, sdílet a hodnotit tuto skladbu. V poslední řadě nastavíme licenci. Tato licence má za účelem určit, co můžeme a co nemůžeme se staženou skladbou dělat. K většině skladeb získáme licenci pro komerční využití, protože jde zde spousta malých a neznámých umělců, co budou rádi, když jejich skladbu někde zaslechnou. [8] [10]

1.3 YouTube music

YouTube music je aplikace pro online poslech hudby a hudebních videí. YouTube music je relativně nová hudební služba od Googlu, která postupně nahrazuje dosavadní Google Play Music. Velkým lákadlem této služby je obrovská databáze hudby a hudebních klipů. YouTube kombinuje klasickou hudební knihovnu s obsahem z YouTube včetně hudebních

klipů, coverů a živých vystoupení. Tato služba je v základu dostupná pro všechny zdarma. [11]



Obrázek 3 – Aplikace YouTube music [11]

1.3.1 Dostupnost

YouTube music je dostupný jak ve webovém provedení, tak i jako aplikace do telefonu. K využívání této služby v desktopovém provedení není potřeba ani registrace, pouze stačí navštívit jejich webové stránky. Pokud se ale zaregistrujeme, získáme mnoho dalších funkcí. U mobilní aplikace po nás bude požadováno přihlášení nebo registrace. Jelikož YouTube Music je aplikace od firmy Google, tak stačí být zaregistrovaný do emailu od Googlu, a vše se nám automaticky propojí. Služba YouTube music je dostupná zdarma, ovšem s několika omezeními a s přidanou reklamou. Pokud si tuto službu zakoupíme, získáme několik výhod. Tou hlavní výhodou je určitě možnost poslechu hudby bez reklam. Dalším lákadlem je určitě stahování hudby do telefonu. Pro uživatele této služby na telefonu je tu velká výhoda. Ti mohou hudbu poslouchat na pozadí svého telefonu a u toho využívat telefon k jiným účelům. Také lze vypnout displej svého telefonu a jen poslouchat hudbu, což u neplacené verze nejde. Premiový účet lze pořídit za 149 korun českých měsíčně. [12]

1.3.2 Aplikace

Pro to, abychom mohli poslouchat hudbu stačí navštívit webové stránky této služby, nebo si stáhnout aplikaci do svého telefonu. Po stažení aplikace po nás bude vyžadováno přihlášení nebo registrace. Po úspěšné registraci a následném přihlášení se můžeme pustit do poslechu hudby. Aplikace nás automaticky převede do její první karty. První karta se jmenuje Domů a nalezneme v ní nejnovější hudební novinky, dnešní hity, mixy sestavené přímo pro nás, naši oblíbenou hudbu, živá vystoupení a mnoho dalšího. Také zde jsou žebříčky těch nejúspěšnějších skladeb. Playlisty sestavené buď samotnou aplikací nebo uživateli. Dále zde jsou různé kategorie hudby. Každá kategorie je určena pro jinou věc. Nalezneme kategorie například pro party, odpočinek, sport a mnoho dalších. Druhá karta má název Prozkoumat. V této kartě nalezneme nové skladby a alba. Hudba je zde rozdělená do žánrů a nálad. V náladách se nachází skladby přesně určené pro nějakou činnost nebo náladu. Například se jedná o hudbu pro cestování, cvičení, dobítí energie, pro dobrou náladu a spousta dalších kategorií. V žánrech nalezneme všechny možné světové žánry od rocku, jazzu, metalu, elektronické hudby, popu a hip-hopu až po country, vážnou hudbu, folklor a mnoho další žánrů. Ve spodní části jsou nám zobrazeny ty nejnovější hudební videoklipy. Třetí karta se jmenuje Knihovna. V této části se nachází naposledy přehrávané skladby. Zde si můžeme vytvořit naše playlisty, do kterých vkládáme hudbu a následně ji můžeme přehrávat nebo sdílet s našimi přáteli. Do knihovny si také ukládáme skladby. Ty jsou rozděleny podle interpretů nebo alb, ve kterých se nacházejí. Také zde nalezneme kategorii sklady. Ve skladbách se nachází veškerá uložená hudba. Pokud chceme přehrát některou ze skladeb, stačí na ni kliknout a automaticky se přehraje. Po kliknutí budeme převedeni na přehrávač. V přehrávači se zobrazí základní informace o skladbě, jako je její název a jméno interpreta, který tuto skladbu složil. Ke každé skladbě je přidělen obrázek alba, ze kterého skladba pochází, nebo videoklip k dané skladbě. Každou skladbu si můžeme uložit do playlistu. Pokud máme prémiový účet, můžeme ji i stáhnout a později offline přehrát. Nalezneme zde i tlačítko pro sdílení. Po kliknutí na toto tlačítko dostaneme na výběr, na které sociální síti chceme danou skladbu sdílet. Jednotlivé skladby můžeme hodnotit. U přehrávače nalezneme pět tlačítek pro ovládání přehrávání. Každé tlačítko má jinou funkci. První zapne náhodné přehrávání skladeb. Druhé přehraje předchozí skladbu. Třetí tlačítko je tu pro zastavení a spuštění písničky. Následující je pro přeskočení na další skladbu a poslední aktivuje přehrávání jedné skladby pořád dokola. U některých skladeb si můžeme zobrazit i texty, které se v nich zpívají. Dále můžeme zobrazit související skladby s tou, která se nám aktuálně přehrává. Poslední funkce, kterou zde

nalezneme, je seznam písniček, které se budou následně přehrávat. Vyhledávání v této aplikaci funguje velmi jednoduše, stačí napsat jakou skladbu, interpreta nebo album hledáme. Ihned po napsání pár písmen nám začnou vyskakovat nabízené možnosti. Vyhledávat lze i pomocí hlasu. Stačí zapnout tuto funkci a jen říct, co přesně hledáme. [11] [13]

2 ROZBOR TECHNOLIÍ

Mezi základní technologie pro vytváření webových stránek patří HTML a CSS. Ty slouží především ke práci s „frontendem“. Tyto technologie jsou ve většině případů doplněny o JavaScript. Pro vytváření databází se využívá dotazovací jazyk SQL a pro analýzu požadavků UML.

2.1 HTML

Hyper Text Markup Language neboli HTML je hypertextový značkovací jazyk. Jedná se o jeden z prvních jazyků, které byly vytvořeny pro vytváření a editování webových stránek v systému World Wide Web. I když se jedná o jeden z prvních jazyků tohoto typu, je v dnešní době stále velmi oblíbený, a to především pro jeho jednoduchost a lehkou implementaci.

Jeho obsah se skládá ze dvou částí. Jednou z částí je samotný obsah webu a druhou jsou takzvané tagy, které jsou uzavřeny do špičatých závorek (<,>). Tyto tagy mají za úkol formulovat a upravovat samotný obsah stránek. Tagy lze rozdělit do dvou skupin, a to konkrétně do párových a nepárových. Většina tagů je párových. To znamená, že je přesně určen začátek i konec vlivu určitého formátování. [14]

2.1.1 HTML5

Jedná se o pátou verzi HTML, která se stala primárním značkovacím jazykem v online prostředí. Používá se pro jednoduché i strukturované weby a takzvané prezentační weby. HTML5 je nejmodernější, konečná a plně kompletní verze toho jazyka. Účelem této verze je co nejdokonaleji vylepšit jazyk, který bude plně podporovat ty nejmodernější multimédia. Současně se snaží zachovat si svou jednoduchost a snadnou čitelnost jak pro lidi, tak i pro počítače nebo ostatní zařízení, které s tímto jazykem pracují. Nejmodernější verze jazyka HTML nabízí spoustu novinek. Například podporuje audio i video a umožňuje spouštět JavaScript přímo na pozadí. Zavádí nové typy řídicích prvků formulářů. Rozšiřuje se o nové elementy a atributy. Mezi ty nejpodstatnější patří například navigace, čas, zvuk, plátno a mnoho dalších. Pomocí nové verze můžeme zakreslovat geometrické tvary. Využívá místní uložení namísto souborů cookie. Nejdůležitější výhodou je plná podpora všemi novými webovými prohlížeči. [15]

2.1.2 Základní struktura kódu

Každý z html souborů by měl obsahovat základní strukturu html kódu. Součástí této struktury jsou tagy <html>, <head>, <title> a <body>. Tagy jsou primárním prvkem každého html kódu. Zapisování tagů se provádí do hranatých závorek (<,>). Většina tagů je doplněna o atributy. Atributy jsou parametry, které se zapisují dovnitř hranatých závorek. Atributy přidělují elementům jejich vlastnosti. Na začátku každého dokumentu nesmí chybět tag „!doctype“, který je doplněn o verzi použitého html v dokumentu. Tento tag specifikuje DTD (Document Type Definition). Jedná se o manuál pro prohlížeč, který bude zpracovávat dokument. Definuje mu, co za elementy se v dokumentu nacházejí a jak s nimi má pracovat. [16]

2.1.2.1 *Html*

Tímto tagem začíná a také končí dokument. Jedná se o párový tag, což znamená, že musí být ukončen tagem </html>. Všechny ostatní obsah musí být vně tohoto tagu, jinak vytvořený kód nebude fungovat. [17] [18]

2.1.2.2 *Head*

Head neboli hlavička dokumentu obsahuje instrukce především pro prohlížeč. Vymezuje hlavičku dokumentu, kam se zapisují metadata. Je to povinný párový tag. Do hlavičky zapisujeme také další tagy, jako jsou title, link, meta a další. [17] [18]

2.1.2.3 *Title*

Popisuje titulek vytvořené stránky. Jedná se o párový tag. Mezi tagy <title> a </title> je vložen text, který bude brán jako titulek stránky. Titulek se objevuje v záhlaví otevřeného okna stránky a také na primárním panelu jako nadpis stránky. Při vyhledávání se nám zobrazuje právě tento text, který je hlavní při indexování stránky. Tento tag nemůže obsahovat žádné další tagy, jeho součástí je pouze čistý text. [17] [18]

2.1.2.4 *Body*

V těle dokumentu se nachází všechny obsah, který se bude zobrazovat na stránce. Body je párový tag. Do těla jsou vkládány tagy, které se zobrazí i uživatelům stránky. Například jde o odstavce, obrázky, videa, tabulky, různé seznamy a tak dále. Tělo dokumentu obvykle bývá doplněno o jeho atributy. Ty definují vlastnosti tohoto tagu. Například může jít o bgcolor, background, text, link a mnoho dalších. Tag „<bgcolor>“ popisuje barvu pozadí celé

stránky, zatímco „<background>“ vkládá obrázek na pozadí. Další „<text>“ určuje barvu textu a „<link>“ barvu odkazu. [17] [18]

2.1.3 Úprava textu

Formátování textu pomocí HTML je velmi jednoduché. Podle toho, co od textu očekáváme, si vybereme odpovídající tag. Do tagu vložíme požadovaný obsah a ukončíme ho. U formátování textu máme na výběr velkou škálu možností. Například můžeme definovat nadpisy, odstavce, seznamy a tak dále. Lze měnit fonty textu a celkově lze využít téměř všechny možné formáty textu, například podobně jako v Microsoft Word. [19]

2.1.3.1 Nadpisy

Nadpisy se tvoří pomocí tagu „header“. Jedná se o párový tag. Příkazy pro vytváření nadpisů se zapisují jako „<H1>“ – „<H2>“. Jsou definovány od prvního až po šestý stupeň, přičemž první je největší a poslední nejmenší. [18] [19]

2.1.3.2 Odstavce

Pro přehlednost a dobrou čitelnost obsahu stránky se používají odstavce. Píšeme je za pomocí tagu „<p>“. Ten se nachází na počátku každého odstavce. S obsahem, který je vymezen pomocí odstavce, se následně jednoduše manipuluje a upravuje. Odstavce jsou párové a můžeme jim definovat jejich atributy. K lepšímu popsání vlastností odstavců se používá CSS, o kterém bude zmínka později. [18] [19]

2.1.3.3 Manipulace s textem

Text v našem obsahu můžeme libovolně měnit. Většina těchto tagů je párových. Například můžeme změnit font písma, a to za pomocí „font“. Tučné písmo se vytváří s „b“, kurzíva „i“ a podtržení textu „u“. Další širokou nabídku formátů písma nalezneme v CSS. [18] [19]

2.1.4 Seznamy

Seznamem se rozumí několik oddělených položek. Seznamy mohou být číslované nebo mohou mít různé typy odrážek. Lze vytvářet i vnořené seznamy. Stavebním prvkem seznamů je „“. U seznamů vždy nalezneme pouze párové tagy. Pro přesné zobrazení se prvek seznamu musí nacházet uvnitř elementu. Na každý jeden řádek kódu se zapisuje jeden prvek seznamu. Ten se vždy zobrazuje na novém řádku. Do „“ zapisujeme typ seznamu

(„type“) a také startovací hodnotu seznamu („start“). Vlastnosti seznamu můžeme vylepšit pomocí CSS. [20]

2.1.4.1 Číslované seznamy

Položky v číslovaném seznamu budou automaticky číslované. Existují různé druhy číslování. Například se jedná o klasické číslování (značené pomocí 1), velké (A) a malé (a) písmové a římské (I). Tento typ seznamu se zapisuje pomocí „“. [20]

2.1.4.2 Odrážkové seznamy

K položkám v seznamu jsou automaticky přiděleny odrážky. Automaticky mají vzhled puntíku. Tento vzhled však lze změnit pomocí „type“. Odrážky mohou mít podobu plného kolečka (disc), čtverce (square), nebo prázdného kolečka (circle). Odrážkové seznamy se vytvářejí pomocí „“. [20]

2.1.5 Odkazy

Odkazy jsou nedílnou součástí HTML. Po kliknutí na ně, nás automaticky přesměrují na určené místo. Tento párový tag zapisujeme pomocí „<a>“. Do odkazů vkládáme jejich atributy, jako například „href“. Ten nám určuje, co přesně se po kliknutí na odkaz stane, tedy kam budeme přesměrováni. [21]

2.1.6 Obrázky

Do HTML se obrázky vkládají pomocí nepárového tagu „img“. Tento tag má jeden povinný a mnoho nepovinných atributů. Povinný atribut je „src“. Ten odkazuje na umístění obrázku. U obrázků také může nastavit jejich velikost. Výšku definujeme pomocí (height) a šířku pomocí (width). Velikost obrázku se udává v pixelech. K zarovnávání obrázků se používá „align“. Do „align“ se anglicky zapíše požadované umístění, jako například right. Obrázek lze také použít jako rozklikávací odkaz. [22]

2.1.7 Tabulky

Někteří vývojáři tabulky v HTML používají nejen jako klasické tabulky, ale i na rozložení stránek. Tabulku lze vytvářet pomocí párového tagu „<table>“. Ten vymezuje začátek a konec tabulky. Do každé tabulky je potřebné vložit další tagy, jako jsou „<tr>“ a „<td>“. První z nich vytváří jednotlivé řádky tabulky a druhý vytváří její buňky. U tabulek lze definovat

spoustu jejich atributů, jako například „frame“. Frame definuje, jakým způsobem bude tabulka ohraničena. Nebo „rules“, který definuje, jaké bude mít mřížky. [23]

2.2 CSS

Cascading Style Sheets neboli CSS popisuje, jakým způsobem se budou HTML dokumenty zobrazovat po vzhledové stránce. To znamená, že v CSS jsou uloženy informace o vlastnostech elementů a atributů, které nalezneme v HTML souboru. Spolu vytvářejí webovou stránku. Hlavní myšlenkou vytvoření kaskádových stylů je umožnění návrhářům oddělit obsah a strukturu webu. Momentálně se využívá nejaktuálnější verze CSS3. Vytvořený soubor musí mít příponu „.css“ a sám o sobě je k ničemu. Jeho funkce se projeví až po připojení k HTML dokumentu. [24]

2.2.1 Výhody využívání CSS

CSS nabízí spoustu výhod, mezi ty hlavní patří například ušetřený čas, rychlost načítání vytvořeného webu a jednoduchá a přehledná kontrola vizuální stránky webu. Šetřit čas lze definováním určitého tagu. HTML může každému tagu přidělit jeho atributy, to ale zabere spoustu času. U CSS to stačí udělat pouze jednou. Například pokud chceme nadpisy h2 zelenou barvou, tak je stačí definovat pomocí kaskádových stylů. Tím dosáhneme toho, že každému nadpisu h2 bude automaticky přidělena zelená barva. Kódy s využitím CSS bývají oproti čistému html kratší. Kratší kód znamená, že má menší velikost a tím i rychlejší načítání na webových stránkách. V CSS má mnohem širší škálu atributů než HTML. [24]

2.2.2 Syntaxe

CSS se zapisují úplně odlišným způsobem jako HTML, ale to nic nemění na tom, že zapisování CSS je velmi jednoduché. Existují jednoduchá pravidla pro zápis. Každý kód musí být vytvořen za pomoci elementu, atributu a jeho hodnoty. Do jednoho stylu lze aplikovat i více elementů a atributů současně. Elementy pouze stačí oddělit čárkou a atributy se oddělují pomocí středníku. [25]

2.2.3 Způsoby použití

Jsou známé tři způsoby, jak lze použít tento styl, jedná se o:

2.2.3.1 *Inline CSS*

Definice stylu nalezneme přímo uvnitř HTML souboru. Používá se pouze pro jeden konkrétní element. Na ostatní části kódu nemá žádný vliv. Formátování elementu se provádí pomocí tagu „style“. [26]

2.2.3.2 *Vestavěné CSS*

Vkládá se přímo do HTML dokumentu, obvykle do hlavičky. Jedná se o jakýsi seznam stylů, který bude využíván pro HTML soubor. Definují se v něm všechny možné elementy a atributy. Ty pak budou použity pro celý kód stejně. Například pokud zadáme, že všechny nadpisy h1 budou červené a zarovnané na střed, tak veškeré nadpisy v tomto souboru budou automaticky definované právě podle této definice. Do hlavičky dokumentu se zapisují pomocí „style“. [26]

2.2.3.3 *Externí CSS*

Informace o CSS jsou zapsané v samostatném souboru, tudíž je nenalezneme v HTML dokumentu. Na tento soubor se pouze odkazuje, a to pomocí tagu „link“. Odkaz se většinou vkládá do hlavičky dokumentu. V souboru CSS jsou umístěny pouze definice atributů a elementů. Po odkázání na tento soubor se automaticky aplikují všechny definice na celý obsah dokumentu. [26]

2.3 SQL

Strukturovaný dotazovací jazyk neboli SQL je programovací jazyk, který pracuje s databázemi. Tento jazyk umožňuje vytvořit databázi, definovat neboli naplnit ji daty, dotazovat se na tyto data a upravovat je. V tomto jazyce můžeme vytvářet strukturované tabulky, naplňovat jejich pole daty, definovat vztahy a organizovat položky dat. Také umožňuje definovat přístup k údajům, tedy přidělovat a odepírat oprávnění přístupu k určitým datům. Tímto chrání data před znehodnocením nebo dokonce i zničením. Zabraňuje neautorizovanému čtení a manipulaci s nimi. Dále zprostředkovává sdílené využití dat a stará se o hladký průběh činností, přistupuje-li k údajům z databáze více uživatelů zároveň.

SQL je především interaktivní dotazovací jazyk. Pomáhá získat odpovědi na velmi obtížné dotazy takřka okamžitě. Je snadno srozumitelný, protože pracuje s daty v podobě tabulek, což je lehce pochopitelné i pro jeho uživatele. Používá relační databáze, v nichž uživatel

sleduje data, která jsou ve formě soustavy propojených tabulek. Každá z tabulek je znázorněna jako množina dat, která jsou seskupena v řádcích a sloupcích.

Ve značném množství případů je důsledkem činnosti popsané v SQL jakási množina dat z jedné či více tabulek. Po zadání nějakého příkazu se nám zobrazí výsledkový tabulka. Výsledková tabulka vždy nemusí být konečným výsledkem. S touto tabulkou můžeme dál pracovat. Například může sloužit jako množina počátečních dat pro další zpracování.

Příkazy SQL se dají rozdělit do čtyř základních skupin. První skupinou je DDL (Data Definition Language). Příkazy patřící do této skupiny umožňují vytvářet, upravovat a odstraňovat objekty databáze. Další skupinou jsou DML (Data Manipulation Language). V této skupině nalezneme příkazy, které dovolují získávat, ukládat a odstraňovat data v databázi. Předposlední je DCL (Data Control Language). Ta má za úkol správu uživatelských rolí a pravomocí. Poslední z těchto skupin je TCL (Transactional Control Language). Do této skupiny jsou zařazeny příkazy pro řízení databázových transakcí. [27] [28]

2.3.1 Základní pojmy

Jedním z hlavních pojmů je v jazyku SQL příkaz. U jakéhokoliv příkazu vždy začínáme klíčovým slovem. Slovo orientačně znázorňuje, jakou akci určený příkaz realizuje. Za klíčovým slovem pak následuje jeden nebo i více klauzulí. Tyto klauzule definují charakter uskutečňované akce. Taky mohou specifikovat data, s nimiž má daný příkaz dělat. Každá z klauzulí musí začínat klíčovým slovem jako jsou například WHERE nebo FROM. Ne všechny klauzule jsou povinné. V některých případech je můžeme vynechat.

Pojmenování objektů v tomto jazyce má standardní délku jeden až osmnáct znaků. Z těchto znaků vždy musí být první znak písmeno. Jména nesmějí obsahovat interpunkční znaky nebo mezery. Může se nám stát, že jména v naší databázi budeme muset kvalifikovat, protože když se nám budou vyskytovat stejná jména v databázi, tak nebude jednoznačně jasné, na který z objektů databáze se jméno odkazuje. [29]

2.3.2 Datové typy v SQL

V SQL je velmi důležité u každé tabulky určit její datový typ. To znamená, že určujeme, jaká data a jakého druhu lze do daného sloupce (tabulky) dosadit. U některých datových typů také nesmíme zapomenout určit jejich nejvyšší možnou šířku sloupce. U číselných datových typů určujeme jejich šířku, ale i počet desetinných míst.

Mezi ty nejpoužívanější datové typy patří například:

SMALLINT – Celá čísla v délce až 6 číslic (včetně znaménka mínus). Rozsah dovolených hodnot je interval v délce od mínus 32 767 až do 32 767.

INTEGER – Celá čísla v délce až 11 číslic (včetně znaménka mínus), Rozsah dovolených hodnot je interval v délce od mínus 2 147 483 648 až do 2 147 483 647.

CHAR – Znakový řetězec s rozpětím n povolených znaků. Rozsah dovolených hodnot pro n je interval od 1 do 32 767. Pokud neurčíme n tak je automaticky rovno 1.

VARCHAR – Znakový řetězec podobný jako u CHAR s tím rozdílem, že povoluje pracovat s řetězcem různých délek v individuálních řádcích. U tohoto datového typu musíme určit maximální horní hranici délky řetězce.

DATE – Jedná se o datumový datový typ. Datum formátu je zde podle nastavení. Základním nastavením je yyyy/mm/dd (rok/měsíc/den)

TIME – Jedná se o časový datový typ. Je ve formátu hodina/minuta/vteřina. [30]

2.3.3 Nejdůležitější příkazy SQL

1. CREATE DATABASE

Klauzulí **CREATE DATABASE** začínáme vytvářet databázi, tedy pokud zadáme tento příkaz, vytvoří se nám databáze. Pro vytvořenou databázi stanovujeme určité vlastnosti. Určujeme pojmenování hlavního souboru, jež popisuje databázi pro uživatele. Většinou je databáze uložena pouze do jednoho souboru. Pokud chceme ukládat databázi do více souborů, tak musíme definovat sekundární uložení. Taky existuje příkaz **ALTER DATABASE**, který upravuje již vytvořenou databázi.

2. CREATE TABLE

Příkaz **CREATE TABLE** je určen k vytváření nových tabulek v současné databázi. Ten, kdo vytvoří danou tabulku v databázi, tomu jsou přidělena veškerá privilegia a stává se jejím vlastníkem.

3. DROP TABLE

Účelem tohoto příkazu je odstraňování tabulek a indexů. Odkazy na tabulku, které nalezneme ve výrazech SQL, náhledy, jež jsou vytvořené na této tabulce, integrita omezení a uchované procedury, nemohou být odstraněny. Pokud odstraňujeme tabulku, která provádí

nějakou transakci, která není dokončená, nelze tuto tabulku odstranit. Musíme počkat, dokud se transakce nedokončí.

4. ALTER TABLE

Příkaz ALTER TABLE dává možnost upravovat struktury již vytvořených tabulek. Může do nich přidat nebo odebrat jeden nebo i více sloupců. Může mnohonásobně přidávat a odstraňovat data z tabulek.

5. SELECT

Klauzule SELECT je základem pro všechny dotazovací příkazy. Pomocí ní se sestavují veškeré dotazovací příkazy, a to od těch jednoduchých, které obsahují pouze pár příkazů, až po ty nejkomplikovanější, které můžou být zapsány i na mnoho řádků.

6. UPDATE

Příkaz UPDATE edituje data v databázi. Může upravovat jeden nebo i více řádků zároveň. Přidáním klauzule WHERE definujeme řádky, které se mají aktualizovat.

7. DELETE

Příkaz DELETE odstraňuje data z databáze. Lze určit jeden či více řádků k odstranění. Data, která chceme odstranit, vybereme klauzulí WHERE, pokud tak neučiníme, po zadání tohoto příkazu dojde k odstranění všech dat z databáze.

8. INSERT

Příkaz INSERT má za úkol vkládat do databáze nová data. Data lze vkládat po jednom, či mnoho najednou. Data jsou vkládána do příkazu po zadání klauzule VALUES. Po ní následuje závorka, do které zapisujeme nová data. Závorka funguje jako tabulka. V závorce jsou jednotlivá data oddělena čárkou. To znamená, že po napsání čárky budou následující data vkládána do dalšího pole tabulky. [31] [32]

2.4 UML

UML (Unified Modeling Language) je grafický jazyk. Tento jazyk se používá pro navrhování, vizualizaci, specifikaci a dokumentaci programových systémů. V oblasti návrhu a analýzy je standardem, a proto je pro programátory důležité, aby se v tomto jazyce orientovali. Slouží jako užitečný nástroj k usnadnění a zjednodušení návrhu a vývoje informačního systému. UML využívá a pracuje s diagramy. Mezi tyto diagramy patří například diagram tříd,

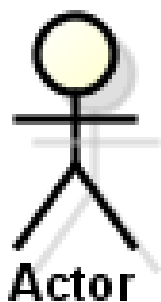
diagram případů užití, objektový diagram, diagram aktivit, sekvenční diagram a mnoho další diagramů. Tyto diagramy se dělí do dvou základních skupin. První skupinou jsou diagramy struktury. Diagramy struktury popisují strukturu celého systému, to znamená, že mají za úkol popsat z čeho se systém skládá. Druhou skupinou jsou diagramy chování. Jejich účelem je co nejlépe popsat chování systému, tedy popisují, jak celý systém funguje. V diagramech chování nalezneme jednu samostatnou skupinu s názvem diagram interakce. Tento diagram má za úkol popsat interakce mezi individuálními částmi systému. [33]

2.4.1 Use Case diagram (případ užití)

Use case diagram popisuje chování celého systému tak, jak ho vidí uživatel tohoto systému. Účelem diagramu je popis funkcionalit systému, tedy toho, co od něj vývojáři, klient nebo my očekáváme. Diagram sděluje, co by měl systém umět, ale neříká, jak to bude dělat. Proto je tento diagram ve většině případů první, který při návrhu informačního systému zhotovujeme. Je velmi důležité se nejdříve domluvit na tom, co má náš systém, aplikace, hra nebo cokoli jiného umět. Až potom co se shodneme na tom, co má aplikace umět, má smysl ptát se, jak toho vlastně dosáhneme. Use Case diagram se skládá z případů užití (Use Case), aktérů (Actors) a vztahy mezi těmito částmi. [34]

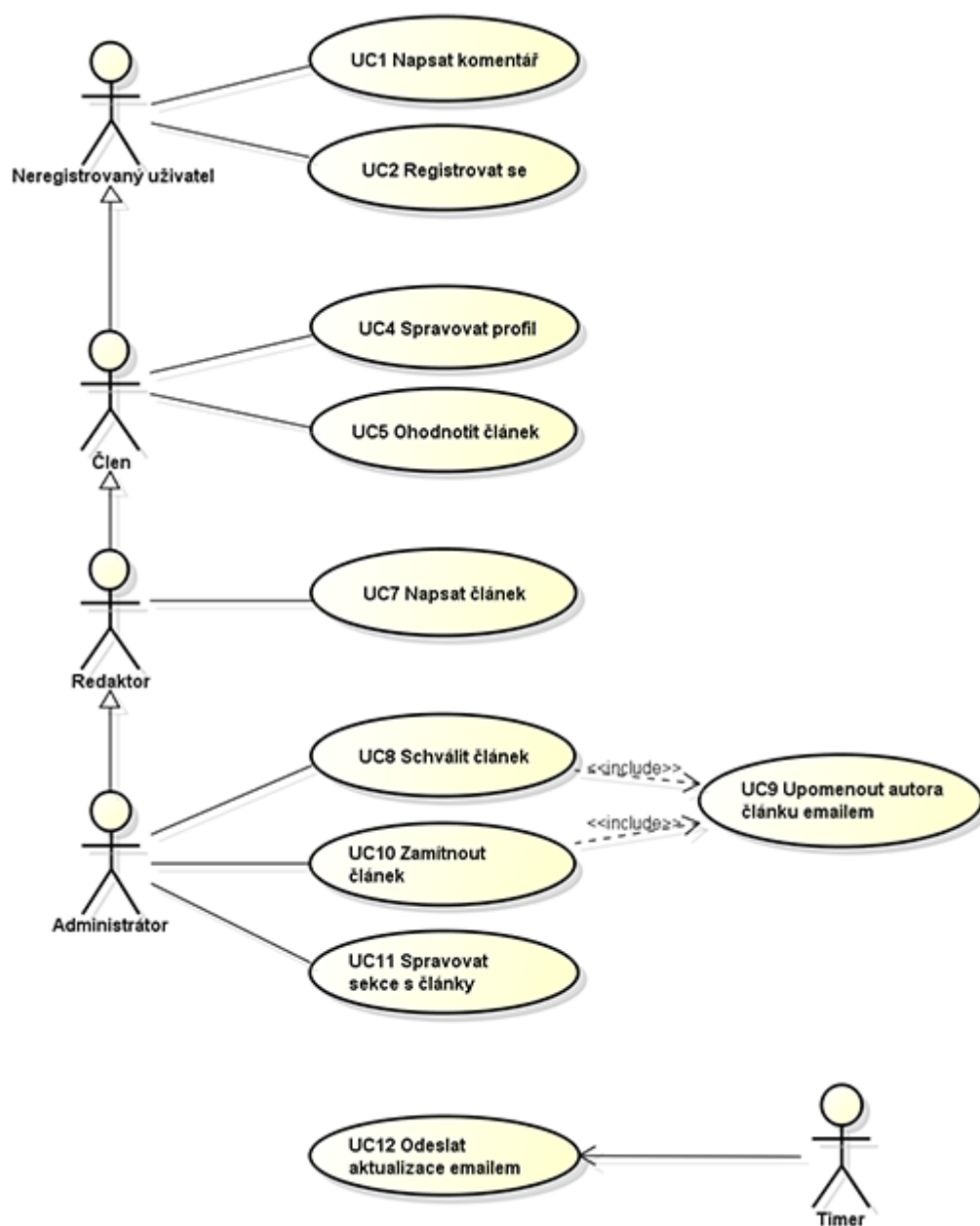
2.4.1.1 Prvky diagramu

Aktér je role, která se dorozumívá se jednotlivými případy užití. V této funkci může být přidělen uživatel nebo externí systém. Aktérem tedy může být uživatel, správce administrátor, server nebo dokonce čas. Aktér pro svůj popis v systému používá dva rovnocenné symboly. První z nich je symbol postavy. Ten se většinou používá pro aktéra představující osobu, tedy fyzickou bytost. Druhým symbolem je čtverec. Čtverec, na nějž je ještě aplikován stereotyp <<systém>> se používá pro externí systémy. Oba tyto symboly musí vždy v diagramu obsahovat název aktéra. Aktéry lze taky rozdělovat na primární a pomocné. [34]



Obrázek 4 – Aktér [34]

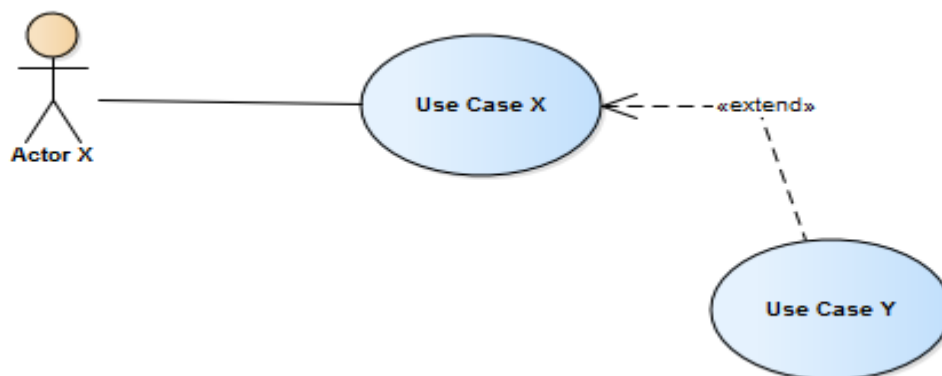
Případ užití je UML diagramech znázorněn jako elipsa s jeho názvem uvnitř. Při tvorbě rozsáhlejších a komplikovaných systémů se pro větší přehlednost a rychlou orientaci doporučuje přikládat k názvu případu jednoznačné identifikační číslo. Ve většině případů se používá UC001, UC002 a tak dále. Případem může být například registrace uživatele, stažení dokumentů nebo přidání nového komentáře. V případě definujeme funkcionalitu, kterou by měl náš navrhovaný systém umět. Popisujeme chování systému tak, jak je viděno z pohledu aktéra. Tento diagram vzniká ze zadání systému od našeho klienta nebo pokud vytváříme systém pro sebe, tak z našich představ, co by měl systém umět. Mluvíme tedy o takzvaném mapování uživatelských požadavků na konkrétní případy užití. [34]



Obrázek 5 – Příklad užití [34]

Vztahy jsou důležitou součástí každého UC diagramu. Aktéři a případy užití mezi sebou komunikují, když jsou mezi nimi vyměňovány informace. Vztahy jsou znázorněny plnými čarami mezi případy užití a aktéry. V případě užití používáme 3 typy vztahů. První z nich je vztah include. Relace include ve významu zahrnuje daný případ užití. Používá se uvnitř systému, kdy se jeden případ užití používá i v rámci jiného. Druhým vztahem je vztah extend. Relace extend ve významu rozšiřuje případ užití. Používá se v případě, že jeden proces může být rozšířen o jiný. Tyto dvě relace se v diagramu značí čárkovanou šipkou s nápisem

include nebo extend. Posledním vztahem je generalization neboli dědičnost. Ta vyjadřuje dědičnost mezi jednotlivými objekty, tedy že jeden případ užití může být specifikován jiným případem. Relace dědičnosti je v diagramu zobrazována jako vyplněná šipka. [34] [35]



Obrázek 6 – Vztah extend [35]

2.4.1.2 Use case Specifikace (specifikace případu užití)

Případ užití nám zobrazuje, jaké funkcionality náš systém zahrnuje a jak a kým jsou zahajovány. Ale kromě pojmenování individuálních případů užití o nich nevíme absolutně nic. Tuto záležitost řeší Use Case Specifikace. Jedná se o doplňující text, který je k diagramu případu užití doplněn. Tato specifikace nemá žádnou striktně danou formu, může být v podobě jednoduchého textu nebo tabulky a bývá dokládána ke smlouvě mezi zákazníkem a vývojovým týmem. Cílem specifikace je přesně definovat požadavky na systém a určit, co bude jednotlivý případ užití dělat, tedy jaké funkcionality bude zahrnovat. Specifikace jednotlivého případu užití by měla určit šest základních bodů. V prvním bodě je krátce popsán náš případ užití. Tento popis má za úkol přesně vysvětlit funkčnost tohoto případu, tedy například případ užití umožňuje registraci do systému. V druhé části jsou vypsáni všichni aktéři, kteří se konkrétního případu účastní. Příkladem této části může být uživatel a administrátor. Ve třetím bodě jsou definované podmínky pro spuštění. Každý z jednotlivých případů může, ale i nemusí mít definované konkrétní podmínky, které musí být splněny, aby mohl být jednotlivý případ rozběhnut. Například se může jednat o nutnost mít internetové připojení. Ve čtvrtém bodě je popsán základní tok interakcí mezi případy užití a jednotlivými aktéry. Jednotlivé body této části zapisujeme jako scénář. V páté části specifikace jsou vypsány alternativní toky, které budou spuštěny, pokud dojde odchylce nebo chybě v hlavním scénáři. Poslední části jsou vypsány podmínky, kterých musíme dosáhnout, abychom mohli

daný případ užití dokončit. Například, že uživatel bude po vyplnění registračního formuláře zaregistrován do systému. [36]

2.4.2 Class diagram (diagram tříd)

Jeho účelem je navrhnout nám statický pohled na modelovaný systém, přesně zobrazit typy jednotlivých objektů, které budou implementovány do systému a také popsat vztahy mezi těmito objekty. Objektem se v tomto ohledu rozumí sdružením dat a funkcionalit, které jsou spolu spojeny za účelem realizace požadované odpovědnosti. V tomto diagramu nalezneme jednotlivé třídy, kterým náleží jejich atributy a operace. Třídy jsou mezi sebou propojeny vazbami a násobností neboli multiplicitou. [37]

2.4.2.1 Třídy

Třída je popis množiny objektů, které mají shodnou strukturu, chování, vztahy a sémantiku. Každá třída musí být jednoznačně určená svým názvem. V jednotlivých třídách definujeme její atributy a operace. Názvy těchto atributů a operací musí jednoznačně určitelné. Pro vizuální zobrazení třídy se používá obdélník, který je rozdělen na 3 části. V první části se nachází název třídy. Ve druhé části nalezneme atributy a v poslední části jsou zapsány operace. [38]

2.4.2.2 Atributy

Atributy neboli vlastnosti třídy, slouží jakožto nositelé informací o dané třídě. Každému atributu je nutné přidělit jeho jméno, typ a viditelnost. Viditelnost atributů má za úkol řídit přístup jednotlivých členů třídy. Existují čtyři základní typy viditelnosti:

- Veřejný (Public) atribut je viditelný pro všechny v rámci celého systému. Veřejná viditelnost se označuje symbolem „+“.
- Soukromý (Private) atribut je viditelný pouze pro členy stejné třídy. Také ho mohou využívat pouze operace dané třídy. Soukromá viditelnost se označuje symbolem „-“.
- Chráněný (Protected) atribut je viditelný v rámci téže třídy a všech potomků této třídy. Chráněná viditelnost se označuje symbolem „#“.
- Balíček (Package) atribut je viditelný jenom v rámci toho balíčku tříd, ve kterém se nachází. Balíček viditelnosti je označován symbolem „~“.

Typ atributů nebo formát má za úkol definovat, v jakém formátu se bude daný atribut zapisovat. Mezi typy formátů patří například Integer (celé číslo), Float (reálné číslo), String (posloupnost znaků) nebo Varchar (textové datové pole) a mnoho dalších. Každý tento formát musí mít přesně definovaný maximální počet jednotlivých znaků. [38]

2.4.2.3 Operace

Operace jsou vlastně funkce jednotlivých tříd, které jsou s těmito třídami propojeny. Každé operaci náleží její vlastní charakteristika. Znamená to, že ji musíme popsat názvem, jejími argumenty a typem návratové hodnoty. U argumentů můžeme určit jejich směr, například „in“ je výchozí hodnota. Tento argument je použit jako vstupní. Nebo „out“, tento argument je použit jako výstupní. Argumentům operací lze přidělit takzvaná implicitní hodnota. To znamená, že když v okamžiku, kdy bude operace zavolána argument nebude mít hodnotu, tak použije hodnotu implicitní. K operacím přikládáme taky viditelnost. [38]

2.4.2.4 Vztahy mezi třídami

2.4.2.5 Asociace

Jedná se o základní vztah mezi třídami. Asociace má za úkol znázornit vztah mezi jednou nebo více třídami, mohou být jednosměrné nebo obousměrné. Značíme ji jako jednoduchou čáru. Tento vztah definuje, v jakém vzájemném vztahu se nachází propojené třídy. Smyslem asociace je znázornit důvod, proč o sobě musí propojené třídy vědět. [38] [39]

2.4.2.6 Agregace

Agregace je speciální druh agregace, který znázorňuje hierarchický vztah, kdy jeden z objektů je ten hlavní a ten se pak skládá z jiných objektů. Zakresluje ji jako jednoduchou čáru, která je z jedné strany zakončena prázdným kosočtvercem. Jako příklad můžeme uvést počítač, ten je ten hlavní celek. Počítač se pak skládá z jiných objektů, tedy komponentů. [38] [39]

2.4.2.7 Kompozice

Kompozice je speciálním typem agregace. U agregace může komponent existovat i bez celku, ale u kompozice nikoliv, tedy v tomto vztahu nemůže samostatně existovat podřízený objekt bez nadřízeného. Kompozici vyznačujeme podobně jako agregaci s tím rozdílem, že kosočtverec je vyplnění. [38] [39]

2.4.2.8 Generalizace

Generalizace neboli dědičnost je hierarchický vztah mezi třídami. Dědičnost umožňuje třídám dědit jejich vlastnosti a charakteristiku. Také zachovává rozdíly mezi podřízenými a nadřízenými třídami. Dědičností se popisuje vztah mezi třídami a jejich potomky. Nadřazené třídy dědí ze svých podřízených tříd jejich operace a atributy. Zakresluje ji pomocí čáry, která má na jednom ze svých konců šipku (trojúhelník). Tento trojúhelník je na straně entity, která je nadřazená. Jako příklad můžeme uvést samy sebe. Lidé jsou podtřídou opic. Opice jsou zase podtřídou savců. To znamená, že všechny opice jsou savci, tudíž přebírají jejich vlastnosti. Opice mají své vlastní atributy, ale ty základní dědí od savců. [38] [39]

2.4.2.9 Násobnost

Multiplicita neboli násobnost nám dává jasný popis omezení, kterých se musíme držet. Můžeme ji použít u vazeb asociace, agregace a kompozice. Při návrhu je potřeba popsat mohutnost jednotlivých vztahů. Násobnost definuje, kolik instancí jedné třídy může být spojeno s instancí druhé třídy. Máme několik druhů násobnosti:

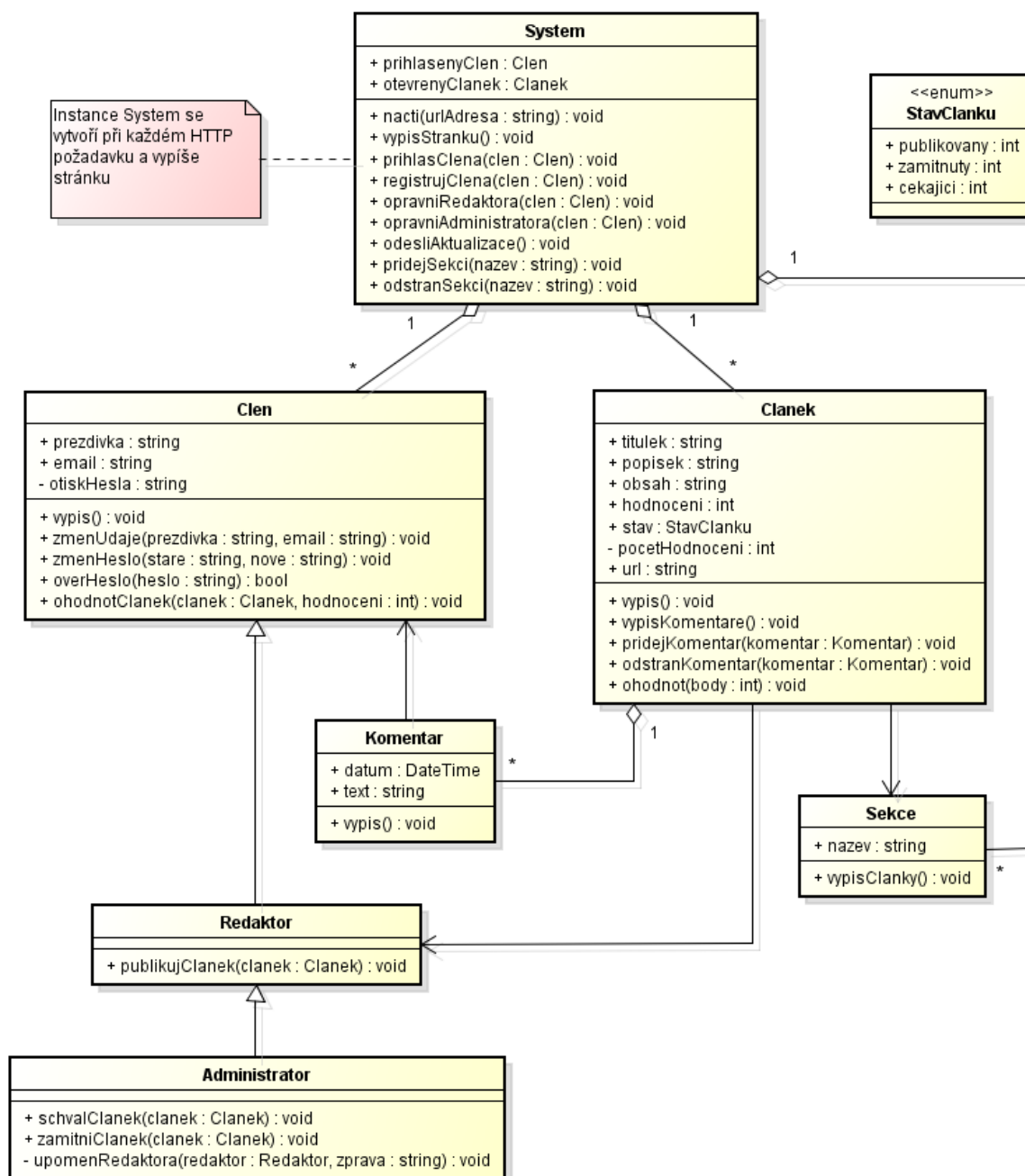
0 – nulu používáme pro to, kde nemůže být žádná instance.

0..1 – interval od nuly do jedné znázorňuje žádnou nebo jednu instanci.

1 – jednička popisuje pouze jednu instanci.

0..* – znázorňuje žádnou nebo mnoho instancí.

1..* – určuje jednu až více instancí. [38] [39]



Obrázek 7 – Diagram tříd [40]

2.5 JavaScript

JavaScript je skriptovací jazyk, který je určený pro vytváření pokrokových dynamických stránek a aplikací. Začal se používat v 90 letech 20. století. Vznikl jako odpověď na klasické statické weby a zpřístupnil možnost vytvářet interaktivní webové stránky i mobilní aplikace, které zlepšují návštěvníkovi zážitek z prohlížení webu.

Programy a kódy, které jsou napsané v JavaScriptu se jmenují skripty. Tyto skripty se vkládají přímo do html kódu, proto je poměrně jednoduchý, oproti jiným programovacím jazykům a mohou s ním pracovat i nováčci.

Dříve měl JS problém s nedostatečnou kompatibilitou ve webových prohlížečích. To mělo za následek, že uživatelům, kteří používali webový prohlížeč, který nepodporoval JS, se navštívené stránky neukazovali dobře. V současnosti je JS nedílnou součástí téměř všech webových stránek a jeho obliba čím dál více roste. Hlavním důvodem je to, že byly vytvořeny frameworky podporující vývoj webů. Mezi ty nejznámější patří například React nebo Angular.

V moderní době chytrých telefonů se JavaScript podílí i na budování mobilních aplikací. Za pomoci technologie React Native se vyvíjí aplikace, které mohou být kompatibilní pro více mobilních operačních systémů. Většinou se jedná o platformy Android a iOS.

JavaScript funguje ve webovém prohlížeči uživatele. Uživatel pomocí prohlížeče pošle požadavek na server stránek, stránky tento požadavek zpracují a odešlou zpět HTML se skriptem. HTML kód následně zpracuje prohlížeč a vydá uživateli obsah stránek. To znamená, že JS není umístěn na serveru. To má za důsledek, že stránky a aplikace jsou rychlejší a nezasekávají se, protože jejich server není zbytečně zatěžován. [41] [42]

2.5.1 Výhody a nevýhody

Největší předností je rychlost načítání stránek a jejich nezatěžování. Další výhodou je plná kompatibilita s HTML a CSS kódem. Díky tomu, že zdrojový kód je zpracován na straně uživatele, lze například změnit obsah zobrazovaný v prohlížeči bez toho, abychom museli znovu načíst navštěvované stránky.

Jako názornou ukázkou si můžeme uvést to, co všichni denně používáme. Máme vyhledávače jako je například Google, Yahoo nebo český Seznam. Když budeme chtít něco vyhledat na Googlu, začneme do pole psát to, co chceme najít. Po pár napsaných znacích nám vyhledávač sám nabídne ty nejvíce používané fráze. Tuto funkci zprostředkovává právě JavaScript a je hojně využívána po celém internetu.

Do nevýhod lze zařadit možné zneužití kódu, protože se zobrazuje v prohlížeči a je dostupný pro všechny uživatele. Dříve zde bylo velké riziko nepodporování JS prohlížeči, to je dneska již výjimkou. Dnes je jedná spíše o zákaz skriptů přímo v prohlížeči. Pokud bude JS zakázán

nebudou se uživatelé správně zobrazovat stránky. V JavaScriptu chybí možnost přístupu a úpravě všech dat kromě cookies. [41]

2.5.2 Využití

Existuje plno možností, jak lze JS využít. Například se může jednat o různé formuláře jako ověřovací element nebo automatické doplňování obsahu. Objevuje se při asynchronní spojení, při kterém není nutné zapojit přímo uživatele, jako je například již zmíněné vyhledávání. K tomu se vztahují i různé předem definované funkce. Například animace kurzoru myši po přejetí určitého bodu na webových stránkách.

Využívá se pro tvorbu hybridních mobilních aplikací a pro vytváření her, které běží v prohlížeči. Je velmi účinným prostředkem pro tvorbu responzivního vzhledu webů a zjednodušením při vytváření dnes velmi oblíbených sociálních sítí nebo CMS. [41] [45]

2.6 Bootstrap

Bootstrap je open-source front-end framework, který je určený k vytváření webů a webových aplikací. Tento framework je postaven na HTML, CSS a může obsahovat i JS. Obsahuje šablony určené k úpravě typografie, navigace, tlačítek, formulářů a mnoho dalších komponentů určených k tvorbě designu webu. Pomocí této knihovny se vytváří responsivní weby, to znamená že, obsah a vzhled webu se upraví podle toho, na jakém zařízení web prohlížíme. Weby vytvořené s touto knihovnou můžeme prohlížet například na počítači nebo na mobilním telefonu.

Pro práci s Bootstrapem je důležité umět alespoň základní syntaxe HTML a CSS. Díky velmi jednoduché implementaci a celkové jednoduchosti je tento framework vhodný nejen pro profesionály, ale i pro úplné nováčky. Každému z nich velice ulehčí práci při komplementaci webu.

Velkou výhodou této knihovny je jednoduché zhotovení uživatelského rozhraní aplikace nebo stránek, a to jak front-endu tak i back-endu. Bootstrap se stal v poslední době hodně používaný, a proto je plně kompatibilní s nejnovějšími verzemi téměř všech webových prohlížečů.

Disponuje veřejnou licencí, tudíž ho může používat úplně kdokoli a nemusí za nic platit. Bootstrap je k dispozici na GitHubu. Jeho dokumentace je přeložena do mnoha světových jazyků, bohužel český překlad stále chybí. [43] [44]

3 ZEBEZPEČENÍ A HROZBY ÚTOKŮ NA WEBOVOU APLIKACI

Zabezpečení by v dnešní době mělo být nedílnou součástí každé webové aplikace, bohužel existuje velké procento stránek, které se touto problematikou vůbec nezaobírají. Zabezpečení chrání jak uživatele webových stránek, tak i samostatné weby například před odcizením citlivých dat jakou jsou třeba platební karty, proto by mělo být vždy na prvním místě.

3.1 HTTPS

Úplným základem zabezpečení je HTTPS. Měla by to být první věc, kterou budete řešit, když zabezpečujete váš web. HTTPS je druh webového protokolu HTTP. Jedná se o způsoby, jakými jsou přenášena data po internetu. Tato data je nutné zabezpečit, protože když budou putovat internetem nezabezpečené, po cestě je může kdokoliv upravit nebo zmanipulovat. U klasického HTTP není přenos dat zabezpečen před odposloucháváním nebo stopováním. Oproti tomu u HTTPS jsou data šifrována, a to pomocí algoritmů SSL a TLS, které mají za úkol zašifrovat všechna přenášená data. Především se jedná o citlivá data jako je například číslo kreditní karty. Po nástupu nové směrnice Evropské unie o GDPR je tento standart téměř nevyhnutelný. Nezabezpečené stránky tímto protokolem lze jednoduše poznat. Stačí se podívat na URL adresu. [46]

3.1.1 Jak se bránit

Pokud se rozhodnete přejít na šifrování HTTPS, budete k tomu potřebovat certifikát, který je ve většině případů veřejně dostupný a jeho základní verze zdarma. Certifikát je určený k prokázání a ověření identity webových stránek. Pokud vlastníte certifikát, tak se v prohlížeči u URL vaší stránky bude zobrazovat, že je zabezpečená.

Po získání certifikátu si musíte se společností, u které máte hosting domluvit, že budete přecházet na HTTPS. Je zapotřebí, aby vám vložili a naistalovali SSL certifikát.

Následně změníte výchozí adresu webu z HTTP na HTTPS. Poté musíte stejným způsobem změnit všechny URL adresy, které na webu máte. Dále zabezpečte všechny externí soubory, jako jsou například obrázky nebo skripty, aby se načítaly z převedených URL adres.

Nutné je zajištění trvalého přesměrování pomocí stavového kódu 301. Tím docílíte toho, že veškeré adresy starého protokolu budou přesměrovány na nový. Důležité je vyměnit v Google Analytics starou adresu za novou. Vyhledávač si ji pak vloží do databáze a bude ji nabízet svým uživatelům. [47]

3.2 SQL injection

Jedná se o typ útoku na databázi systému, přes nezabezpečený vstup. Pokud se takový útok podaří, tak se útočník může dostat ke všem citlivým datům, které se v ní nacházejí. Také může databázi kompletně vymazat nebo upravit její obsah pro své účely. [48]

3.2.1 Zabezpečení proti SQL injection

Dříve se zabezpečovalo pomocí ošetřování proměnných (escapování). To ale v některých případech nestačilo. Nejlepší možností, jak se chránit pro tomto útku, je nekládání proměnných do dotazů a využívat takzvané parametrizované dotazy (Prepared Statements). Hlavní problém je totiž v tom, že zapisujeme proměnné přímo do SQL.

Do SQL kódu místo normálních hodnot vepíšeme náhradní znaky. Nastane situace, kdy budou dotazy odděleny od hodnot. Databáze si je pak sama automaticky převezme a zapíše tak, aby bylo to bylo bezpečné. Můžeme být v klidu, protože samotný systém databáze chyby nedělá. Pokud provedeme tuto situaci nevystavujeme se žádnému riziku.

Nesmíme zapomínat i na ochranu samotné databáze. Je důležité, aby k ní měli přístup jen vybraní uživatelé. Pokud by se útočník dostal k samotné databázi, tak do ní může vkládat SQL příkazy. Do nezbytně nutných oprávnění patří dotazy pro úpravu tabulek a zobrazování celé struktury databáze. [48]

3.2.2 Ukázka útoku

SQL injection lze rozdělit do čtyř základní druhů útoků.

1. Útok, který používá propojení více tabulek v databázi

V tomto případě využívá útočník toho, že výsledky zadaných dotazů se vrací přímo do prezentční vrstvy databáze. Škoditel používá operátor „UNION“, který zpřístupňuje používání více dotazů „SELECT“ do jediného výsledku.

Jako vzor nám poslouží e-shop s elektronikou. Pokud najedeme v e-shopu na kategorii slevy aplikace odešle URL dotaz do databáze. Když útočník toto URL upraví za pomoci již zmíněného operátoru „UNION“, tak upravené URL může vypadat například takto:

```
https://e-shop.cz/products?category=Slevy' UNION SELECT username, password FROM users--
```

Obrázek 8 – Ukázka útoku pomoci SQL příklad č.1 [48]

V případě že takto upraví URL v SQL se spustí dotaz:

```
SELECT *  
FROM products  
WHERE category = 'Slevy'  
UNION SELECT username, password  
FROM users --
```

Obrázek 9 – Ukázka útoku pomocí SQL příklad č.1 obr. 2 [47]

Jako výsledek tohoto dotazu se nám zobrazí všechny slevněné produkty co nalezneme v této webové aplikaci, ale i všechna uživatelská jména a hesla. [48]

2. Útočník obchází aplikační logiku

Jako příklad si uvedeme přihlášení do aplikace. Máme pouze jednoduchou tabulku pro přihlášení, která není nijak zabezpečená. Po stisknutí tlačítka přihlásit se, se do databáze odešle dotaz, který může útočník jednoduše upravit. Pouze k němu přidá apostrof a znak „-“, který vloží do místa pro přihlašovací jméno. Takto upravený dotaz může vypadat následovně:

```
SELECT `id`  
FROM `user`  
WHERE `name` = '' OR admin = 1-- AND `password` = '19c44a96a09dc0088f88d...'  
LIMIT 1
```

Obrázek 10 – Ukázka útoku pomocí SQL příklad č.2 [48]

Díky vloženému apostrofu se při zpracování takového dotazu ukončí řetězec u sloupce pro jméno a za tenhle řetězec se vloží podmínka „OR admin = 1“. Díky tomu, že dotaz obsahuje dvě pomlčky, tak je vnímán jako komentář. Takhle upravený dotaz provede pouze jedinou věc a to takovou, že zobrazí ID z tabulky uživatelů, kde platí, že jméno se rovná nenaplněnému řetězci nebo admin se rovná jedné. Pokud nastane tato situace, tak systém vybere z databáze administrátora, kterého aplikace přihlásí. Útočník tedy získá přístup pod administrátorem. [48]

3. Pozměnění databáze

Tento typ útoku využívá další základní dotazy SQL jako jsou například „UPDATE“ nebo „DELETE“. Náznornou ukázkou si můžeme opět předvést na přihlašovacím formuláři. Útočník může vložit další záznam do tabulky za pomoci příkazu „INSERT INTO“, nebo příkazem „UPDATE“ změnit obsah databáze. Například může odebrat uživatele, vložit nového uživatele nebo změnit přístupová hesla. [48]

```
' ; UPDATE user SET password='noveHeslo' WHERE password='stareHeslo'; --
```

Obrázek 11 – Ukázka útoku pomocí SQL příklad č.3 [48]

4. Získání struktury databáze

K předchozím případům je zapotřebí, aby útočník znal název databáze, sloupce nebo tabulky. Bohužel většina běžných webů používá základní pojmenování tabulek jako je například „user“. To znamená, že i bez znalostí struktury databáze má útočník velkou šanci na odhalení těchto názvů. Navíc lze využít tento typ útoku. Provedením získá obecné informace o struktuře databáze webu.

Existuje několik druhů útoků. Ty se vždy liší podle typu používané technologie databáze. Pro ukázkou si vybereme například MySQL. Opět se budeme zabývat přihlašovacím formulářem, do kterého útočník vloží dotaz „UNION SELECT @@version--,..“. Tím zjistí, jakou verzi databáze aplikace používá. Poté už je maličností dohledat dotazy, které popisují tabulky, většinou se jedná o příkaz „SELECT“. [48]

```
SELECT `id`  
FROM `user`  
WHERE `name` = '' UNION SELECT @@version-- AND `password` = '19c44a96a09dc0088f88d...'  
LIMIT 1
```

Obrázek 11 – Ukázka útoku pomocí SQL příklad č.4 [48]

3.3 Cross site scripting (XSS)

Jedná se o jednu z nejčastějších chyb v zabezpečení webové aplikace. Může nastat v momentu, kdy z aplikace proudí nezašifrovaná a neověřená data do webového prohlížeče uživatele. Tato situace poskytne útočníkovi příležitost spustit nežádoucí skripty v prohlížeči, zobrazovat uživatelské relace, zaměnit webové stránky za stránky útočnickovy, nebo může dojít až k phishingovým nebo malwarovým útokům. Takové napadení webu obvykle

probíhá pomocí Javascriptu, který dává útočníkovi příležitost upravovat veškerý obsah webové aplikace.

Jedná se tedy o cílený útok na uživatele webové aplikace. Pokud útočník prolomí web a vloží do něj škodlivý skript může díky němu například vyměnit přihlašovací formulář tak, aby když jej návštěvník vyplní a potvrdí, tak vyplněná data putují rovnou k útočníkovi nebo může ukrást přímo uživateli cookies.

S takovými daty je pro útočníka jednoduché vydávat se za uživatele, kterému byly tato data odcizena. Uživatelem aplikace je i administrátor, který po přihlášení může například spravovat obsah webu nebo vyřizovat objednávky.

Jeden z nejznámějších útoků tohoto typu se stal na společnost PayPal. Při útoku se podařilo prolomit web a přesměrovat všechny uživatele na podvrženou stránku s varováním, že jejich účty jsou ohrožené. Uživatelé byli přesměrováni na phishingový server, kde byli vyzváni k vyplnění přihlašovacích údajů. Všechny tyto vyplněné údaje poté putovali přímo k útočníkům. [49] [50]

3.3.1 Jak chránit uživatele

Chránit uživatele lze například zainteresováním seznamu typu Whitelist. Ten se používá k ověřování všech příchozích a odchozích dat. Umožňuje zamítnutí všech dat, která nejsou upřesněna v tomto seznamu jako nesprávná.

Další možností, jak zabezpečit stránku před XSS je možnost odfiltrovat nežádoucí znaky z uživateli integrace při vyplňování dat. To znamená, že se převedou na HTML entity. K tomuto kroku jsou dostupné funkce. Například v jazyce PHP se jedná o funkci „htmlspecialchars“. Důležité je zakázat používání HTML příkazů ve vstupech běžných uživatelů, například v přihlašovacím formuláři.

K útokům tohoto typu jsou většinou využíváni tzv. boti neboli roboti, kteří jsou plně automatizováni. Z tohoto důvodu je doporučeno zainteresovat do webové aplikace ověření CAPTCHA. Nejlépe pak přímo kódu, který bude nucen uživatel vyplnit před potvrzením formuláře. Přítomnost ověření CAPTCHA těmto robotům znemožní odesílání dat do systému aplikace. [49] [50]

3.4 Útok hrubou silou neboli DDoS

Jedná se o jeden z nejvíce používaných útoků napříč celým internetem. Největší problém spočívá v tom, že je velmi levný. Obvykle stačí někomu zaplatit pouze pár dolarů a útok objednat na konkurenci vašeho webu nebo pouze z pomsty. Tento útok se dělí na dva druhy Dos a DDos.

DoS (Denial of Service) funguje tak, že vaši aplikaci v jeden moment zahltí obrovské množství návštěvníků nebo dotazů. To má za důsledek přetížení serveru a následnou nedostupnost pro uživatele.

DDoS (Distributed Denial of Service) je prakticky totožný s předchozím útokem. Jediný rozdíl spočívá v tom, že se ve stejný moment na vaši aplikaci útočí více zdrojů. Většinou se jedná o internetové roboty. V tomto případě je komplikovanější vyhledat zdroje útoku a zabránit mu. [51] [52]

3.4.1 Jak se bránit proti DDoS útoku

Ochrana před tímto útokem je velmi složitá a většina lidí si sama neporadí, proto musejí vyhledat odborníka s velmi dobrými znalostmi kybernetické bezpečnosti. Strach z něj mají i ty největší z největších, jako je například Google nebo Instagram.

Existuje pár preventivních doporučení, jak se dá před tímto útokem bránit. Mezi tyto doporučení například patří:

Vlastnit nejnovější verze všech používaných pluginů a bezpečnostních záplat.

Nejaktuálnější verze systému.

Aktivování filtrování UDP požadavků z internetu.

Vypnout veškeré služby které momentálně nevyužíváte.

Pořídít si záložní zdroje a ukládat zálohy. V případě útoku přepnout na tyto zdroje. Tím zamezíte výpadku provozu aplikace. [51] [52]

II. PRAKTICKÁ ČÁST

4 ANALÝZA POŽADAVKŮ A VYPRACOVÁNÍ DATABÁZE

V této části je popsána funkcionalita systému, tedy co musí aplikace pro hudební portál obsahovat, jak by měla fungovat, a hlavně jaké funkce musí obsahovat.

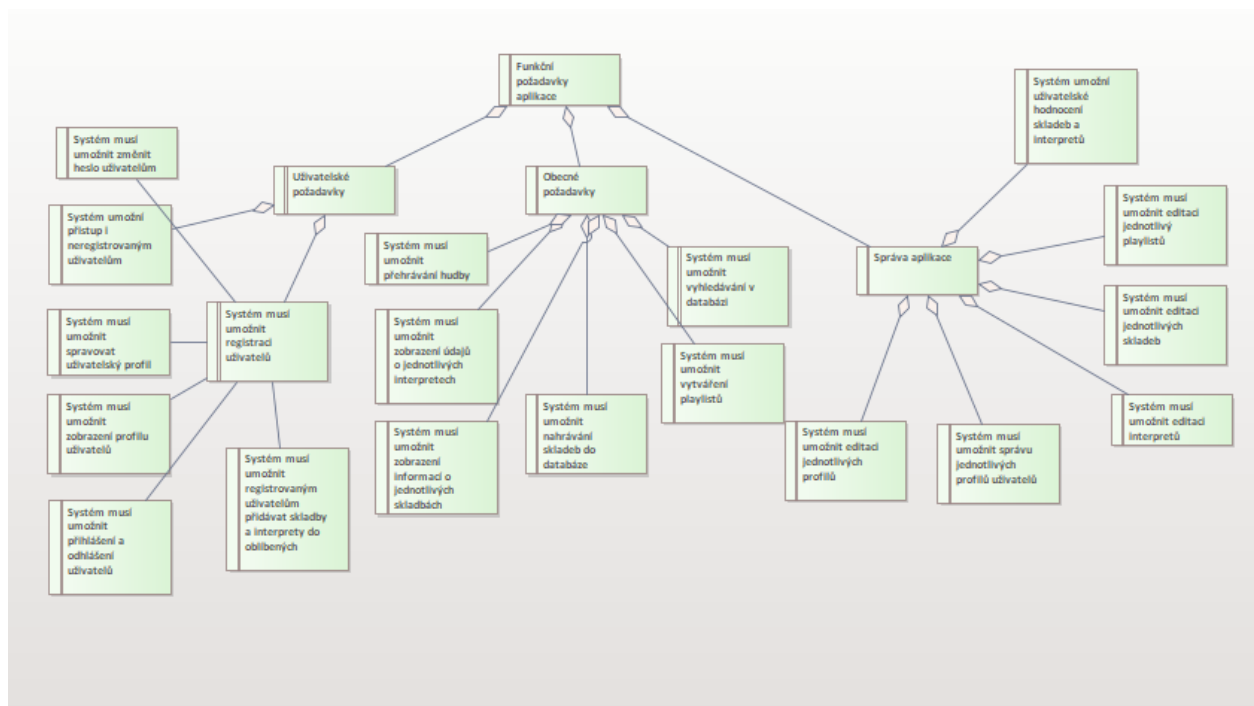
Požadavky jsou rozděleny do dvou kategorií. První z nich jsou funkční požadavky a druhé nefunkční požadavky. Funkční požadavky líčí, co bude vytvořený systém dělat. Nefunkční požadavky popisují, jak výsledná aplikace bude pracovat v kontextu použití. Jsou složeny z různých kritérií na danou aplikaci a mají za úkol popsat kvalitu aplikace. Mezi ty základní patří například výkon, spolehlivost, bezpečnost a provoz.

Funkční požadavky	Nefunkční požadavky
☒ + Funkční požadavky aplikace ☒ + Obecné požadavky ☒ + Správa aplikace ☒ + Systém musí umožnit editaci interpretů ☒ + Systém musí umožnit editaci jednotlivých playlistů ☒ + Systém musí umožnit editaci jednotlivých profilů ☒ + Systém musí umožnit editaci jednotlivých skladeb ☒ + Systém musí umožnit nahrávání skladeb do databáze ☒ + Systém musí umožnit přehrávání hudby ☒ + Systém musí umožnit přihlášení a odhlášení uživatelů ☒ + Systém musí umožnit registraci uživatelů ☒ + Systém musí umožnit registrovaným uživatelům přidávat skladby a interprety do oblíbených ☒ + Systém musí umožnit spravovat uživatelský profil ☒ + Systém musí umožnit správu jednotlivých profilů uživatelů ☒ + Systém musí umožnit vyhledávání v databázi ☒ + Systém musí umožnit vytváření playlistů ☒ + Systém musí umožnit změnit heslo uživatelům ☒ + Systém musí umožnit zobrazení informací o jednotlivých skladbách ☒ + Systém musí umožnit zobrazení profilu uživatelů ☒ + Systém musí umožnit zobrazení údajů o jednotlivých interpretech ☒ + Systém umožní přístup i neregistrovaným uživatelům ☒ + Systém umožní uživatelské hodnocení skladeb a interpretů ☒ + Uživatelské požadavky	☒ + Bezpečnost systému ☒ + Čas odezvy ☒ + Chránění dat ☒ + Dostupnost ☒ + Jednoduchá údržba ☒ + Jednoduchost systému ☒ + Kompatibilita ☒ + Paralelní připojení ☒ + Požadavky pro provoz ☒ + Přihlášení a odhlášení ze systému ☒ + Přihlášení pomocí hesla ☒ + Roziřitelnost systému ☒ + Rychlé aktualizace systému ☒ + Šifrování komunikace ☒ + Škálovatelnost ☒ + Spolehlivost ☒ + Správa obsahu ☒ + Správa uživatelských profilů ☒ + Uložistě ☒ + Výkon ☒ + Zabezpečení před napadnutím systému ☒ + Zálohování dat ☒ + Zálohování systému

Obrázek 12 – Funkční a nefunkční požadavky

4.1 Funkční požadavky

Funkční požadavky jsou rozděleny do tří základních kategorií. Obecné požadavky, uživatelské požadavky a požadavky pro správu aplikace.



Obrázek 13 – Funkční požadavky

4.1.1 Obecné funkční požadavky

V této kategorii jsou popsány ty nejzákladnější funkce aplikace. Tyto funkce jsou nepostradatelné pro chod aplikace, jako je například přehrávání hudby.

Hlavním požadavkem aplikace je přehrávání a správa hudby. Tudíž je nezbytné, aby bylo možné hudbu jak nahrávat, tak i přehrávat. K tomuto požadavku se váže i další nezbytná funkce, a to podrobný popis jak skladeb, tak i interpretů. Pro usnadnění hledání v databázi hudby je zapotřebí vyhledávání. Pro lepší manipulaci s obsahem aplikace je zapotřebí umožnit uživatelům vytvářet si vlastní playlisty oblíbené hudby, případně hudbu i hodnotit.

Systém musí umožnit přehrávání hudby

- Všechny skladby v hudební databázi bude možné přehrávat. Po přehrání jedné skladby bude automaticky zapnuta další skladba. Jednotlivé skladby půjde přeskokovat nebo posunout. Přehrávač bude obsahovat ovládání hlasitosti.
- U skladeb nalezneme hodnocení jednotlivých uživatelů a stručný popis.

Systém musí umožnit zobrazení informací o jednotlivých skladbách

- U každé ze skladeb budou k dispozici základní informace o dané skladbě.

- Mezi tyto základní informace bude patřit název jednotlivé skladby, rok, ve kterém byla vydána, jméno interpreta, který ji vydal, album, do kterého patří, délku skladby a žánr.

Systém musí umožnit zobrazení údajů o jednotlivých interpretech

- Každý z interpretů bude mít svůj vlastní profil, kde budou k nalezení základní údaje o něm a jeho tvorbě. K profilu bude přiřazena fotografie.
- Na tomto profilu nalezneme jméno a příjmení, jeho věk, umělecké jméno, žánr a seznam jeho vydaných skladeb.

Systém musí umožnit nahrávání skladeb do databáze

- Do hudební aplikace bude možné nahrát jakoukoliv hudbu z vašeho zařízení. Při procesu nahrávání bude po uživateli požadováno, aby vyplnil základní informace o dané hudbě. Skladba pak bude uložena do databáze hudby, kde bude k dispozici ostatním uživatelům systému. Z legislativních důvodů bude nahrávání skladeb omezeno pouze na administrátory aplikace. Uživatel bude mít možnost je kontaktovat a po předložení licence k této hudbě jim bude zaslán formulář, který vyplní a nahrají skladbu. Administrátor tento formulář schválí a následně bude skladba přidána do databáze.

Systém musí umožnit vytváření playlistů

- Každý z jednotlivých uživatelů bude mít možnost vytvořit si vlastní playlist jeho oblíbené hudby. Do playlistu bude moci přidávat jakékoliv skladby z celé nabídky databáze napříč všemi interprety.

Systém musí umožnit vyhledávání v databázi

- Pro jednodušší orientaci v aplikaci bude možné v aplikaci vyhledávat. Hledat půjde jak skladby, tak i alba nebo interprety.

4.1.2 Uživatelské požadavky

Systém musí umožnit registraci uživatelů

- Do systému se bude možné registrovat. Při registraci uživatel vyplní své základní údaje a nastaví si přihlašovací emailovou adresu a heslo.
- Uživatelské účty budou omezeny pouze na jednu emailovou adresu.

Systém umožní přístup i neregistrovaným uživatelům

- Aplikace bude dostupná i pro neregistrované uživatele. Neregistrovaní uživatelé budou mít omezené funkce. Nebudou moci hodnotit skladby ani si vytvářet vlastní playlisty. Nahrávání skladeb bude povoleno jen registrovaným uživatelům.
- Vytvoření uživatelského účtu bude zcela dobrovolné a nebude zapotřebí pro používání hudební aplikace.

Systém musí umožnit přihlášení a odhlášení uživatelů

- Uživatel aplikace bude mít možnost využívat aplikaci jak z pohledu registrovaného uživatele, tak i neregistrovaného.
- Pro přihlášení bude zapotřebí emailová adresa a heslo.
- Po odhlášení uživatele lze nadále využít aplikace, ale uživatelské osobní věci, jako například playlisty, zmizí.

Systém musí umožnit spravovat uživatelský profil

- Každý registrovaný uživatel aplikace bude mít svůj vlastní profil. Tento profil bude možné editovat.
- Uživatel bude moci změnit svůj profilový obrázek, heslo a přezdívku.
- Pokud si uživatel vytvoří svůj vlastní playlist, nalezne ho na svém profilu.

Systém musí umožnit zobrazení profilu uživatelů

- Registrovaní uživatelé budou mít možnost navštívit svůj vlastní profil. Na svém vlastním profilu naleznou své oblíbené skladby a interprety.

Systém musí umožnit změnit heslo uživatelům

- Každému registrovanému uživateli bude umožněno změnit si své heslo, případně pokud ho zapomene, tak ho obnovit.
- Při změně hesla bude po zapotřebí vyplnit heslo dvakrát z důvodu, aby nedošlo k chybě při vytváření nového hesla.
- Po změně hesla dojde na uživatelskou emailovou adresu potvrzovací kód.

Systém musí umožnit registrovaným uživatelům přidávat skladby a interprety do oblíbených

- Registrovaní uživatelé budou mít možnost přidat si svou oblíbenou skladbu nebo oblíbeného interpreta do svých oblíbených, ty budou pak k nalezení na profilu uživatele.

4.1.3 Správa aplikace**Systém musí umožnit editaci jednotlivých profilů**

- U každého profilu bude možné upravit uživatelské údaje, pouze emailová adresa, která byla vložena při registraci, nepůjde změnit.

Systém musí umožnit správu jednotlivých profilů uživatelů

- Jednotlivé profily bude možné editovat a odstraňovat. Odstraňování profilů bude povoleno jen vlastníkovvi profilu nebo administrátorovi aplikace.
- Pokud bude profil nevhodný nebo bude obsahovat některé nepovolené věci bude administrátorem smazán a uživateli s touto emailovou adresou bude odepřen přístup.

Systém musí umožnit editaci interpretů

- Každý z interpretů bude mít svůj vlastní profil na němž nalezneme základní informace o daném interpretovi, jeho skladby a hodnocení. Editovat interpreta bude možné pouze správcem aplikace.

Systém musí umožnit editaci jednotlivých skladeb

- Skladby, které budou nahrány do databáze aplikace bude možné dodatečně upravovat. Bude možné změnit jejich stručný popis nebo odstranit. Editace bude povolena jen uživateli, který danou skladbu přidal, a nebo administrátorovi aplikace.

Systém musí umožnit editaci jednotlivý playlistů

- Vytvořené playlisty půjdou dále upravovat. Uživatel bude moci změnit jejich jméno a přidávat nebo odstraňovat skladby.

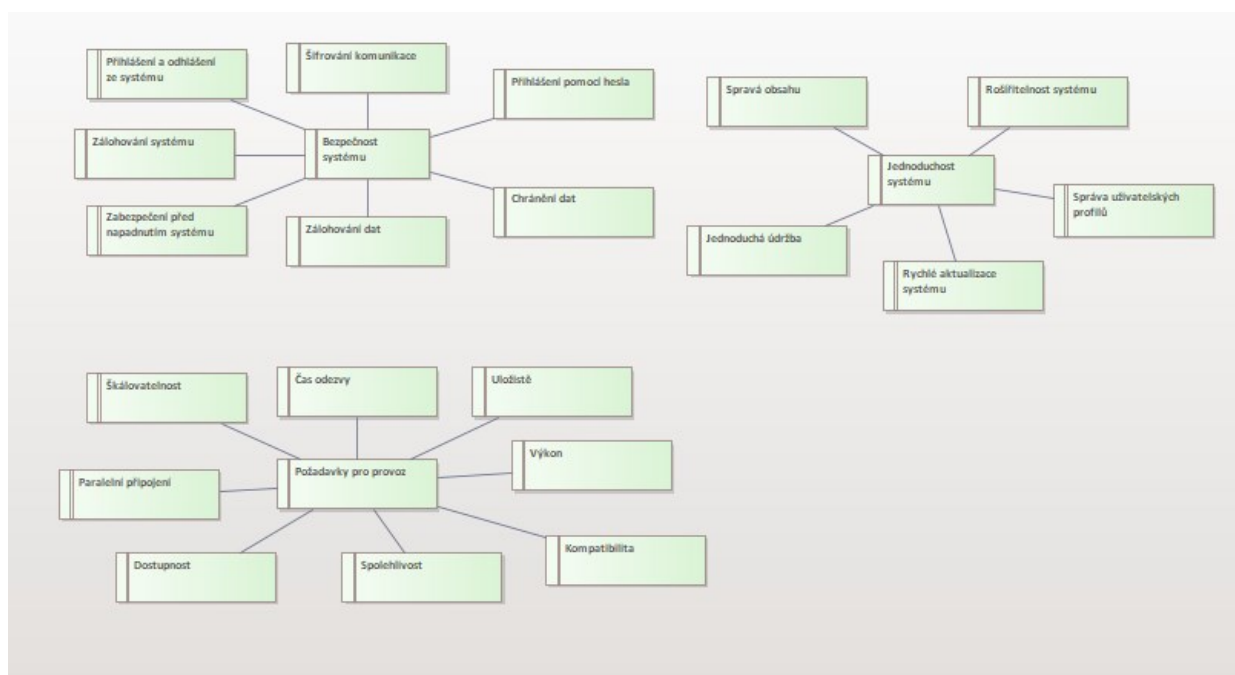
Systém umožní uživatelské hodnocení skladeb a interpretů

- Registrovaní uživatelé budou mít možnost hodnotit jak skladby, tak i interprety. V popisu skladeb a interpretů bude k nalezení toto hodnocení.

4.2 Nefunkční požadavky

Nefunkční požadavky jsou rozděleny na tři základní části – bezpečnost systému, jednoduchost systému a požadavky na provoz. Do těchto částí jsou vloženy nefunkční požadavky na systém. Aplikace musí splňovat mnoho nároků pro její hladké, a hlavně bezpečné fungování a provoz, také by měla splňovat legislativní a právní normy dané země ve které bude v provozu. Důležité je, aby byla kompatibilní pro většinu systémů a zařízení, které jsou na trhu, z důvodu dostupnosti pro co největší množství potencionálních uživatelů.

Bezpečnost systému má za účel co nejlépe ochránit aplikace před napadnutím nebo neoprávněným přístupem. Jednoduchost systému klade požadavky na co nejjednodušší správu aplikace. Požadavky na provoz jsou důležité pro hladký provoz a fungování celé aplikace.



Obrázek 14 – Nefunkční požadavky

4.2.1 Bezpečnost systému

Bezpečnost systému má za účelem co nejlépe ochránit aplikaci před napadnutím nebo neoprávněným přístupem. Také se zabývá komunikací mezi serverem a uživatelem a případnému zamezení ztráty dat.

Přihlášení a odhlášení ze systému

- Pro větší zabezpečení účtů bude zapotřebí pro přihlášení uživatelů heslo.

Šifrování komunikace

- Komunikace mezi aplikací a uživatelem bude šifrovaná, aby nemohlo docházet ke sledování komunikace uživatelů, a aby docházelo k zabezpečenému ukládání dat.

Přihlášení pomocí hesla

- Heslo musí splňovat určité požadavky na silné heslo pro zamezení prolomení a přístupu neoprávněných uživatelů.
- Požadavky na heslo jsou minimální délka znaků, musí obsahovat číslice a speciální znaky.

Zálohování systému

- Zálohování bude prováděno automaticky každý den v 1 hodinu v noci.
- Zálohovaná data budou ukládána mimo server pro případ, kdyby se porouchal, aby nedošlo ke ztrátě všech dat.

Chránění dat

- Všechna data na serveru musí být zabezpečena pro případ neoprávněnému přístupu. Přístup ke všem datům bude mít pouze administrátor aplikace.

Zabezpečení před napadnutím systému

- Aplikace musí být chráněna před všemi možnými způsoby napadnutí.

Zálohování dat

- Nová data přidaná uživateli budou automaticky každou hodinu uložena a zálohována.
- Budou vytvořeny kopie dat, které budou uloženy na bezpečné místo mimo server.

4.2.2 Jednoduchost systému

Systém musí být jednoduchý, přehledný a srozumitelný jak pro uživatele aplikace, tak i pro administrátory a správce serveru.

Správa obsahu

- Správa celého obsahu aplikace musí být jednoduchá a srozumitelná i pro úplné začátečníky. Přehlednost a jednoduchá orientace v aplikaci je prioritou.

Rozšiřitelnost systému

- Systém musí být snadno a jednoduše rozšiřitelný. Je potřeba poskytnout co největší uložení pro případ přidávání nových funkcionalit, bez toho, aniž by se ovlivnil provoz aplikace.

Jednoduchá údržba

- Údržby budou probíhat automaticky každý den v ranní hodinách, aby nebránili provozu aplikace. Údržba tak zabere co nejmenší čas a aplikace bude následně za co nejmenší dobu dostupná.

Rychlé aktualizace systému

- Aktualizace budou probíhat automaticky. Data se budou nahrávat po údržbě systému z důvodu, aby se co nejvíce zkrátila doba nedostupnosti aplikace.

Správa uživatelských profilů

- Administrátoři budou moci spravovat uživatele a jejich profily. Pokud to bude nutné, je možné uživatelům odebrat jejich profil a zablokovat je.

4.2.3 Požadavky pro provoz

Požadavky pro provoz jsou velmi důležitou součástí balíčku nefunkčních požadavků. Definiují celkovou výkonost aplikace.

Škálovatelnost

- Systému aplikace bude poskytnut požadovaný výkon i s rezervami, pro případ nečekaného náporu uživatelů.

Čas odezvy

- Pro co nejlepší uživatelský zážitek bude aplikaci poskytnut takový výkon, aby 90 % všech zpracovaných žádostí bylo vyřízeno do 3 sekund. Žádná ze žádostí nesmí přesáhnout dobu svého vyřízení 10 sekund.

Uložení

- Jedná se o hudební aplikaci, tudíž veškerá data budou zabírat hodně místa, proto je důležité poskytnout odpovídající uložení. Systém po zaplnění 70 % uložení upozorní administrátora a ten bude muset navýšit jeho kapacitu.

Paralelní připojení

- Systém bude škálovatelný pro sto uživatelů, pro případ, aby nedošlo k situaci, kdy bude uživatel nucen čekat, než se pro něj uvolní místo.

Výkon

- Systému musí být dostatečně výkonný. Zjištění potřebného výkonu bude probíhat v testovací fázi aplikace, kdy budou testéři simulovat běžné procesy v aplikaci. Vše se změří a odpovídající výkon i s rezervou se zajistí.

Dostupnost

- Mimo vymezené časy aktualizace a údržby musí být v aplikaci dostupné všechny její služby.

Spolehlivost

- Systém musí být zajištěn bezproblémový běh po celou dobu fungování aplikace, a to i při zvýšeném nárustu uživatelů.

Kompatibilita

- Aplikace musí být podporovaná pro všechny druhy zařízení od stolního počítače až po tablety. Potencionální uživatelé nesmějí být omezováni výběrem zařízení k navštívení aplikace.

4.3 Use Case model**4.3.1 Scénář UC01: Registrace uživatele**

Aplikace je přístupná i pro neregistrované uživatele, ale nemají k dispozici všechny její funkce. Pro zpřístupnění všech funkcí je zapotřebí registrace. Při vytváření uživatelského účtu bude po uživateli vyžadováno jméno, příjmení, pod kterou se budou přihlašovat do aplikace a email. V poslední řadě si každý uživatel nastaví heslo.

Tabulka 1 - UC01: Registrace uživatele

Jméno scénáře	UC01: Registrace uživatele
Zúčastnění aktéři	Neregistrovaný uživatel

Hlavní scénář	1. Uživatel si vyžádá registraci. 2. Přejde na stránku s registrací. 3. Vyplní jméno, příjmení, email a nastaví heslo. 4. Potvrdí registraci, čímž odešle formulář s registrací. 5. Systém ověří data. 6. Systém vloží data do databáze.
Vedlejší scénář	4.1 Uživatel špatně vyplnil data. Systém jej vyzve k opravě vložených údajů, čímž přechází do třetího kroku hlavního scénáře.
Podmínky pro dokončení	Vytvoří se uživatelský účet. Uživatel má následně možnost přihlásit se do aplikace.

4.3.2 Scénář UC02: Zobrazení registračního formuláře

Aplikace umožňuje registraci uživatelů, k tomu je zapotřebí registrační formulář. Registrační formulář se uživateli zobrazí po vyžádání registrace.

Tabulka 2 - UC02: Zobrazení registračního formuláře

Jméno scénáře	UC02: Zobrazení registračního formuláře
Zúčastnění aktéři	Neregistrovaný uživatel
Hlavní scénář	1. Uživatel si vyžádá registraci. 2. Aplikace přejde na stránku s registrací. 3. Vyčká na vyplnění dat od uživatele. 4. Po vyžádání registrace odešle data do systému. 5. Systém ověří data a odešle je do databáze. 6. Aplikace potvrdí registraci a přejde na stránku s přihlášením.

Vedlejší scénář	4.1 Systém při provádění validace dat zjistil chybu. Vyzve uživatele k nápravě, tedy ke třetímu kroku hlavního scénáře.
Podmínky pro dokončení	Správně vyplněná data od uživatele a vložení těchto dat do databáze.

4.3.3 Scénář UC03: Přihlášení

Pro přihlášení do aplikace bude zapotřebí registrace. Po registraci se uživateli zobrazí přihlašovací formulář. Do tohoto formuláře musí uživatel vložit své přihlašovací jméno a heslo. Pokud tato data zadá uživatel správně dojde k přihlášení. V opačném případě nedojde k přihlášení a systém upozorní uživatele na špatně vložená data.

Tabulka 3 - UC03: Přihlášení

Jméno scénáře	UC03: Přihlášení
Zúčastnění aktéři	Registrovaný uživatel
Hlavní scénář	1. Uživatel si vyžádá přihlášení. 2. Přejde na stránku s přihlášením. 3. Vyplní přihlašovací jméno a heslo. 4. Potvrdí přihlášení 5. Systém ověří data a přihlásí uživatele.
Vedlejší scénář	4.1 Systém při provádění validace dat zjistil chybu. Vyzve uživatele k nápravě, tedy ke třetímu kroku hlavního scénáře.
Podmínky pro dokončení	Vložené uživatelská data se shodují s daty v databázi systému.

4.3.4 Scénář UC04: Zobrazení přihlašovacího formuláře

Pro zpřístupnění všech funkcí aplikace je nutné se do aplikace přihlásit. K tomu je zapotřebí přihlašovací formulář. Ten se zobrazí po registraci nebo po vyžádání přihlášení od uživatele.

Tabulka 4 - UC04: Zobrazení přihlašovacího formuláře

Jméno scénáře	UC04: Zobrazení přihlašovacího formuláře
Zúčastnění aktéři	Registrovaný uživatel
Hlavní scénář	1. Uživatel si vyžádá přihlášení. 2. Aplikace přejde na stránku s přihlášením. 3. Vyčká na vyplnění dat od uživatele. 4. Po potvrzení přihlášení odešle data do systému. 5. Systém ověří data. 6. Aplikace přihlásí uživatele.
Vedlejší scénář	5.1 Systém při ověřování vyplněných dat zjistil chybu. Nepotvrdí přihlášení, ale vyzve uživatele, aby údaje zadal správně, tedy ke třetímu kroku hlavního scénáře.
Podmínky pro dokončení	Vyplněná data od uživatele se shodují s daty v databázi systému.

4.3.5 Scénář UC05: Změna uživatelských údajů

Uživatel bude mít možnost upravit si profil podle své libosti. V profilu uživatele nalezneme základní informace o uživateli. Bude možné přidat profilový obrázek. Na profilu také nalezneme seznam oblíbené hudby.

Tabulka 5 - UC05: Změna uživatelských údajů

Jméno scénáře	UC05: Změna uživatelských údajů
Zúčastnění aktéři	Uživatel
Hlavní scénář	1. Uživatel si zvolí profil. 2. Aplikace přejde na stránku s profilem. 3. Uživatel zvolí upravení profilu.

	4. Aplikace zobrazí stránku s formulářem pro editaci profilu. 5. Uživatel vyplní formulář. 6. Aplikace potvrdí editaci a odešle data do systému.
Vedlejší scénář	6.1 Uživatel při vyplňování dat udělá chybu. Aplikace ho na ni upozorní a uživatel přehází zpět ke kroku číslo 5 hlavního scénáře.
Podmínky pro dokončení	Uživatel vyplní správně data a systém je zaeviduje v databázi.

4.3.6 Scénář UC06: Změna hesla

Uživatel bude mít možnost změnit své heslo. Heslo musí splňovat jistá kritéria, aby bylo povoleno. Z důvodu zabezpečení musí mít minimální počet znaků, obsahovat velká a malá písmena a obsahovat minimálně jeden speciální znak.

Tabulka 6 - UC06: Změna hesla

Jméno scénáře	UC06: Změna hesla
Zúčastnění aktéři	Registrovaný uživatel
Hlavní scénář	1. Uživatel si najede na svůj profil. 2. Na svém profilu vybere změnu hesla. 3. Uživateli se zobrazí formulář pro změnu hesla. 4. Uživatel vyplní formulář a odešle ho. 5. Systém ověří vyplněná data a v databázi porovná staré heslo s heslem zadaným uživatelem, pokud se shodují odešle na email uživatel ověřovací kód. 6. Po zadání ověřovacího kódu do aplikace bude provedena změna hesla. 7. Systém změní uživateli heslo.

Vedlejší scénář	5.1 Při ověřování dat nastala chyba, buď to se heslo neshoduje s heslem v databázi nebo uživatel špatně vyplnil formulář. Pokud nastane tato událost uživatel bude přesunut k bodu číslo 4 hlavního scénáře.
Podmínky pro dokončení	Uživatel potvrdí nové heslo pomocí kódu, který nalezne na své emailové adrese.

4.3.7 Scénář UC07: Obnova zapomenutého hesla

Pokud uživatel zapomene přihlašovací údaje, může požádat o jejich obnovení. K tomu bude nutný email, který uživatel zadal při registraci. Po vyplnění bude na tento email zaslaný potřebný kód pro obnovu hesla. Kód pak předloží a po úspěšné validaci mu bude umožněno změnit si své heslo.

Tabulka 7 - UC07: Obnova zapomenutého hesla

Jméno scénáře	UC07: Obnova zapomenutého hesla
Zúčastnění aktéři	Registrovaný uživatel
Hlavní scénář	1. Uživatel požádá o obnovu hesla. 2. Systém ho přesměruje na formulář, do kterého vyplní svou emailovou adresu. 3. Systém zkontroluje, jestli se adresa shoduje s adresou v databázi. 4. Po úspěšné validaci odešle na zadaný email ověřovací kód. 5. Uživatel předloží ověřovací kód a bude mu umožněno zvolit si nové heslo. 6. Systém ověří nové heslo, zda splňuje všechna potřebná kritéria. 7. Nové heslo uloží do databáze.

Vedlejší scénář	1.1 Zadaná adresa se neshoduje s adresou v databázi. Systém na to upozorní uživatele, který bude přeměrován ke kroku číslo 2 hlavního scénáře. 1.2 Nové heslo nesplňuje všechna potřebná kritéria, tudíž nemůže být použito a uživatel bude vyzván k opravě, tudíž se přesune zpět k 5 bodu hlavního scénáře.
Podmínky pro dokončení	Uživatel předloží platnou emailovou adresu a ověřovací kód.

4.3.8 Scénář UC08: Přehrávání hudby

V aplikaci bude možné přehrávat hudbu všem návštěvníkům. Pro přehrávání bude stačit kliknout na skladbu, kterou bude chtít uživatel přehrát a ona se automaticky zapne, nebo může navštívit přímo přehrávač.

Tabulka 8 - UC08: Přehrávání hudby

Jméno scénáře	UC08: Přehrávání hudby
Zúčastnění aktéři	Uživatel
Hlavní scénář	1. Uživatel si z nabídky hudby vybere skladbu, kterou bude chtít přehrát a klikne na ni. 2. Systém načte požadovanou skladbu z databáze. 3. Aplikace automaticky začne přehrávat vybranou skladbu. 4. Pokud uživatel klikne znovu hudba se stopne. 5. Po druhém kliknutí aplikace přestane s přehráváním skladby.
Vedlejší scénář	1.1 Uživatel navštíví přehrávač, ve kterém mu bude umožněno poslouchat hudbu. Přehrávač bude vybaven mnoha funkcemi pro zpríjemnění uživatelského

	zážitku jako bude například náhodné přehrávání nebo změna hlasitosti.
Podmínky pro dokončení	Skladby budou načteny ze serverové knihovny.

4.3.9 Scénář UC09: Vyhledávání hudby

Pro snadnější orientaci v aplikaci bude umožněno vyhledávání. Vyhledávat půjde například skladby, interprety nebo hudební alba.

Tabulka 9 - UC09: Vyhledávání hudby

Jméno scénáře	UC09: Vyhledávání hudby
Zúčastnění aktéři	Uživatel
Hlavní scénář	1. Uživatel si na hlavní liště vybere stránku vyhledávání. 2. Aplikace ho přesměruje na vyhledavač. 3. Uživatel do vyhledávacího pole napíše, co přesně chce najít. 4. Aplikace vyhledá zadané údaje v databázi. 5. Pokud se vyhledávaná data nachází v databázi, tak je server zobrazí uživateli.
Vedlejší scénář	4.1 Pokud nejsou nalezena vyhledávaná data v databázi, tak se uživateli například zobrazí že danou skladbu bohužel nevlastníme v naší databázi a bude vyzvat ať zkusí štěstí s jinou, tudíž se vrátí k bodu číslo 3 hlavního scénáře.
Podmínky pro dokončení	Vyhledávaný obsah musí být k dispozici v databázi aplikace.

4.3.10 Scénář UC10: Zobrazení základních informací o skladbách

Každé skladbě budou přiděleny základní informace o dané skladbě jako je například kdo ji složil, jak se jmenuje a do jakého žánru patří. Uživatel bude mít možnost do těchto informací nahlédnout.

Tabulka 10 - UC10: Zobrazení základních informací o skladbách

Jméno scénáře	UC10: Zobrazení základních informací o skladbách
Zúčastnění aktéři	Uživatel
Hlavní scénář	1. Uživatel si z nabídky obsahu vybere skladby. 2. Aplikace ho přesune na stránku se skladbami. 3. Systém načte veškeré skladby z databáze. 4. Uživatel může prohlížet všechny skladby. Každá z nich bude obsahovat základní informace o ní.
Vedlejší scénář	2.1 Systému se nepodaří načíst skladby z databáze, tudíž je bude muset načíst opakovaně proto se vrátí zpět ke 3 kroku hlavního scénáře.
Podmínky pro dokončení	Obsah musí být načten z databáze aplikace.

4.3.11 Scénář UC11: Zobrazení základních informací o interpretech

V aplikaci budou dostupné základní informace o interpretech. Každý z nich bude mít svůj vlastní profil, kde nalezneme například jeho jméno, umělecké jméno, datum narození, dále v jakém žánru hudby se pohybuje a seznam jeho skladeb.

Tabulka 11 - UC11: Zobrazení základních informací o interpretech

Jméno scénáře	UC11: Zobrazení základních informací o interpretech
Zúčastnění aktéři	Uživatel

Hlavní scénář	1. Uživatel v aplikaci přejde na sekci interpreti. 2. Aplikace ho přesune na stránku s umělci. 3. Systém načte veškerá data z databáze. 4. Uživatel může prohlížet všechny interprety, pokud bude chtít o nějakém z nich vědět víc informací klikne na něj. Po kliknutí bude přesměrován na profil interpreta. 5. Aplikace přesune uživatele na jeho profil. 6. Uživatel si může procházet jeho profil a přehrávat umělcovi skladby.
Vedlejší scénář	2.1 Systému se nepodaří načíst data z databáze, tudíž je bude muset načíst opakovaně proto se vrátí zpět ke 3 kroku hlavního scénáře.
Podmínky pro dokončení	Obsah musí být načten z databáze aplikace.

4.3.12 Scénář UC12: Zobrazení základních informací o hudebních žánrech

V aplikaci budou k nalezení základní informace o hudebních žánrech. Každý z těchto žánrů bude mít svou vlastní stránku, kde nalezneme například umělce, kteří v tomto žánru skládají hudbu a nebo skladby jenž patří do tohoto odvětví.

Tabulka 12 - UC12: Zobrazení základních informací o hudebních žánrech

Jméno scénáře	UC12: Zobrazení základních informací o hudebních žánrech
Zúčastnění aktéři	Uživatel
Hlavní scénář	1. Uživatel v aplikaci přejde na sekci žánry. 2. Aplikace ho přesměruje na odpovídající stránku. 3. Systém načte veškerá data z databáze.

	4. Uživatel může prohlížet všechny dostupné žánry, pokud bude chtít o nějakém z nich vědět víc informací klikne na něj. Po kliknutí bude přesměrován na stránku konkrétního žánru. 5. Aplikace přesune uživatele na stránku konkrétního žánru. 6. Uživatel si může procházet dostupné informace a přehrávat skladby které spadají do tohoto odvětví.
Vedlejší scénář	2.1 Systému se nepodaří načíst data z databáze, tudíž je bude muset načíst opakovaně proto se vrátí zpět ke 3 kroku hlavního scénáře.
Podmínky pro dokončení	Obsah musí být načten z databáze aplikace.

4.3.13 Scénář UC13: Přidávání skladeb

Do aplikace bude umožněno nahrávat nové skladby. Z legislativních důvodů to bude umožněno pouze administrátorovi. Uživatelé budou mít taky možnost přidávat obsah, ale musí kontaktovat administrátora a doložit licenci k obsahu, který budou chtít přidat.

Tabulka 13 - UC13: Přidávání skladeb

Jméno scénáře	UC13: Přidávání skladeb
Zúčastnění aktéři	Uživatel, Administrátor
Hlavní scénář	1. Uživatel kontaktuje administrátora. Doloží veškeré potřebné licence. 2. Administrátor zkontroluje dodané licence. 3. Pokud je všechno v pořádku odešle uživateli formulář. 4. Uživatel vyplní formulář a přidá k němu skladbu kterou bude chtít vložit do aplikace.

	5. Administrátor zkontroluje vyplněný obsah. 6. Přidá obsah do databáze systému.
Vedlejší scénář	2.1 Dodané licence nejsou v pořádku, tudíž nemohou být použity a uživatel se přesouvá zpět k bodu číslo 1 hlavního scénáře.
Podmínky pro dokončení	Licence musí být v pořádku. Obsah musí být nahrán do databáze.

4.3.14 Scénář UC14: Vyhledávání obsahu podle potřeb uživatele

Uživatel bude mít možnost vyhledávat v aplikaci podle jeho potřeb. Vyhledávat půjde podle skladeb, interpretů a žánrů.

Tabulka 14 - UC14: Vyhledávání obsahu podle potřeb uživatele

Jméno scénáře	UC14: Vyhledávání obsahu podle potřeb uživatele
Zúčastnění aktéři	Uživatel
Hlavní scénář	1. Uživatel přejde do jedné z nabízených sekcí. 2. Jakmile bude v nabízené sekci tak přejde na vyhledávání. 3. Aplikace ho přesune na stránku s vyhledáváním. 4. Systém automaticky zpracuje že bude chtít vyhledávat podle toho z které části aplikace se přesunul. 5. Uživatel zadá jeho požadavek na vyhledávání. 6. Aplikace vyhledá zadané údaje v databázi. 7. Pokud se vyhledávaná data nachází v databázi, tak je server zobrazí uživateli.
Vedlejší scénář	6.1 Pokud nejsou nalezena vyhledávaná data v databázi, tak se uživateli například zobrazí že danou skladbu

	bohužel nevlastníme v naší databázi a bude vyzvat ať zkusí štěstí s jinou, tudíž se vrátí k bodu číslo 5 hlavního scénáře.
Podmínky pro dokončení	Vyhledávaný obsah musí být k dispozici v databázi aplikace.

4.3.15 Scénář UC15: Podle skladeb

Uživatel bude mít možnost vyhledat hudbu v aplikaci podle názvu skladby.

Tabulka 15 - UC15: Podle skladeb

Jméno scénáře	UC15: Podle skladeb
Zúčastnění aktéři	Uživatel
Hlavní scénář	1. Uživatel přejde na stránku se skladbami. 2. Ve skladbách se přesune na vyhledávání. 3. Vypíše požadavky na hledání 4. Systém vyhledá data v databázi. 5. Pokud se vyhledávaná data nachází v databázi, tak je server zobrazí uživateli.
Vedlejší scénář	3.1 Pokud nejsou nalezena vyhledávaná data v databázi uživatel musí pozměnit vyhledávaný obsah. V tomto případě bude pokračovat krokem číslo 3 hlavního scénáře.
Podmínky pro dokončení	Vyhledávaný obsah musí být k dispozici v databázi aplikace.

4.3.16 Scénář UC16: Podle interpretů

Uživateli bude umožněno vyhledat hudbu v aplikaci podle interpreta, který tuto skladbu složil.

Tabulka 16 - UC16: Podle interpretů

Jméno scénáře	UC16: Podle interpretů
Zúčastnění aktéři	Uživatel
Hlavní scénář	1. Uživatel přejde na stránku s interprety. 2. Na této stránce se přesune na vyhledávání. 3. Vypíše požadavky na hledání. 4. Systém vyhledá data v databázi. 5. Pokud se vyhledávaná data nachází v databázi, tak je server zobrazí uživateli.
Vedlejší scénář	3.1 Pokud nejsou nalezena vyhledávaná data v databázi uživatel musí pozměnit vyhledávaný obsah. V tomto případě bude pokračovat krokem číslo 3 hlavního scénáře.
Podmínky pro dokončení	Vyhledávaný obsah musí být k dispozici v databázi aplikace.

4.3.17 Scénář UC17: Podle žánrů

Uživateli bude nabídnuta možnost vyhledávat skladby podle žánrů do kterých daná hudba spadá.

Tabulka 17 - UC17: Podle žánrů

Jméno scénáře	UC17: Podle žánrů
Zúčastnění aktéři	Uživatel
Hlavní scénář	1. Uživatel přejde na stránku se žánry. 2. Na této stránce se přesune na vyhledávání. 3. Vypíše požadavky na hledání. 4. Systém vyhledá data v databázi.

	5. Pokud se vyhledávaná data nachází v databázi, tak je server zobrazí uživateli.
Vedlejší scénář	3.1 Pokud nejsou nalezena vyhledávaná data v databázi uživatel musí pozměnit vyhledávaný obsah. V tomto případě bude pokračovat krokem číslo 3 hlavního scénáře.
Podmínky pro dokončení	Vyhledávaný obsah musí být k dispozici v databázi aplikace.

4.3.18 Scénář UC18: Editace obsahu

V aplikaci bude administrátorům umožněno spravovat obsah dat v aplikaci.

Tabulka 18 - UC18: Editace obsahu

Jméno scénáře	UC18: Editace obsahu
Zúčastnění aktéři	Administrátor
Hlavní scénář	1. Administrátor provede úpravu dat v databázi aplikace. 2. Systém překontroluje, zda jsou tyto data upravené správně. 3. Systém odešle data do databáze a uloží je. 4. Data z databáze se nahrají do aplikace.
Vedlejší scénář	2.1 Pokud nebudou data upravena správně, systém nepovolí vložení těchto dat do databáze. Když nastane tento scénář tak případ užití bude pokračovat krokem číslo jedna hlavního scénáře.

Podmínky pro dokončení	Správná úprava dat a jejich uložení v databázi.
-------------------------------	---

4.3.19 Scénář UC19: Správa uživatelů

Administrátoři budou mít možnost spravovat uživatele. Pokud některý uživatel poruší pravidla používání aplikace, bude odebrán. Administrátor má také možnost editovat profily uživatelů.

Tabulka 19 - UC19: Správa uživatelů

Jméno scénáře	UC19: Správa uživatelů
Zúčastnění aktéři	Administrátor
Hlavní scénář	1. Administrátor odebere uživatele. 2. Systém vyšle příkaz k odebrání uživatele z databáze. 3. Databáze smaže záznam uživatele. 4. Uživatel bude odebrán z aplikace
Vedlejší scénář	3.1 Pokud nebude smazán záznam z databáze, tak nemůže být uživatel odebrán. Po uplatnění tohoto scénáře se vrátíme k bodu číslo 2 hlavního scénáře.
Podmínky pro dokončení	Data budou vymazána z databáze.

4.3.20 Scénář UC20: Přehled statistik

Administrátor bude mít možnost prohlédnout si statistiky aplikace. Například bude mít možnost sledovat návštěvnost webu nebo počet nově registrovaných uživatelů.

Tabulka 20 - UC20: Přehled statistik

Jméno scénáře	UC20: Přehled statistik
----------------------	-------------------------

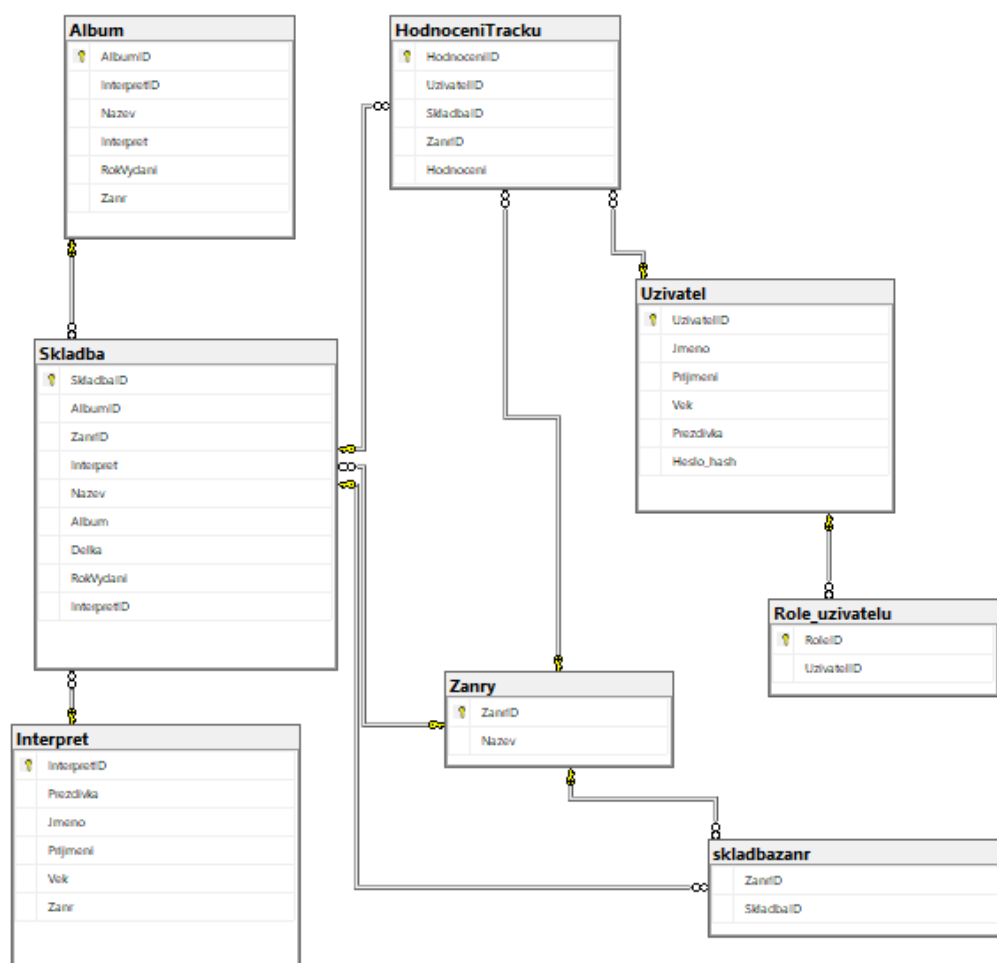
Zúčastnění aktéři	Administrátor
Hlavní scénář	1. Administrátor v administrátorské sekci aplikace vybere přehled statistik. 2. Systém vydá příkaz pro nahrání dat z databáze. 3. Databáze nahraje data do systému. 4. Systém zpracuje data a aplikace je zobrazí. 5. Administrátorovi se zobrazí statistiky.
Vedlejší scénář	3.1 Pokud se data správně nenahrají musí se znovu provést krok číslo 2 hlavního scénáře.
Podmínky pro dokončení	Systém správně nahraje data z databáze.

4.4 Databáze

Pro navržení a vypracování databáze jsem využil program Microsoft SQL Management Studio, ve které jsem za pomoci SQL vytvořil databázi spolu s relačním diagramem. V databázi jsem vytvořil tabulky spolu s jejími atributy. V následující části budou u každé tabulky popsány její základní informace spolu s datovými typy a SQL kódem.

4.4.1 Diagram databáze

Pro lepší znázornění vytvořené databáze jsem namodeloval diagram, který je srozumitelný a přehledný. Názorně zobrazuje, jaké tabulky se v databázi nacházejí. V tabulkách nalezneme jejich primární klíče a další hodnoty. Dále definuje vztahy mezi jednotlivými tabulkami. Důležitý je také popis primárních a cizích klíčů. Primární klíče patří vyhraně jedné tabulce, tudíž je jedinečný a podle něj můžeme jednoznačně určit, do které tabulky patří. Cizí klíče jednoznačně popisují řádky, které jsou vkládány do tabulky z jiné tabulky.



Obrázek 15 – Diagram databáze

4.4.2 Vytvoření databáze

Nejprve jsem musel vytvořit samotnou databázi. Za pomoci SQL příkazu:

```
CREATE DATABASE HUDEBNIAPLIKACE
```

Následně jsem do databáze vkládal tabulky spolu s jejími atributy.

4.4.2.1 Tabulka Uživatel

Do tabulky „Uživatel“ jsem vložil její primární klíč, které bude v každé tabulce znázorněn datovým typem int (integer), tedy budou popsány celými čísly. Dále jsem do tabulky přidal atributy potřebné pro uživatelský profil. Do těchto atributů patří jméno a příjmení uživatele, jeho věk a přezdívka. Důležitou součástí je taky uživatelské heslo. U jména, příjmení a přezdívky jsem zvolil datový typ „nvarchar“, na věk jsem použil „date“ a hesla „varchar(32)“ z důvodu zabezpečení hesla, které bude ukládáno pomocí hash funkce.


```
CREATE TABLE Uzivatel  
(  
    UzivatelID int NOT NULL PRIMARY KEY IDENTITY(0,1),  
    Jmeno nvarchar(30) NULL,  
    Prijmeni nvarchar(50) NULL,  
    Vek date NULL,  
    Prezdivka nvarchar(25) NOT NULL,  
    Heslo_hash varchar(32) NOT NULL,  
)
```

4.4.2.2 Tabulka Interpret

Tabulka „Interpret“ je důležitá pro získání základních informací o umělcích. V tabulce interpret nalezneme atributy přezdivka neboli jeho umělecké jméno, jméno a příjmení, věk a žánr, do kterého daný interpret spadá. U textových atributů jsem využil datový typ „nvarchar“, u věku „integer“.

```
CREATE TABLE Interpret  
(  
    InterpretID int NOT NULL PRIMARY KEY IDENTITY(0,1),  
    Prezdivka nvarchar(25) NOT NULL,  
    Jmeno nvarchar(30) NULL,  
    Prijmeni nvarchar(50) NULL,  
    Vek int NULL,  
    Zanr nvarchar(25) NULL,  
)
```

4.4.2.3 Tabulka Album

Tabulku „Album“ jsem vytvořil pro popsání základních informací o vydaných hudebních alb. V tabulce jsou jen ty nejzákladnější informace jako je název vydaného alba, interpret, který dané album vydal, rok vydání a žánr, do kterého spadá. Textové atributy jsou popsány datovým typem „nvarchar“. U roku vydání jsem použil „date“. Důležitou součástí této tabulky je cizí klíč. Ten je zde potřebný pro získání informací o interpretovi z tabulky interpret.

```
CREATE TABLE Album
(
AlbumID int NOT NULL PRIMARY KEY IDENTITY(0,1),
InterpretID int NOT NULL FOREIGN KEY(InterpretID) REFERENCES Interpret(InterpretID),
Nazev nvarchar(50) NOT NULL,
Interpret nvarchar(255) NOT NULL,
RokVydani date NULL,
Zanr nvarchar(25) NULL,
)
```

4.4.2.4 Tabulka Žánry

Tabulka „Zanry“ je důležitá pro popis informací o žánrech. Pro názornou ukázkou jsem popsal pouze tu nejhlavnější informaci, a to je jméno. U jména je použitý datový typ „nvarchar“.

```
CREATE TABLE Zanry
(
ZanrID int NOT NULL PRIMARY KEY IDENTITY (0,1),
Nazev nvarchar(255) NOT NULL,
)
```

4.4.2.5 Tabulka Skladba

Jelikož se jedná o databázi hudební aplikace, tak tabulka skladba jedna z nejdůležitějších tabulek z celé databáze. Ve skladbách nalezneme jméno interpreta, který danou skladbu složil, název skladby, délku skladby a rok vydání. U textových polí jsem zvolil datový typ „nvarchar“, délku jsem popsal za pomoci „time“ a rok vydání „date“. V této tabulce nalezneme hned 3 cizí klíče. Bez těchto klíčů bychom zdlouhavě museli do databáze vkládat nesmyslné množství informací, kdežto s jejich pomocí si je vezmou automaticky. První z cizího klíče je klíč pro tahání informací o albech z tabulky alb. Následující je žánr, ten je zde zapotřebí pro popis žánru, do kterého daná hudba spadá. Poslední je interpret, ten je nejdůležitější, protože z něj získáme informace o umělcovi, který skladbu vytvořil.

```
CREATE TABLE Skladba
(
```

```
SkladbaID int NOT NULL PRIMARY KEY IDENTITY(0,1),  
AlbumID int NOT NULL FOREIGN KEY(AlbumID) REFERENCES Album(AlbumID),  
ZanrID int NOT NULL FOREIGN KEY(ZanrID) REFERENCES Zanry(ZanrID),  
InterpretID int NOT NULL FOREIGN KEY(InterpretID) REFERENCES Interpret(InterpretID),  
Interpret nvarchar(255) NOT NULL,  
Nazev nvarchar(255) NOT NULL,  
Album nvarchar(50) NULL,  
Delka time NULL,  
RokVydani date NULL,  
)
```

4.4.2.6 Tabulka Skladba_zanr

Toto je pomocná tabulka databáze, která propojuje tabulky skladba a žánry. V tabulce nalezneme pouze cizí klíče obou tabulek.

```
CREATE TABLE skladbazanr  
(  
ZanrID int NOT NULL FOREIGN KEY(ZanrID) REFERENCES Zanry(ZanrID),  
SkladbaID int NOT NULL FOREIGN KEY(SkladbaID) REFERENCES Skladba(SkladbaID),  
)
```

4.4.2.7 Tabulka Role_uzivatelu

Tabulku „role_uzivatelu“ jsem vytvořil pro popsání rolí neboli práv jednotlivých uživatelů. Je zde důležitá z důvodu, že do aplikace se bude moci i registrovat, tudíž je důležité rozdělit práva administrátorů a uživatelů. Do tabulky jsem vložil pouze její ID a cizí klíč z tabulky uživatelů. Z důvodu přesného popsání uživatelů a jejich rolí.

```
CREATE TABLE Role_uzivatelu  
(  
RoleID int NOT NULL PRIMARY KEY IDENTITY (0,1),  
UzivatelID int NOT NULL FOREIGN KEY(uzivatelID) REFERENCES Uzivatel(uzivatelID),  
)
```

4.4.2.8 Tabulka *Hodnoceni_skladby*

Jako poslední tabulku jsem vytvořil tabulku pro hodnocení skladby. Do tabulky jsem vložil všechny potřebné atributy, to jsou především cizí klíče. Jelikož je to tabulka, která bude ukládat hodnocení jednotlivých skladeb, je důležité, aby měla přístup ke všem potřebným informacím o skladbách, uživateli a žánru. Toho jsem dosáhl propojením těchto tabulek za pomoci cizího klíče. Jako jediný vlastní atribut tabulky je hodnocení, které má datový typ „float“. Jelikož se bude počítat průměr všech hodnocení uživatelů, je zapotřebí datový typ, který bude umět pracovat s desetinou čárkou.

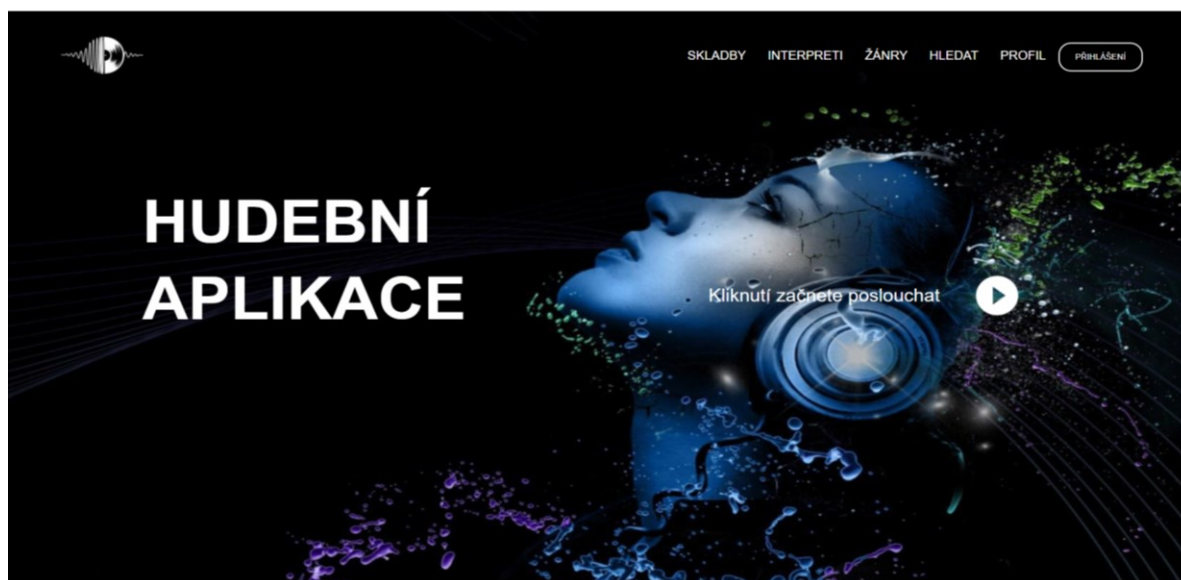
```
CREATE TABLE Hodnoceni_skladby
(
HodnoceniID int NOT NULL PRIMARY KEY IDENTITY(0,1),
UzivatelID int NOT NULL FOREIGN KEY(UzivatelID) REFERENCES Uzivatel(UzivatelID),
SkladbaID int NOT NULL FOREIGN KEY(SkladbaID) REFERENCES Skladba(SkladbaID),
ZanrID int NOT NULL FOREIGN KEY(ZanrID) REFERENCES Zanry(ZanrID),
Hodnoceni float NOT NULL,
)
```

5 MODEL NAVRŽENÉ APLIKACE

Model aplikace jsem navrhoval za pomoci HTML, CSS, JS a u některých stránek i Bootstrap. Základní kód jsem napsal za pomoci HTML. Design je pak vytvořen s využitím CSS. U některé ze stránek jsem pro design tabulky použil již zmíněný Bootstrap. Jelikož se jedná o hudební aplikaci, tak bylo vhodné zprovoznit přehrávání hudby. K přehrávání hudby jsem použil jednoduchý script, který jsem vložil přímo do HTML kódu.

Navrhoval jsem model aplikace tak, aby pro vývojáře bylo co nejjednodušší vytvořit funkční aplikaci podle tohoto návrhu.

5.1 Hlavní stránka



Obrázek 16 – Hlavní stránka

Jako první jsem začal vytvářet hlavní stránku, na kterou se uživatel dostane po prvním vstupu do hudební aplikace. Na stránce se nachází menu vytvořené pomocí navigační baru spolu s tlačítkem pro přihlášení. Do levého rohu jsem umístil logo aplikace, které funguje jako odkaz. Po kliknutí na logo bude návštěvník přesměrován na hlavní stránku. Dále zde naleznete obrázek se symbolem „play“. Tento obrázek funguje jako tlačítko. Pokud kliknete na toto tlačítko automaticky se spustí hudba. Po opětovném kliknutí se skladba zastaví.

```
<body>
  <div class="container">
    <div class="navbar">
      <a href="index.html">
        
      </a>
      <ul>
        <li><a href="skladby.html">SKLADBY</a></li>
        <li><a href="interpreti.html">INTERPRETI</a></li>
        <li><a href="zanry.html">ŽÁNRY</a></li>
        <li><a href="hledani.html">HLEDAT</a></li>
        <li><a href="profil.html">PROFIL</a></li>
        <button onclick="location.href='login.html'"
          style="width:auto;">PŘIHLÁŠENÍ</button>
      </ul>
    </div>
  </div>
```

Obrázek 17 – Navigační menu

5.2 Přihlášení a registrace



Obrázek 18 – Stránka pro registraci

Pro přihlášení do aplikace jsem vytvořil tlačítko, které je umístěno na hlavní stránce. Po kliknutí bude uživatel přesměrován na stránku s formulářem pro přihlášení. Pokud máte již vytvořený uživatelský účet, můžete vyplnit přihlašovací údaje do formuláře a přihlásit se. Pokud nevlastníte účet můžete se registrovat, ale není to potřeba. Aplikaci lze využívat i bez uživatelského účtu. Uživatelé, kteří nebudou chtít využít možnosti registrace, budou moci využívat téměř všechny funkce, až na pár výjimek, jako je například profil uživatele. Na stránce s registrací naleznete formulář pro vyplnění vašich údajů. U registrace po vás bude vyžadováno vaše jméno a příjmení, email, uživatelské jméno a heslo. Tento formulář se automaticky validuje, takže pokud vyplníte například email bez zavináče, tak vás aplikace

upozorní na chybu a nepustí vás dál, dokud chybu neopravíte. Formulář jsem vytvořil za pomoci „form“ a vkládaný obsah je zformulován pomocí tagu „input“.

```
<main>
  <div class="background">
    <div class="text">
      <h1>Registrace</h1>
      <p> Máš už účet? <a href="login.html">Přihlas se</a></p>
    </div>
    <div class="box">
      <form class="form" action="login.html">
        <input type="text" class="name" placeholder="Jméno" required>
        <input type="text" class="name" placeholder="Příjmení"
          required>
        <input type="email" class="email" placeholder="Email" required>
        <input type="text" class="username" placeholder="Uživatelské
          jméno" required>
        <input type="password" class="password" placeholder="Heslo"
          required>
        <input type="submit" class="button" value="Registrovat">
      </form>
    </div>
  </div>
</main>
```

Obrázek 19 – Tvorba registračního formuláře

5.3 Stránka skladby

Databázi hudby naleznete na stránce skladby. Zde jsem vytvořil galerii s hudbou za pomoci blokového elementu „div“ a definování kaskádových stylů. U každé skladby je k dispozici obrázek a základní informace o tom, kdo ji složil a jak se jmenuje.

Jelikož se jedná o hudbu je důležité, aby ji bylo možné přehrát. Pro spuštění skladby stačí kliknou na její fotku a přehrávání se spustí. Toho jsem docílil pomocí skriptu, který jsem nadefinoval pro každou z jednotlivých skladeb. Bez vložení hudby do základního kódu by to nešlo. K vložení jsem použil tag „audio“, který jsem vybavil o „id“ které je důležité pro již zmíněný skript. Také k obrázku jsem musel pro fungování přehrávání definovat „id“. Skript funguje tak, že po kliknutí na obrázek s odpovídajícím „id“ se začne přehrávat vložená hudba s přesně definovaným „id“ a po opětovném kliknutí se zastaví. Toto „id“ jsem implementoval do skriptu proto, abych přesně označil a poznal cesty po kterých se má skript vy-
dávát.

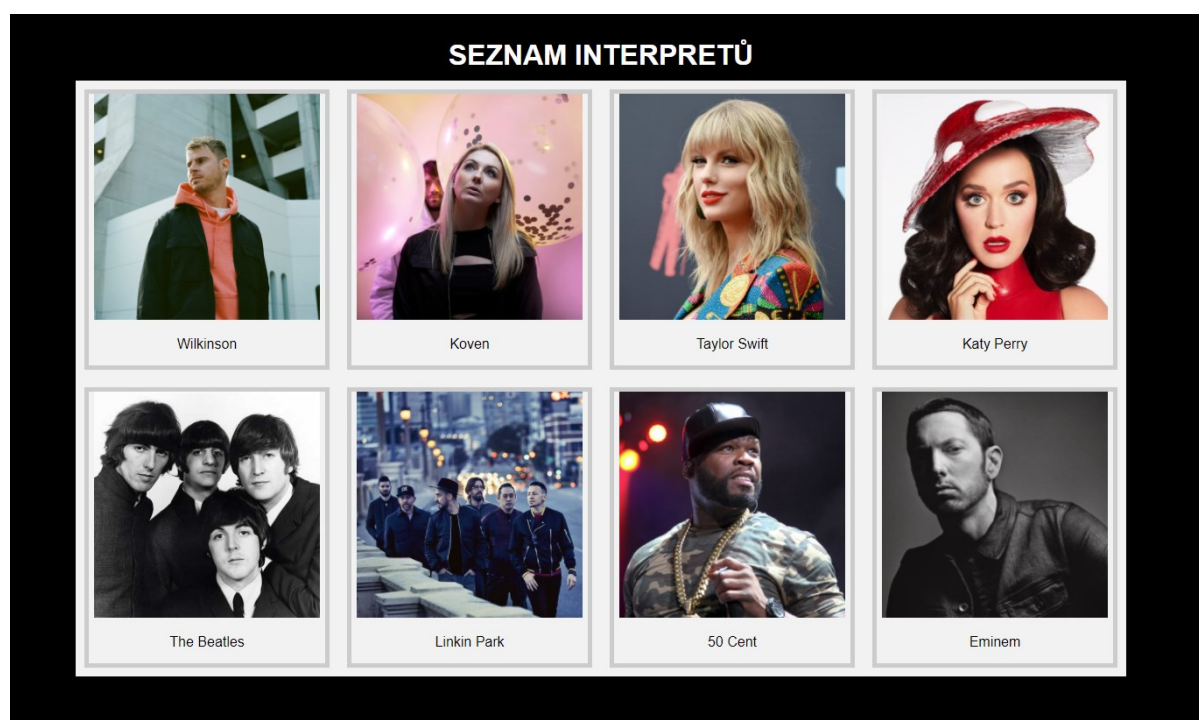
```
<script>
var mySong = document.getElementById("mySong");
var icon = document.getElementById("icon");

icon.onclick = function(){
  if(mySong.paused){
    mySong.play();
    icon.src = "media/after.jpg";
  }else{
    mySong.pause();
    icon.src = "media/after.jpg";
  }
}
</script>
```

Obrázek 20 – Ukázka JavaScriptu

5.4 Stránka interpreti

Na této stránce naleznete všechny interprety, kteří se nachází v databázi. Vypracoval jsem ji podobně jako u předchozí stránky. Všichni dostupní interpreti mají v této galerii svoje profilové obrázky doplněné o jejich jméno. Obrázky fungují jako odkazy. Po kliknutí na interpreta budete přesunuti na jeho profil. Na profilu naleznete všechny dostupné informace. Například se jedná o to, do kterého žánru spadá nebo kdy se narodil. U každého je také spočítáno kolik vydal alb a skladeb. Některé vybrané skladby jsou vloženy do jeho profilu a lze je přehrát.



Obrázek 21 – Stránka interpreti

5.5 Stránka žánry

Na stránce jsou vypsané všechny žánry. Po rozkliknutí jednotlivého žánru bude uživatel přeměrován na další stránku, kde nalezne podrobné údaje o žánru. Jsou zde spočítané všechny skladby, všichni interpreti a veškerá alba, která spadají do tohoto odvětví. Dále zde je seznam interpretů a skladeb.

```
.navbar{  
  width: 90%;  
  margin: auto;  
  padding: 5px 0;  
  display: flex;  
  align-items: center;  
  justify-content: space-between;  
}
```

Obrázek 22 – Ukázka kaskádových stylů

5.6 Profil uživatele

Profil uživatele je dostupný jen pro registrované uživatele a je možné ho zobrazit až po přihlášení. Na profil jsou vloženy uživatelské statistiky, které se týkají přehrávání hudby. Ty jsou doplněny o základní informace a jeho playlist. K vytváření všech profilů jsem použil technologii Bootstrap. Tu jsem musel importovat do HTML kódu pomocí tagu „link“.

```
<link rel="stylesheet" href='https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css'>  
<script src='https://stackpath.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.bundle.min.js'></script>  
<script src='https://cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js'></script>
```

Obrázek 23 – Připojení Bootstrapu

ZÁVĚR

Cílem této práce bylo navrhnout aplikaci pro hudební portál. V teoretické části práce jsem jako první popsal řešerše hudebních aplikací. Jedná se o aplikace Spotify, SoundCloud a Youtube music. Zmínil jsem zde dostupnost pro uživatele těchto aplikací. Popsal jsem, jak vypadá jejich uživatelské prostředí a jak moc kvalitní hudbu dokážou přehrát. V kapitole teoretické části byli rozebrány technologie, které jsem následně použil v praktické části. U každé z technologií jsou popsány základní informace a způsob používání. V poslední části jsem se věnoval zabezpečení a možným útokům na webové aplikace. Popsal jsem, jak takové útoky fungují a jak se proti nim dá bránit. V praktické části popisují funkcionality navrhované aplikace spolu s případy užití aktérů. Ke každému případu užití jsem vytvořil jeho scénář, ve kterém jsou krok po kroku popsány jednotlivé úkony. Poté jsem vymodeloval databázi, kterou bude tato aplikace potřebovat pro svůj provoz. Nakonec jsem za pomoci HTML kódu vytvořil prototyp aplikace. Pro design stránek jsem použil technologii CSS, kterou jsem u některých případů doplnil o Bootstrap. K přehrávání hudby jsem použil JavaScript. Vytvořený prototyp jsem popsal, aby bylo zřejmé, jak funguje.

Kompletace bakalářské práce mě bavila, i když nebyla zrovna nejjednodušší. Některé části práce mě dali opravdu zabrat, ale alespoň jsem získal cenné zkušenosti, které mohu následně uplatnit v reálném životě.

SEZNAM POUŽITÉ LITERATURY

- [1] *Skrblik* [online]. [cit. 2021-5-31]. Dostupné z: <https://www.skrblik.cz/telefon/online-hudba-zdarma-srovnani-spotify-deezer-google-play-a-apple-music/#spotify>
- [2] *Zive* [online]. [cit. 2021-5-31]. Dostupné z: <https://www.zive.cz/clanky/spotify-chce-byt-prehravacem-pro-vsechny-na-androidu-planuje-prehravani-lokalni-hudby/sc-3-a-207361/default.aspx>
- [3] *Spotify* [online]. [cit. 2021-5-31]. Dostupné z: <https://www.spotify.com/cz/premium/>
- [4] *Letemsvetemapplem* [online]. [cit. 2021-5-31]. Dostupné z: <https://www.letemsvetemapplem.eu/2019/08/23/spotify-premium-vse-co-jste-chteli-vedet/>
- [5] *Letemsvetemapplem* [online]. [cit. 2021-5-31]. Dostupné z: <https://www.letemsvetemapplem.eu/2019/03/30/10-tipu-a-triku-ktere-musi-znat-kazdy-uzivatel-spotify/>
- [6] *Jablickar* [online]. [cit. 2021-5-31]. Dostupné z: <https://jablickar.cz/spotify-tipy-pro-zacatecniky-jak-hudebni-sluzbu-plne-vyuzivat/>
- [7] *Svetandroida* [online]. [cit. 2021-5-31]. Dostupné z: <https://www.svetandroida.cz/top-10-aplikace-na-streamovani-hudby/>
- [8] *Digitips* [online]. [cit. 2021-5-31]. Dostupné z: <https://digitips.cz/soundcloud-poslouchejte-a-sdilejte-hudbu-zdarma/>
- [9] *Soundcloud* [online]. [cit. 2021-5-31]. Dostupné z: <https://checkout.soundcloud.com/pro>
- [10] *Napocitaci* [online]. [cit. 2021-5-31]. Dostupné z: https://www.napocitaci.cz/33/soundcloud-socialni-sit-pro-sdileni-audio-uniqueid-gOkE4NvrWuNY54vrLeM67zPQ3mSYj0OsAg_rzFJ8D5c/

- [11] *Idnes* [online]. [cit. 2021-5-31]. Dostupné z: https://www.idnes.cz/tech-net/audio-foto-video/youtube-music-premium-startuje-v-cr-za-149-kc-mesicne.A190513_083517_audio-foto-video_nyv
- [12] *Dotekomanie* [online]. [cit. 2021-5-31]. Dostupné z: <https://dotekomanie.cz/2019/05/youtube-music-premium-cesko-cena/>
- [13] *Letemsvetemapplem* [online]. [cit. 2021-5-31]. Dostupné z: <https://www.letemsvetemapplem.eu/2019/08/06/10-tipu-a-triku-jak-vyuzit-youtube-premium-a-youtube-music-na-maximum/>
- [14] *Developer* [online]. [cit. 2021-5-31]. Dostupné z: <https://developer.mozilla.org/en-US/docs/Web/HTML>
- [15] *Jakpsatweb* [online]. [cit. 2021-5-31]. Dostupné z: <https://www.jakpsatweb.cz/html/html-5.html>
- [16] *Jakpsatweb* [online]. [cit. 2021-5-31]. Dostupné z: <https://www.jakpsatweb.cz/html/struktura.html>
- [17] *Itnetwork* [online]. [cit. 2021-5-31]. Dostupné z: <https://www.it-network.cz/html-css/html5/html-manual/html-css-html-manual-struktura/html-struktura-stranky-cesky-manual/>
- [18] JOHNSON, Glenn. *Programming in HTML5 with JavaScript and CSS3: training guide*. Redmond, Wash.: Microsoft, 2013. ISBN 978-0735674387
- [19] *Jakpsatweb* [online]. [cit. 2021-5-31]. Dostupné z: <https://www.jakpsatweb.cz/html/text.html>
- [20] *Jakpsatweb* [online]. [cit. 2021-5-31]. Dostupné z: <https://www.jakpsatweb.cz/html/seznamy.html>
- [21] *Tvorba-webu* [online]. [cit. 2021-5-31]. Dostupné z: <https://www.tvorba-webu.cz/html/odkazy.php>
- [22] *Jakpsatweb* [online]. [cit. 2021-5-31]. Dostupné z: <https://www.jakpsatweb.cz/obrazky.html>

- [23] *Jakpsatweb* [online]. [cit. 2021-5-31]. Dostupné z: <https://www.jakpsatweb.cz/html/tabulky.html>
- [24] *Itnetwork* [online]. [cit. 2021-5-31]. Dostupné z: <https://www.itnetwork.cz/html-css/css3/uvod-do-css3>
- [25] *Jakpsatweb* [online]. [cit. 2021-5-31]. Dostupné z: <https://www.jakpsatweb.cz/css/css-uvod.html#syntaxe>
- [26] *Klikzone* [online]. [cit. 2021-5-31]. Dostupné z: <https://www.klikzone.cz/CSS-navod/zpusoby-pouziti-CSS.php>
- [27] *Literama* [online]. [cit. 2021-5-31]. Dostupné z: <https://www.literama.sk/data/produkty/dokumenty/1/6/8/7/4/6/285-1001-tipu-a-triku-pro-sql.pdf>
- [28] *Voho* [online]. [cit. 2021-5-31]. Dostupné z: <http://voho.eu/wiki/sql/>
- [29] *Dotnetportal* [online]. [cit. 2021-5-31]. Dostupné z: <https://www.dotnetportal.cz/clanek/50/Uvod-do-jazyka-SQL>
- [30] *Itnetwork* [online]. [cit. 2021-5-31]. Dostupné z: <https://www.itnetwork.cz/ms-sql/ms-sql-datove-typy-podrobneji>
- [31] *Votruba* [online]. [cit. 2021-5-31]. Dostupné z: <http://votruba.in/int/sql-prikazy.pdf>
- [32] *Dotnetportal* [online]. [cit. 2021-5-31]. Dostupné z: <https://www.dotnetportal.cz/clanek/50/Uvod-do-jazyka-SQL>
- [33] *Is.muni* [online]. [cit. 2021-5-31]. Dostupné z: <https://is.muni.cz/th/j5o82/BP.pdf>
- [34] *Itnetwork* [online]. [cit. 2021-5-31]. Dostupné z: <https://www.itnetwork.cz/navrh/uml/uml-use-case-diagram>
- [35] *Objects* [online]. [cit. 2021-5-31]. Dostupné z: https://www.objects.cz/wp/is_uml/kdy-pouzit-v-use-case-diagramu-vztah-extend-a-kdy-include-cast-4/

- [36] *Itnetwork* [online]. [cit. 2021-5-31]. Dostupné z: <https://www.it-network.cz/navrh/uml/uml-use-case-specifikace-diagram>
- [37] *Uml.czweb* [online]. [cit. 2021-5-31]. Dostupné z: http://uml.czweb.org/diagram_trid.htm
- [38] *Moodle.fel.cvut* [online]. [cit. 2021-5-31]. Dostupné z: https://moodle.fel.cvut.cz/pluginfile.php/103006/mod_label/intro/Prednask5UML-diagramTrid_a_Stavu.pdf
- [39] *Ecom.ef.jcu* [online]. [cit. 2021-5-31]. Dostupné z: http://ecom.ef.jcu.cz/web2/download/teorie/08_diagramy_trid_pou-ziti.pdf
- [40] *Itnetwork* [online]. [cit. 2021-5-31]. Dostupné z: <https://www.it-network.cz/navrh/uml/uml-class-diagram-tridni-model>
- [41] *Rascasone* [online]. [cit. 2021-5-31]. Dostupné z: <https://www.rascasone.com/cs/blog/co-je-javascript-pro-zacatecniky>
- [42] *Strafelda* [online]. [cit. 2021-5-31]. Dostupné z: <https://www.strafelda.cz/javascript>
- [43] *Getbootstrap* [online]. [cit. 2021-5-31]. Dostupné z: <https://getbootstrap.com/docs/4.1/about/overview/>
- [44] *Zdrojak* [online]. [cit. 2021-5-31]. Dostupné z: <https://zdrojak.cz/clanky/k-cemu-je-dobry-bootstrap-frontend-frameworky/>
- [45] *Itnetwork* [online]. [cit. 2021-5-31]. Dostupné z: <https://www.it-network.cz/javascript/zaklady/javascript-tutorial-uvod-do-javascriptu-nepochopeny-jazyk>
- [46] *7interactive* [online]. [cit. 2021-5-31]. Dostupné z: <https://www.7interactive.cz/cz/stranka/rady-a-tipy/jak-si-zabezpecit-svuj-web-pomoci-https.html>

- [47] *Martindomes* [online]. [cit. 2021-5-31]. Dostupné z: <https://www.martindomes.cz/jak-na-https-aneb-jak-se-zbavit-nalepky-nezabezpeceneho-webu/>
- [48] *Itnetwork* [online]. [cit. 2021-5-31]. Dostupné z: <https://www.it-network.cz/php/bezpecnost/technika-utoku-sql-injection>
- [49] *Computerworld* [online]. [cit. 2021-5-31]. Dostupné z: <https://computerworld.cz/internet-a-komunikace/hlavni-hrozby-pro-firemni-web-3400>
- [50] *Systemonline* [online]. [cit. 2021-5-31]. Dostupné z: <https://www.systemonline.cz/it-security/hrozby-pro-bezpecnost-webovych-aplikaci-a-serveru.htm>
- [51] *Bartvisions* [online]. [cit. 2021-5-31]. Dostupné z: <https://www.bartvisions.cz/2019/05/zaklady-zabezpeceni-webovych-stranek-ddos-utok/>
- [52] *Raidboxes* [online]. [cit. 2021-5-31]. Dostupné z: <https://raidboxes.io/cs/blog/security/ddos-attack/>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

EUR Euro

USD Americký dolar

HTML Hypertext Markup Language

CSS Cascading Style Sheets

JS JavaScript

DTD Document Type Definition

SQL Structured Query Language

DML Data Manipulation Language

DDL Data Definition Language

DCL Data Control Language

TCL Transactional Control Language

UML Unified Modeling Language

UC Use case

iOS Mobilní operační systém pro telefony iPhone společnosti Apple

CMS Content management systém

HTTPS Hypertext Transfer Protocol Secure

GDPR Nařízení o ochraně osobních údajů

URL *Uniform Resource Locator*

ID Identifikace

XSS Cross site scripting

CAPTCHA Test sloužící k odlišení počítačů od lidí

Dos Denial of Service

DDos Distributed Denial of Service

SEZNAM OBRÁZKŮ

Obrázek 1 – Spotify aplikace [2]	11
Obrázek 2 – SoundCloud aplikace [8]	14
Obrázek 3 – Aplikace YouTube music [11]	17
Obrázek 4 – Aktér [34]	29
Obrázek 5 – Příklad užití [34]	31
Obrázek 6 – Vztah extend [35]	32
Obrázek 7 – Diagram tříd [40]	36
Obrázek 8 – Ukázka útoku pomocí SQL příklad č.1 [48]	40
Obrázek 9 – Ukázka útoku pomocí SQL příklad č.1 obr. 2 [47]	41
Obrázek 10 – Ukázka útoku pomocí SQL příklad č.2 [48]	41
Obrázek 11 – Ukázka útoku pomocí SQL příklad č.4 [48]	42
Obrázek 12 – Funkční a nefunkční požadavky	46
Obrázek 13 – Funkční požadavky	47
Obrázek 14 – Nefunkční požadavky	51
Obrázek 15 – Diagram databáze	71
Obrázek 16 – Hlavní stránka	76
Obrázek 17 – Navigační menu	77
Obrázek 18 – Stránka pro registraci	77
Obrázek 19 – Tvorba registračního formuláře	78
Obrázek 20 – Ukázka JavaScriptu	79
Obrázek 21 – Stránka interpretu	79
Obrázek 22 – Ukázka kaskádových stylů	80
Obrázek 23 – Připojení Bootstrapu	80

SEZNAM TABULEK

Tabulka 1 - UC01: Registrace uživatele	54
Tabulka 2 - UC02: Zobrazení registračního formuláře	55
Tabulka 3 - UC03: Přihlášení	56
Tabulka 4 - UC04: Zobrazení přihlašovacího formuláře	57
Tabulka 5 - UC05: Změna uživatelských údajů	57
Tabulka 6 - UC06: Změna hesla	58
Tabulka 7 - UC07: Obnova zapomenutého hesla	59
Tabulka 8 - UC08: Přehrávání hudby	60
Tabulka 9 - UC09: Vyhledávání hudby	61
Tabulka 10 - UC10: Zobrazení základních informací o skladbách	62
Tabulka 11 - UC11: Zobrazení základních informací o interpretech	62
Tabulka 12 - UC12: Zobrazení základních informací o hudebních žánrech	63
Tabulka 13 - UC13: Přidávání skladeb	64
Tabulka 14 - UC14: Vyhledávání obsahu podle potřeb uživatele	65
Tabulka 15 - UC15: Podle skladeb	66
Tabulka 16 - UC16: Podle interpretů	67
Tabulka 17 - UC17: Podle žánrů	67
Tabulka 18 - UC18: Editace obsahu	68
Tabulka 19 - UC19: Správa uživatelů	69
Tabulka 20 - UC20: Přehled statistik	69

SEZNAM PŘÍLOH

P1 Obsah DVD

PŘÍLOHA P I: NÁZEV PŘÍLOHY

DVD obsahuje všechny zdrojové kódy práce.

- Text práce uložený ve formátu PDF.
- Veškeré zdrojové soubory jsou ve formátu ZIP.