

Mendelova univerzita v Brně
Provozně ekonomická fakulta

Aktualizace webové aplikace společnosti Capellen Orchestra

Bakalářská práce

Vedoucí práce:
Ing. Oldřich Faldík

Marie Dorušková

Brno 2016

Děkuji svému vedoucímu bakalářské práce Ing. Oldřichu Faldíkovi za jeho cenné odborné rady, vedení, zprostředkování tématu a ochotnou komunikaci v průběhu celého řešení. A další děkuji patří Mgr. Petru Pololáníkovi za umožnění pro něj pracovat.

Čestné prohlášení

Prohlašuji, že jsem tuto práci: **Aktualizace webové aplikace společnosti Capellen Orchestra**

vypracovala samostatně a veškeré použité prameny a informace jsou uvedeny v seznamu použité literatury. Souhlasím, aby moje práce byla zveřejněna v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů, a v souladu s platnou *Směrnicí o zveřejňování vysokoškolských závěrečných prací*.

Jsem si vědoma, že se na moji práci vztahuje zákon č. 121/2000 Sb., autorský zákon, a že Mendelova univerzita v Brně má právo na uzavření licenční smlouvy a užití této práce jako školního díla podle § 60 odst. 1 Autorského zákona.

Dále se zavazuji, že před sepsáním licenční smlouvy o využití díla jinou osobou (subjektem) si vyžádám písemné stanovisko univerzity o tom, že předmětná licenční smlouva není v rozporu s oprávněnými zájmy univerzity, a zavazuji se uhradit případný příspěvek na úhradu nákladů spojených se vznikem díla, a to až do jejich skutečné výše.

místo a datum prohlášení

.....

Abstract

Dorušková, M. Update of Capellen Orchestra web application. Bachelor thesis. Brno: Mendel university, 2016

The aim of thesis was to create a new web application for the company on behalf of Capellen Orchestra. As the first step was necessary to analyze the condition of the original application and consult with the client functional and nonfunctional requirements. After discovering the situation on the current market with editorial systems, it has been designed a custom solution that includes database design, system design and graphical interface. Subsequently, was proposed solution deployed by Nette framework in conjunction with other tools and MySQL database. In the final stage, the application has been tested and deployed to production environment.

Keywords

Web application, MySQL, Nette framework, HTML5, CSS3, JavaScript

Abstrakt

Dorušková, M. Aktualizace webové aplikace společnosti Capellen Orchestra. Bachelářská práce. Brno: Mendelova univerzita v Brně, 2016

Cílem práce bylo vytvořit novou webovou aplikaci pro společnost jménem Capellen Orchestra. Nejdříve bylo nutné analyzovat stav původní aplikace a zkontrolovat se zadavatelem funkční a nefunkční požadavky. Po zjištění situace na současném trhu s redakčními systémy bylo navrženo vlastní řešení, které zahrnuje návrh databáze, návrh systému a grafického rozhraní. Následně bylo navržené řešení naimplementováno pomocí frameworku Nette společně s dalšími nástroji a databází MySQL. V konečné fázi byla aplikace otestována a nasazena do produkčního prostředí.

Klíčová slova

Webová aplikace, MySQL, Nette framework, HTML5, CSS3, JavaScript

Obsah

1	Úvod a cíl práce	7
1.1	Úvod	7
1.2	Cíl práce	7
2	Analýza současného stavu	8
2.1	Funkční požadavky	9
2.2	Nefunkční požadavky	10
2.3	Existující redakční systémy	10
2.3.1	WordPress	10
2.3.2	Joomla	11
2.3.3	Drupal	12
2.3.4	TextPattern	12
2.4	Hodnocení současných redakčních systémů na trhu	13
3	Návrh vlastního řešení	14
3.1	Diagram případu užití	14
3.2	Návrh databázového schématu	15
3.2.1	Popis jednotlivých entit	16
3.3	Zajištění historických dat	17
3.4	MVC a MVP architektura	18
3.5	Návrh uživatelské struktury aplikace	19
3.5.1	Hlavní stránka	19
3.5.2	Barevné schéma a typografie	20
3.5.3	Uživatelské rozhraní	21
3.5.4	Administrativní rozhraní	21
4	Implementace	23
4.1	Použité technologie	23
4.1.1	Nette Framework	23
4.1.2	MySQL	23
4.1.3	Nette Componette: Addons, plugins and components	23
4.1.4	Použité grafické prvky	24
4.2	Vývojové prostředí	25
4.2.1	WampServer	25
4.2.2	NetBeans	25
4.2.3	IBM Bluemix	25
4.3	Vybrané funkce systému	26
4.3.1	Generace faktury	26
4.3.2	Online platba faktury	27
4.3.3	LiveStream	29
4.4	Diagram tříd	29
4.5	Bezpečnost	30

4.5.1	Autorizace na základě rolí	30
4.5.2	Bezpečnostní rizika	30
5	Testování	32
5.1	Uživatelské testování	32
5.2	Selenium IDE	32
6	Nasazení	34
6.1	Výsledné prostředí	34
7	Závěr	35
8	Reference	36
	Přílohy	39
A	Původní systém	40
B	Fyzický model	41
C	Diagram tříd	42

1 Úvod a cíl práce

1.1 Úvod

V současné době člověk žije v době moderních technologií, které se neustále vyvíjí a ve svém bodě už dospěly i k tomu, že dokážou nahradit určité činnosti samotného člověka. Lidé si ani neuvědomují, kde všude se dají nové informační technologie využít, například hudební odvětví. Lidé chodí do kina, dívají se na seriály a v pozadí vnímají hudbu, která ovlivňuje emoce a veškeré pocity člověka. Vykresluje tedy celou atmosféru týkající se daného filmu či seriálu. Kde se ale tato hudba vzala? Kde vznikl ten prvotní krok pro nahrání určité hudební myšlenky?

I v této oblasti jsou informační technologie potřeba a usnadňují mnoho práce nejen všem producentům nebo majitelům hudebních zařízení, ale i samotným zájemcům o danou službu. V každém případě je nutné evidovat mnoho informací jak o klientech, tak o různých datech ať už důvěrných či veřejných. Dále se vytváří objednávky, faktury, platby a tak dochází k určitému komunikačnímu toku mezi uživatelem a majitelem. Představa uchovávat všechny tyto informace v papírové podobě je náročná a situace se následně stává nepřehlednou. Komunikace by byla také náročnější, kdyby si pokaždé měli všichni volat a domlouvat se v současném světě plném stresu. Lidé pomocí těchto technologií dokážou zefektivnit práci tak, aby informace byly uchovány přehledně a dalo se k nim jednoduše dostat. Komunikace je lehčí a nedochází k nesrovnalostem díky systému, který automaticky vyhodnotí danou situaci.

Vhodným řešením je tedy tvorba nové webové aplikace založená na požadavcích zadavatele. Tato aplikace umožní například evidovat zákazníky, tvořit objednávky pro nahrávání živého orchestru například do filmu, generovat, platit faktury pomocí online platební brány a odlehčí tak celkovou administrativní práci, kterou majitel musí zajistit. Celá aplikace tak usnadňuje práci majiteli a lidé si mohou užít hudbu v pozadí filmu či seriál.

1.2 Cíl práce

Současná aplikace, která je používána majitelem společnosti Capellen Music Production je zcela nevyhovující vzhledem k rozšiřujícím požadavkům majitele, ale také z pohledu bezpečnosti. Schéma databáze je zastaralé a vzhledově web zaostává za současnými trendy ve světě webového designu. Cílem práce je tedy analyzovat všechny požadavky, které majitel vyžaduje. Na základě získaných informací navrhnout a nasadit novou webovou aplikaci, která bude postavena na moderních technologiích. V první fázi je nutné zanalyzovat redakční systémy, které nabízí současný trh a vyhodnotit, zda jsou tyto systémy vhodné svým charakterem pro zadanou aplikaci. Po této fázi lze přejít na návrh vlastního řešení a implementaci, která bude funkční a rozsahem vyhovující.

2 Analýza současného stavu

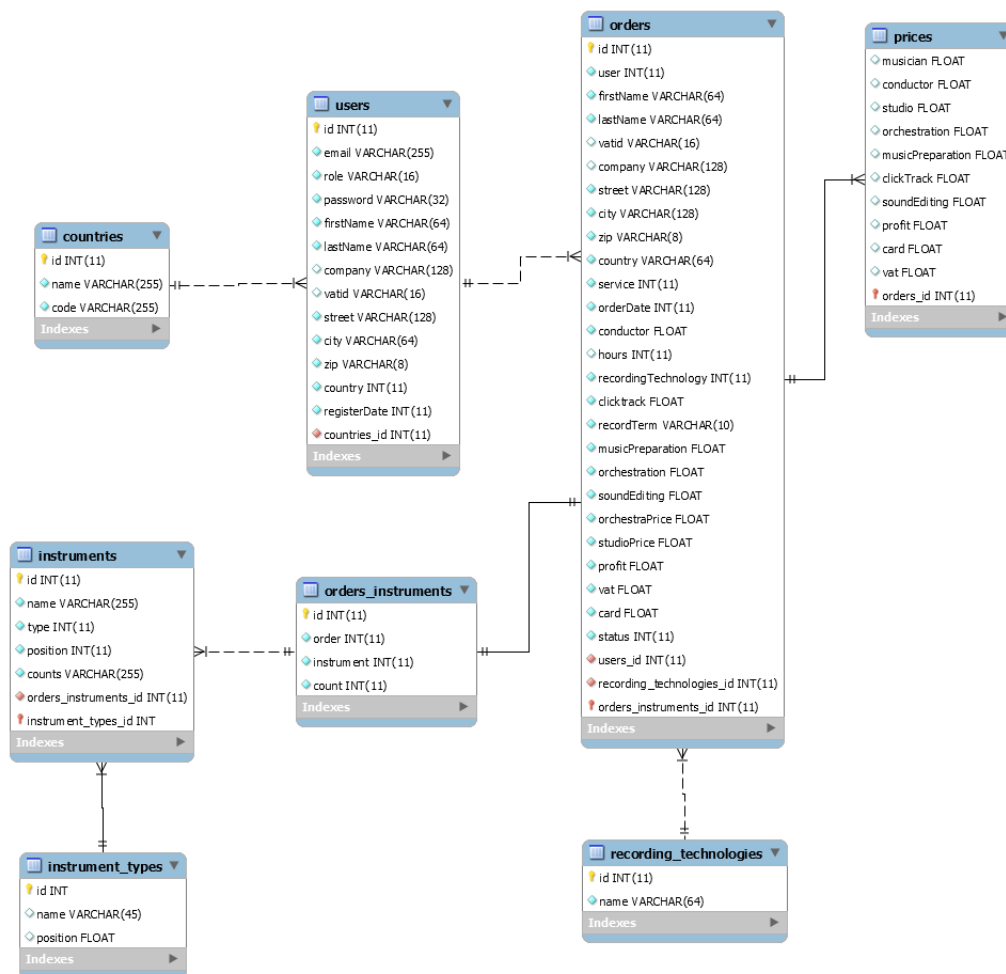
Firma *Capellen Music Production* se zabývá zpracováním orchestrálního aranžmá a nahráváním filmové hudby. Klienti této společnosti jsou převážně zahraniční, a proto tato společnost vyžaduje novou webovou prezentaci.

V současné době majitel této firmy využívá původní webovou aplikaci, která je schopna zaregistrovat zákazníka a následně zpracovat objednávku dle zadaného vzorce. Dále umí zasílat faktury po vytvoření objednávky na e-mail klienta a objednaná služba je klientovi dodávána pomocí FTP manuálně odkazem. Z pohledu administrátora je systém schopen upravovat ceny služeb, přidávat či odebírat určité typy služeb nebo upravovat hudební nástroje. Na první pohled tedy aplikace dělá co má, ale způsob, jakým byla aplikace vytvořena už správný není. Fyzický model databáze (viz obr. 1) má některé tabulky typu MyISAM a některé InnoDB, což způsobuje zmatek, nekonzistenci a necelistvost dat. V celém schématu nejsou rozpoznávány cizí klíče, tudíž se neustále opakují v tabulkách stejné atributy a tabulka se zbytečně rozrůstá. Neexistující relace v databázi porušuje integritu databáze a vztahy mezi tabulkami neexistují. Z těchto důvodů nelze přenést historická data na nový systém s novou databází. Atributy jsou pojmenovány nepřesně a není jednoznačný jejich smysl. Problém je i u některých tabulek, protože například tabulka s typy instrumentů je v systému nepoužitelná a ztrácí tedy svůj význam. Co se týče rozšiřitelnosti původní databáze do budoucna, je tato možnost zcela vyloučena právě z již jmenovaných důvodů.

Z pohledu uživatele a případně potenciálního klienta je tento web nezvládnutý i co se týče designu, marketingu a vizuální prezentace společnosti. Webová stránka působí na první pohled neprofesionálně, zastarale a její struktura je nepřehledná. Není využito responzivity, proto dochází k problému zobrazení tohoto webu na jiném zařízení, například na telefonu či tabletu. Tato skutečnost je zachycena na obr. 14.

Ve chvíli, kdy klient zadá adresu této stránky (*capellen.cz*), dostane se na tzv. front-page, která funguje jako rozcestník pro dvě jazykové verze webu. Problémem je, že tyto verze nejsou identické a pouze na anglické verzi se může klient zaregistrovat a vytvořit objednávku. Při procházení stránky se uživatel nachází v obrovském množství textu a pokud chce něco vyhledat, pak se vším musí složitě proklikat. Tento problém je způsoben nejasnou navigací.

Pokud se člověk přihlásí nebo zaregistruje do systému, pak může pouze učinit objednávku v rozklikávacím formuláři, který je ukončen tlačítkem pro platbu objednávky pomocí PayPal účtu nebo kreditní karty. Zákazník není celou dobu informován o tom, zda byla objednávka zaevidována v systému. Ukáže se mu pouze celková částka a tlačítko k platbě PayPal. V profilu uživatele také nejsou vedeny žádné informace o objednavce. Zákazník je o objednavce informován následně e-mailem, o kterém není zákazník informován. Systém nerozlišuje již uskutečněné přihlášení zákazníka. Musí být opakovaně zadávány přihlašovací údaje pro vstup k tvorbě objednávky. Z jmenovaných důvodů bylo tedy vyhodnoceno, že původní aplikace je nepoužitelná a musí být navržena zcela nová webová aplikace.



Obrázek 1: Původní schéma databáze

2.1 Funkční požadavky

Majitel společnosti během konzultací vyslovil několik konkrétních požadavků na webovou aplikaci pro Capellen Music Production. Funkční požadavky jsou následující:

- kalkulace služeb – výpočet částky vybraných služeb v objednávce pomocí zadaného vzorce,
- živý přenos – integrace přehrávače do profilu uživatele. Přenos pomocí kamery bude zajištěn externí firmou LIVEBOX, a.s.,
- online platby přes PayPal – propojení aplikace s platební bránou PayPal,
- fakturace – generování faktur do formátu PDF,
- schopnost rozlišit uživatele podle rolí – zda se jedná o administrátora, který

může upravovat ceny služeb nebo běžného uživatele, který pouze zadává data do systému,

- multimediální galerie – umožnit zobrazení fotek, přehrávání hudby a videí.

2.2 Nefunkční požadavky

Dále byly přidány tyto nefunkční požadavky:

- použití frameworku Nette při implementaci,
- použití MySQL jako řízení báze dat,
- responzivní web a podpora v současných prohlížečích (Mozilla Firefox, Safari, Google Chrome) zajišťující dobrou přístupnost webu,
- zajištění bezpečnosti dat,
- autentizace a autorizace uživatele, který se do systému přihlásí,
- zajištění rychlosti načtení webu a jeho plynulý chod.

2.3 Existující redakční systémy

Na dnešním trhu existuje mnoho CMS (*Content Management System*) systémů pro správu obsahu. Lze je rozdělit na placené a takzvané Open Source systémy. Open Source znamená, že se jedná o neplacenou verzi s otevřeným zdrojovým kódem na kterém se může každý podílet. Placenou část mohou tvořit dodatečné pluginy a moduly. Placené CMS systémy nebyly zařazeny do výběru, jelikož zadavatel nepočítá s velkým nárůstem nákladů.

Výběr redakčních systémů byl proveden náhodně, ale byly zahrnuty jak známé, tak i méně známé redakční systémy. Byly vybrány čtyři CMS Open Source, které využívají programovacího jazyku PHP. U každého z těchto redakčních systémů bude zhodnoceno, zda je daný redakční systém použitelný pro webovou aplikaci. Na závěr bude provedeno celkové zhodnocení těchto systémů.

2.3.1 WordPress

Redakční systém WordPress je v současnosti velice oblíbeným Open Source CMS systémem a řadí se na první příčky v oblíbenosti u uživatelů. Jeho zdrojový kód je otevřený a každý se na rozvoji systému může podílet. Je napsán ve skriptovacím jazyce PHP a pro ukládání dat a práci s nimi používá databázi MySQL (Kudláček, 2010). WordPress je licencován pod *GNU Public Licence* (GPL) verze 2, obsahující soubor *licence.txt* (Williams, Damstra, Stern, 2013).

WordPress nabízí mnoho rozšíření v podobě pluginů, šablon a widgetů. Tento systém ale nenabízí pluginy, které by obstály v konkrétních požadavcích zadavatele.

Například týkající se kalkulace konkrétních služeb. WordPress nabízí velké množství šablon jak placených, tak i neplacených. Tyto šablony jsou většinou blogového charakteru, ale lze i najít šablony firemního stylu. Velkou nevýhodou je identická šablona na několika webových stránkách, která je demotivující pro oslovení potenciálního klienta. Výsledky odpovídají tomu, že se v podstatě jedná o automaty, které stránky vytvoří na základě vašeho zadání (Polzer, 2012). Vše lze vytvořit i individuálně podle představ uživatele, ale při použití mnoha zásuvných modulů má stránka tendenci k výraznému zpomalení samotného načtení. Problémem mohou být i časté nové verze WordPressu, ty mohou vést k nekompatibilitě nainstalovaných pluginů a následně vedou k nefunkčnosti určitých částí webu. Problém je v tom, že každý plugin je jiný a držet krok s vývojáři, mít přehled co a jak funguje je náročné (Shima, 2015). Většina lidí jen slepě klikne na aktualizaci a ani neřeší jestli někdo jiný s pluginem nemá problém (Shima, 2015). Obrovské množství nainstalovaných pluginů také vede k nepřehlednosti, protože dochází k přibývání ikon v levém panelu uživatelského okna.

Zhodnocení s ohledem na požadavky:

I když je WordPress oblíbený, je vhodnější spíše pro menší projekty s ne tak specifickými požadavky. Vhodný zejména pro osobní blogy. Tento trend patří do efektivního marketingu dnešní doby. Tento systém nabízí mnoho možností, ale představa instalace mnoha pluginů k dosažení určitého výsledku je nevhodná a zbytečná. Důvodem pro vyřazení WordPressu z nabídky je tedy fakt, že tento systém je vnímán spíše jako blogovací platforma, kterou nelze pro tento projekt využít.

2.3.2 Joomla

Joomla se řadí v pořadí jako druhý nejoblíbenější redakční systém. Je napsána v jazyce PHP a využívá databáze MySQL. Na rozdíl od WordPressu není závislá na MySQL, ale lze využít i PostgreSQL, Oracle a další. Provozovat ji lze na webovém serveru s Apache nebo IIS (Bart, 2012). Stejně jako WordPress obsahuje rozšiřující komponenty a různé pluginy.

Výhodou tohoto redakčního systému je větší funkcionálna a společně s WordPressem si udržuje uživatelskou přívětivost. Je sice nutná technická znalost, ale uživatele systém nezatěžuje. S WordPressem má tento systém jinak téměř identické vlastnosti, tudíž za nevýhodu lze označit nedostačující pluginy jak z hlediska zpomalení, tak i kvůli konkrétním požadavkům například na výpočty služeb zadavatele. Je také možné, že v Joomla uživatel narazí na pluginy, které si bude muset zaplatit na rozdíl třeba od WordPressu, kde by je měl zdarma. On-page SEO optimalizace je poměrně složitá, vyžaduje nainstalování různých pluginů (Kvasnicová, 2014).

Zhodnocení s ohledem na požadavky:

Joomla je sice vhodnější, ale je nedostačující z pohledu zadaných funkčních požadavků. Tento redakční systém není tak “user-friendly” jako WordPress a objevují se zde i další problémy. Například horší možnost optimalizace v prohlížečích a těžko-

pádnost systému způsobující zpomalení. Z těchto důvodů byl tento systém vyřazen z výběru.

2.3.3 Drupal

Drupal je modulární framework napsaný ve skriptovacím jazyce PHP, který obsahuje CMS, systém modulů a API pro rychlý vývoj webových stránek a webových aplikací (Redding, 2010). Drupal může běžet na každé počítačové platformě, která podporuje webový server schopný spustit PHP verzi 5.4.5+ (zahrnující Apache, IIS, lighttpd a nginx) a databázi (MySQL, SQLite nebo PostgreSQL) pro ukládání obsahu a nastavení (Tomlinson, 2015).

K tomuto systému jsou nutné určité technické znalosti programovacích jazyků, tudíž pro začátečníky není vhodný vzhledem ke své složitosti. Rozhraní tohoto redakčního systému není uživatelsky přívětivé, i když možnosti, které Drupal nabízí jsou velmi dynamické a dají se velmi dobře využít. Drupal CMS je nový systém pro správu obsahu ve srovnání s jinými starými a již zavedenými systémy pro správu obsahu. Drupal CMS není kompatibilní s jiným softwarem (Karthikeyan, 2016). Webové stránky vytvořené v Drupalu nikdy nelze otevřít s pomalým připojením k internetu (Karthikeyan, 2016). Takže lze hodnotit, že je tento systém tak složitý až těžkopádný a přístupnost webu je omezena.

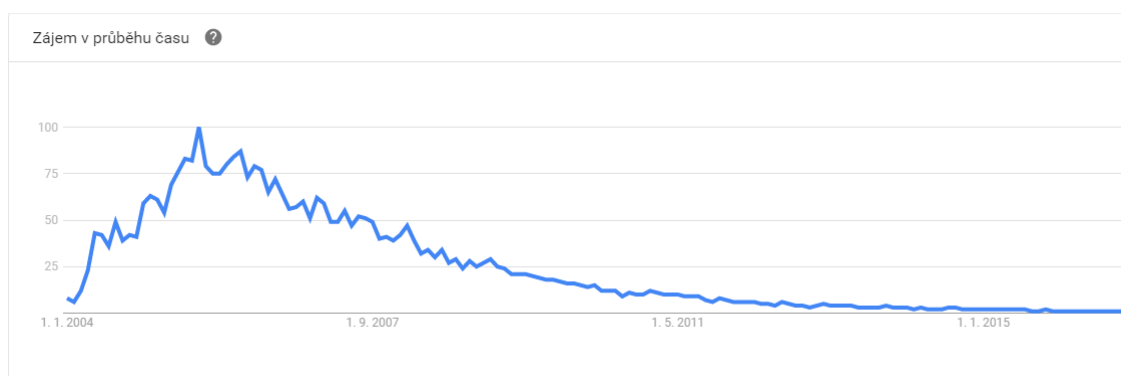
Zhodnocení s ohledem na požadavky:

I když Drupal nabízí největší možnosti z těchto Open Source systémů, nelze tento systém vybrat jako vhodný pro tvorbu zadané práce. Je zbytečně složitý a náročný na výkon. Vhodný spíše pro velké projekty, které nekladou důraz na vzhled. Bylo nutné zohlednit skutečnost, že zadavatel chce mít možnost částečně zasahovat do obsahu. Tudíž je tento systém nevhodný vzhledem ke svému složitému prostředí.

2.3.4 TextPattern

Textpattern je redakční systém napsaný v jazyce PHP a využívá MySQL databáze. Tento systém není obecně tak znám a nemá tak silnou propagaci jako například WordPress. Přesto i tento systém nabízí flexibilitu, rychlost a patrnou volnost především v psaní článků.

Jeho silnou nevýhodou je malé množství pluginů, které se dá vysvětlit menší popularitou u uživatelů. Tato skutečnost byla ověřena pomocí trendové křivky od společnosti Google, kde křivka jasně vykresluje silný propad v užívání tohoto systému (viz obr. 2). Textpattern disponuje velmi strohým a až moc obyčejným rozhraním. Nabídka šablon je také velmi omezená, takže tvorba firemního webu je komplikovanější. I když je tento systém tak jednoduchý a prostý, je nutné mít alespoň určitou znalost v programování. Zápis textů je v Textile místo WYSIWYG editoru (například TinyMCE). Takový zápis je neintuitivní a ocení jej snad jen programátor a ještě málokterý. Existují pluginy pro jiné formy zápisu textu, je však nutné je doinstalovat (Dvořák, 2013).



Obrázek 2: Křivka Google Trends pro Textpattern

Zhodnocení s ohledem na požadavky:

Textpattern je vhodným řešením spíše pro osobní blogy, které nebudou chtít komerčně uspět. Tím je myšleno, že je vhodný pro tzv. psaní do šuplíku. Tento systém byl shledán nezajímavým a je možná i otázkou času, kdy dojde k jeho zániku či ještě silnějšímu propadu. Vytvořit v něm firemní prezentaci by bylo neefektivní a nesplňuje většinu funkčních požadavků na inovaci, proto není vhodným kandidátem na řešení webové aplikace pro Capellen Music Production.

2.4 Hodnocení současných redakčních systémů na trhu

I když každý z jmenovaných redakčních systémů nabízí širokou škálu celkem zajímavých možností, ani jeden nevyhovuje dostatečně funkčním požadavkům zadavatele. Je důležité si také uvědomit, že i když se jedná o Open Source CMS, pak neznamená, že je vše zdarma. Nejedná se o obrovské investice, ale je třeba i na tento problém případně poukázat.

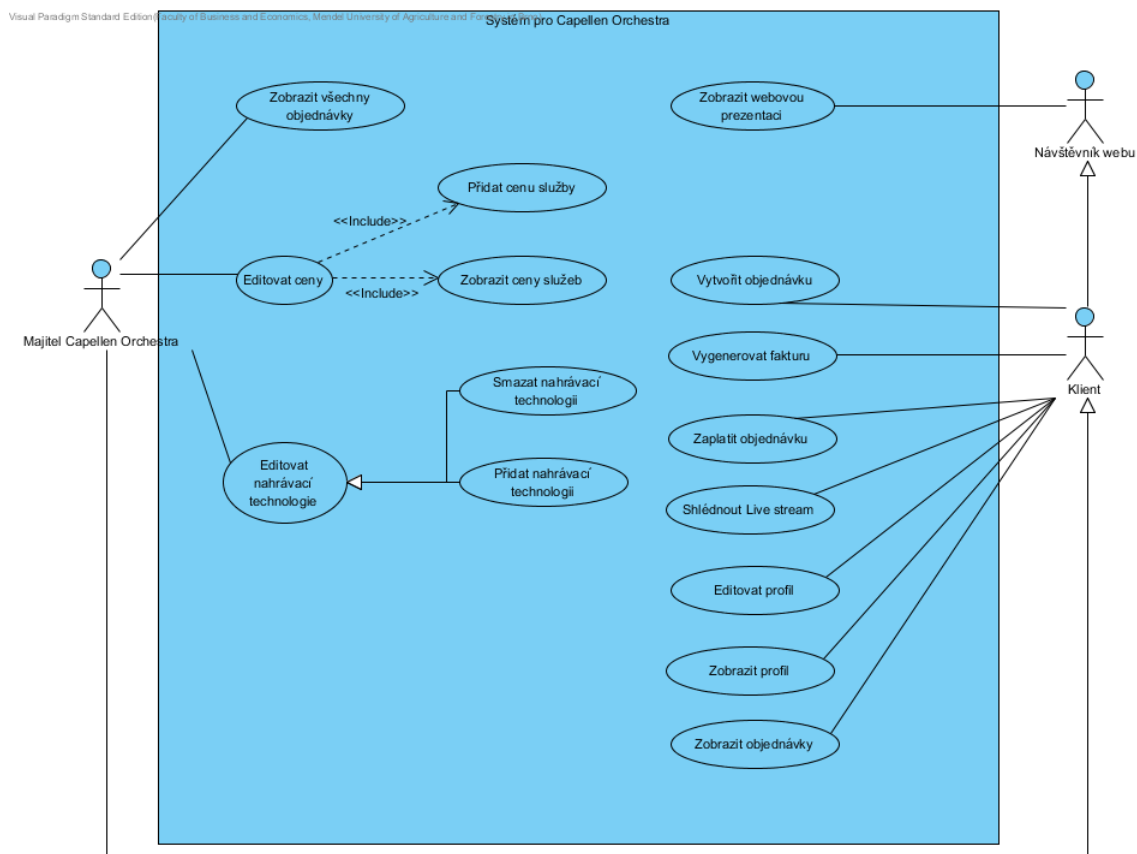
Další věc, která byla u redakčních systémů zjištěna, je skutečnost, že uživatel musí sledovat aktualizace nainstalovaných pluginů a tím pádem i kompatibilitu se systémem. Každá aktualizace má své úskalí a nelze jen odkliknout možnost aktualizace. Celá aplikace by se mohla v průběhu postupně rozsypávat. Některé systémy byly vyhodnoceny jako poměrně složité a náročné na výkon, tudíž lidé s nižším připojením k internetu by neměly šanci se k webu dostat. Poměrně nejvhodnější redakční systém by byl nějaký mezi Joomla a Drupal, ale tato možnost prozatím neexistuje. U každého redakčního systému nastávají konkrétně popsání problémy, které se vzájemně vylučují s požadavky, proto z těchto důvodů nelze doporučit vybrané redakční systémy k řešení nového návrhu aplikace.

3 Návrh vlastního řešení

Na základě analýzy současného stavu webové aplikace Capellen Music Production bylo rozhodnuto, že požadavky zadavatele jsou svým charakterem příliš specifické a nejlepšího možného výsledku lze dosáhnout pouze návrhem aplikace na míru. Tato aplikace by měla maximálně vyhovovat funkčním požadavkům a tím i usnadnit velkou část práce, kterou majitel musí vykonávat.

3.1 Diagram případu užití

Pro celkové pochopení a upřesnění funkcionality webové aplikace bude použit *Use-Case diagram* neboli diagram případu užití. Tento diagram se používá jako první při vytváření informačního systému a dokáže definovat, co bude systém dělat. Patří do seznamu UML (*Unified Modeling Language*), tudíž díky tomuto jazyku lze snadno vyjádřit a vizualizovat systém v podobě Use-Case diagramu. V případě webové aplikace pro Capellen Music Production je diagram případu užití zobrazen na obr. 3.

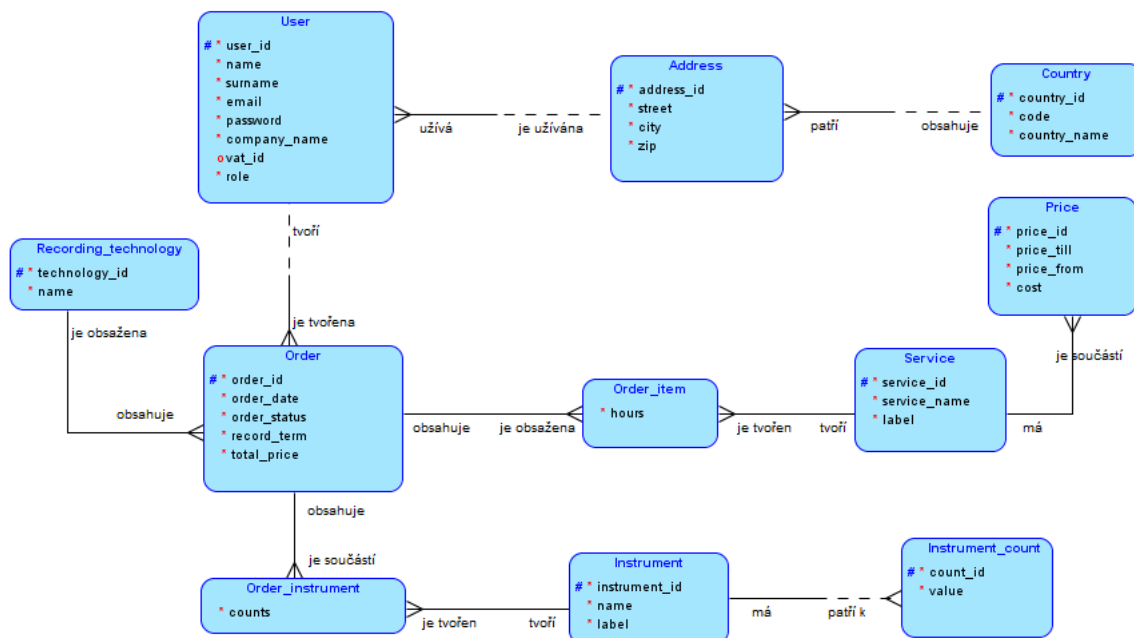


Obrázek 3: Use-Case diagram systému pro Capellen Orchestra

V Use-Case diagramu byli definováni 3 aktoři. Prvním aktorem je *Návštěvník webu*, který představuje náhodného uživatele vstupujícího na veřejně dostupné stránky. Tento aktor si může prohlédnout webovou prezentaci nebo se zaregistrovat pomocí formuláře. Případ užití registrace, přihlášení a možnosti zaslání zapomenutého hesla byly brány již v potaz, tudíž nejsou zaznamenány v diagramu. Druhý aktor s názvem *Klient* může provádět to samé co *Návštěvník webu*, jelikož od něj dědí všechny jeho funkce. *Klient* navíc může například vytvářet objednávky, generovat faktury a další funkce vyjmenované v diagramu. Posledním aktorem je *Majitel Capellen Orchestra*, který dědí od *Klienta*, tudíž může provádět všechny *Klientovy* funkce a navíc má definované své funkce mezi které patří například editace cen, nahrávacích technologií nebo zobrazení všech objednávek zákazníků. Z diagramu je evidentní, že *Majitel Capellen Orchestra* má ze všech aktorů nejvíce funkcionalit, které může využívat.

3.2 Návrh databázového schématu

Dalším krokem pro vytvoření webové aplikace byl návrh databáze. K tomuto kroku bylo nutné se opakovaně vracet, jelikož schéma postupem času při vývoji potřebovalo dodatečné úpravy či zaznamenat nové skutečnosti, které musely být do schématu přidány dodatečně. Pro znázornění vztahu mezi tabulkami byl vytvořen logický model databáze. Model databáze si lze prohlédnout na obr. 4. Výsledný fyzický model databáze lze shlédnout na obr. 15.



Obrázek 4: Logický model databáze

3.2.1 Popis jednotlivých entit

- **Country**

Slouží zejména při vytváření nového profilu uživatele nebo při jeho editaci. Tato entita má v sobě uložené všechny názvy zemí a jejich zkratky, tudíž u každého uživatele lze identifikovat, ze které země pochází.
- **Address**

Entita *Address* ukládá všechny adresy uživatelů. Každá adresa má svůj primární klíč s názvem `address_id`, `street`, `city`, `zip`.
- **User**

Entita *User* slouží k evidenci zaregistrovaných uživatelů. Pomocí této entity lze rozlišit role, které jsou podstatné pro udělení pravomocí, které může člověk v systému dělat. User obsahuje povinné atributy `user_id`, `name`, `surname`, `email`, `password`, `company_name`, `role` a nepovinné `vat_id`. Heslo je šifrováno pomocí hashovací funkce `sha1`, a jelikož nebylo možné získat původní hesla uživatelů, musí si je při prvním navštívení nové aplikace změnit pomocí funkce zapomenutého hesla.
- **Order**

Order slouží k ukládání základních informací o objednávce. Obsahuje primární klíč `order_id`. Dále je zde definováno `order_date`, `order_status`, `record_term` a `total_price`. Tato entita podává informace o tom, kdy dojde k nahrávání orchestru a kdo objednávku vytvořil. Lze zjistit celkovou cenu služby, kdy byla objednávka vytvořena a v jakém stavu se objednávka nachází. Při vytvoření objednávky je tento atribut nastaven na stav “unprocessed”. Uživatel může vytvořit objednávek jednu nebo i více. Jejich počet není ničím omezen.
- **Recording technology**

Entita *Recording technology* udává informace o nahrávacích technologiích, které budou využity při nahrávání orchestru. Tato entita obsahuje pouze atribut `name` a primární klíč `technology_id`. Recording technology je využito pouze při vytváření objednávky, kde si klient vybere ze šesti způsobů nahrávání ve formuláři. Musí být vybrána pouze jedna z těchto možností.
- **Order item**

Entita *Order item* ukládá informace o počtu hodin objednaných pro nahrávání orchestru pod atributem `hours`. Dále informuje o tom, jaké služby byly objednány.
- **Service**

Service ukládá názvy nabízených dostupných služeb. Entita obsahuje atributy `name`, `label` a primární klíč `service_id`. Ve formuláři pro objednávku jsou zveřejněny pouze některé služby, zbytek je zasazen uvnitř vzorce a jsou automaticky započítány.

- **Price**

Entita *Price* udává informace o cenách nabízených služeb, které jsou definovány v tabulce *Service*. Obsahuje atributy *cost*, *price_till*, *price_from*, primární klíč *price_id*. Atributy *price_till* a *price_from* byly do této tabulky zasazeny z důvodu, aby se dala vygenerovat zpětně faktura z historických cen. Zkoumá se tedy, zda datum faktury spadá do *price_till* a *price_from*. S cenami manipuluje administrátor, který má možnost ve svém účtu přidat nebo mazat ceny vybraných služeb.

- **Order instrument**

Ke zpětnému zjištění, co si klient objednal za hudební nástroj a kolik si jich objednal, slouží entita *Order instrument*. Do atributu *counts* se ukládá počet objednaných hudebních nástrojů. Lze říci, že lze najít tyto hodnoty v tabulce *Instrument count*, ale sloupec *counts* byl vytvořen zejména kvůli zajištění historických dat pro počet objednaných hudebních nástrojů.

- **Instrument**

Instrument má v sobě uložené informace o hudebních nástrojích, které si lze objednat pomocí formuláře pro vytvoření objednávky. Entita obsahuje atributy *name*, *label* a primární klíč *instrument_id*. Atribut *label* byl vytvořen pouze z důvodu lepšího označení při vytváření popisků ve formuláři.

- **Instrument count**

Entita *Instrument count* ukládá počty hudebních nástrojů, které si lze navolit ve formuláři pro objednávku. Jelikož majitel nabízí jen určité množství hudebníků hrající na daný nástroj a jejich počty nejdou hned za sebou v dané posloupnosti a jedná se o konkrétní náhodné hodnoty, pak bylo nutné si tyto hodnoty uložit zvlášť do tabulky. Například pro hudební nástroj *Violin II* lze zvolit 0, 6, 4, 8 nebo 10 muzikantů. U každého nástroje jsou hodnoty rozdílné a někde lze zvolit pouze 0 nebo 1 muzikanta. Každý hudební nástroj může mít tedy více hodnot.

3.3 Zajištění historických dat

Důležitou část systému tvoří i zajištění historických dat při zpětné generaci faktury. Pokud by systém neřešil tuto problematiku, mohlo by dojít k problému, který by hodně zkreslil a znepríjemnil situaci jak zákazníkovi, tak majiteli společnosti. Pro příklad by mohla být uvedena situace, kdy zákazník si například po roce chce vygenerovat ze systému znovu svou fakturu. Během tohoto roku si ale majitel společnosti přepočítal ceny, protože zjistil, že jej ovlivňuje například inflace, konkurence a další ekonomické faktory. Musel tedy přistoupit ke změnám cen určitých služeb, které nabízí.

V případě, že by systém nebyl schopen zajistit uchování historických dat, pak by zákazník v okamžiku nového vygenerování zjistil, že takovou částku neplatil, jelikož nebyla o tolik vyšší. Je možné, že by si žádné změny nevšiml a vygeneroval

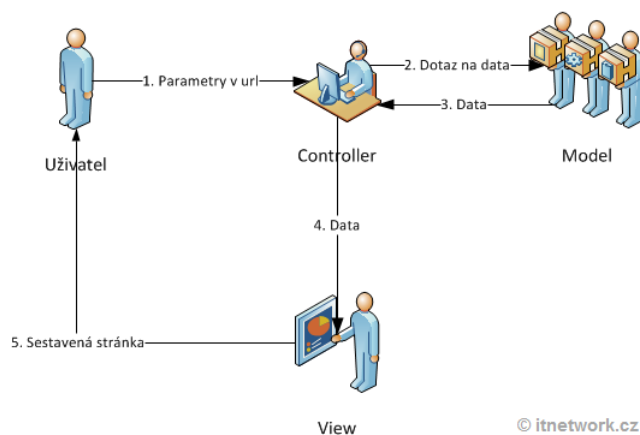
by fakturu znovu, ale dochází tu ke zkreslení skutečných informací a mohlo by to vést k zásadnímu problému. A jak by si zákazník mohl ověřit pravost dat na faktuře? Nejspíše pouze tím, že si částku zkontroluje pomocí svého bankovního účtu.

Aby tato situace nikdy nenastala, byly do tabulky *price* zařazeny dva atributy, které se nazývají *price_from* a *price_till*. Tyto údaje uvádějí platnost ceny (od, do). Pro každou službu je tedy přiřazena cena, která má zadanou platnost. Řešení této skutečnosti spočívá v tom, že je sestaven jednoduchý SQL dotaz, pomocí kterého lze zjistit, zda datum objednávky spadá do zadaného rozsahu platnosti cen. Pokud tedy datum objednávky spadá do platnosti cen od/do, pak se vygenerují tomu příslušné ceny, které byly stanoveny v tomto období. Tudíž zákazník se po roce může vrátit a vygenerovat si fakturu s částkou, která byla v minulém roce aktuální a není v ní obsažena žádná změna cenové politiky majitele, která nastala během roku. Samozřejmě, že při nové objednávce bude tato změna ovlivňovat cenu objednaných služeb.

3.4 MVC a MVP architektura

Architektura MVC odděluje prvky systému a umožňuje jejich nezávislou změnu (Sommerville, 2013). Tento architektonický vzor se stal oblíbeným pro složitější weby, a proto je využíván mnoha frameworky, mezi které patří například i framework Nette. Vzor MVC dělí aplikaci na 3 části, a to na *Model*, *View* a *Controller*.

Komponenta *Model* spravuje systémová data a přidružené operace s těmito daty (Sommerville, 2013). Lze tedy konstatovat, že tato část architektonického vzoru je jádrem logiky aplikace a spravuje data. *View* prezentuje data a zobrazuje výstup uživateli ve vhodné podobě. Tato podoba je dána většinou šablonou v HTML a CSS. Třetí část tvoří komponenta *Controller*, která přijímá podněty od uživatele a promítá změny do komponenty *View* a *Model*. Celková logika MVC architektury je zachycena na obr. 5 (ITNetwork, 2016). Pokud je vše dodrženo, pak je výsledkem přehledný kód s určitou logikou.



Obrázek 5: Princip MVC architektury

Architektura MVP (*Model, View, Presenter*) je více charakteristická pro framework Nette. Presenter má téměř identickou roli jako *Controller* v architektuře MVC. Vybírá pohled (View) a předává mu model, nebo data z modelu. Udržuje stav persistentních proměnných. A především zpracovává reakce uživatele. (Grundl, 2009)

3.5 Návrh uživatelské struktury aplikace

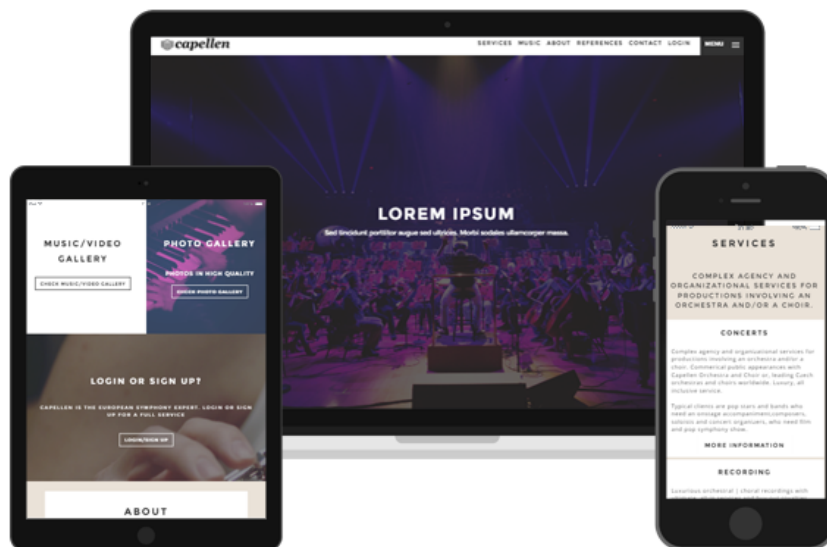
V analýze současného stavu aplikace byla aplikace shledána jako vizuálně neatraktivní, proto součástí nového řešení je i návrh kompletního designu aplikace. Aplikace rozlišuje pohled uživatele a administrátora. V obou případech je vzhled rozhraní řešení jednoduše a efektivně. Aplikace je navržena tak, že při otevření hlavní stránky se otevře webová prezentace společnosti, kde uživatel v hlavní navigaci nalezne tlačítko k přihlášení. Tím se dostane k přihlašovacímu nebo registračnímu formuláři. Má i možnost si nechat zaslat nové heslo v případě jeho ztráty. V celé aplikaci včetně hlavní stránky je brán velký důraz na moderní, plochý, čistý vzhled a responzivitu, která umožňuje větší přístupnost webu.

3.5.1 Hlavní stránka

Při otevření hlavní stránky se uživateli zobrazí webová prezentace společnosti, která je postavena jako rolovací stránka, tudíž uživatel projíždí za pomoci scrollingu mezi sekcemi. Sekce mají podobu pruhů zarovnaných pod sebe. Hlavní stránka je koncipována tak, aby neobsahovala příliš textu a návštěvník webu či potenciální klient byl informován o hlavních nabízených službách co nejrychleji.

Záhlaví hlavní stránky disponuje responzivní horizontální navigací, která slouží k rolování po daných sekcích. V levém horním rohu navigace je vloženo logo společnosti, které bylo převzato z původní stránky. V průběhu jednání se zadavatelem došlo i ke konzultaci nového loga, které bylo vytvořeno určenou osobou, ale výsledek se neosvědčil jako dostačující, tudíž se zadavatel přiklonil ke starší přehlednější verzi. Pro větší dynamičnost byl zvolen do záhlaví slider, ve kterém se mění obrázky v časovém intervalu. Následně uživatel prochází sekcemi, které lze rozkliknout do větších podrobností. Na konci stránky se uživatel může jednoduše dostat zpět na začátek stránky pomocí tlačítka na pravé straně, které se zobrazuje pouze při najetí do sekcí. Tato možnost usnadňuje orientaci ve stránce.

Webová prezentace společnosti by měla splňovat kufrový test, který slouží k tomu, aby byla definována jednoznačná navigace a uživatel se orientoval na stránce co nejrychleji a nejefektivněji. Pokud se na určitou podstránku podíváme, jednoznačně vidíme označení stránky, loga, na které stránce se nyní nacházíme a co na ní můžeme dělat. Vyhledávání nebylo použito, jelikož nebylo vzhledem k množství informací nutné. Na jedné z podstránek je definována i galerie, která dokáže načíst videa z YouTube. Finální proměnu hlavní stránky lze shlédnout na obr.6

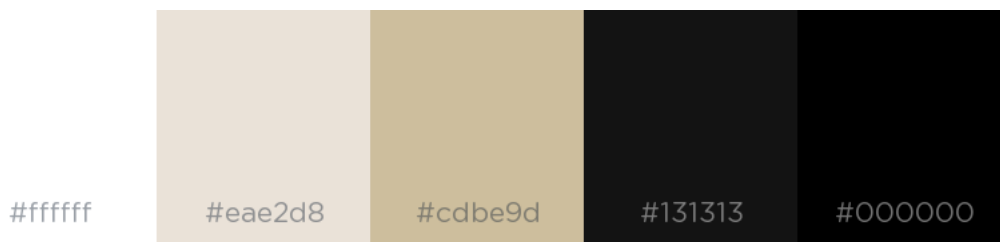


Obrázek 6: Nová webová prezentace

3.5.2 Barevné schéma a typografie

Po několika konzultacích se zadavatelem bylo rozhodnuto, že barevné schéma celé aplikace bude složeno z béžové, hnědé, tmavě šedé, bílé a černé. Barevnost stránky doplňují i obrázky, které jsou v tlumeném podtónu nebo obsahují fialovou barvu symbolizující nahrávací budovu ve Zlíně, která má interiér ve fialové barvě. Barvy byly vybrány tak, aby působily neagresivně, klidně, dostatečně seriózně a umělecky. Základní barevné schéma je možné vidět v následujícím obr. 7.

Během konzultací bylo také rozhodnuto, že stránka bude pouze v anglickém jazyce z důvodu pouze zahraniční klientely. K této skutečnosti bylo pro nadpisovou část zvoleno proporcionální bezpatkové písmo *Montserrat* bez rozšíření. Pro textovou část bylo zvoleno bezpatkové písmo *Open Sans*. Tento typ písma byl zvolen zejména z důvodů, kdyby mělo dojít k přeložení do českého či jiného jazyka s diakritikou.



Obrázek 7: Použité barevné schéma

3.5.3 Uživatelské rozhraní

Po přihlášení má uživatel možnost vytvořit objednávku, zaplatit ji pomocí PayPal účtu, shlédnout všechny objednávky (viz obr. 8) a ke všem si vygenerovat fakturu i z historických dat nebo může editovat údaje o své osobě ve svém profilu a odhlásit se. Má k dispozici i streamovací okno, kde může sledovat živý přenos z nahrávání koncertů v kongresové centra ve Zlíně. Může si také během přihlášení znovu prohlédnout webovou prezentaci firmy a následně se zpětně dostat do svého profilu.

Po vytvoření objednávky je uživatel o zpracování objednávky informován pomocí e-mailu. Pokud uživatel zapomene heslo, může si vygenerovat heslo nové, které mu bude zasláno na jeho e-mailovou adresu. Uživatelské rozhraní je identické s administrativním, akorát obsahuje méně funkcí. Tento fakt byl zjištěn na základě Use-Case diagramu, kde jsou definovány funkce pro konkrétní aktory. Je nutné počítat i s faktem, že obyčejný návštěvník webu je potenciálním klientem, který se bude chtít stát uživatelem. Pro tuto skutečnost byl vytvořen registrační formulář, kdy po jeho vyplnění je uživateli zaslán e-mail s přihlašovacími údaji.



The screenshot shows a web interface titled "MY ORDERS" with a table listing six orders. Each row contains the user's name, order ID, order date, status, record term, total price in CZK, an invoice generator icon, and a PayPal payment method icon.

NAME	ORDER ID	ORDER DATE	ORDER STATUS	RECORD TERM	TOTAL PRICE (CZK)	INVOICE GENERATOR	PAYMENT METHOD
marie doruskova	772	23.08.2016	unprocessed	07.08.2016	55243.76		
marie doruskova	882	25.08.2016	unprocessed	01.08.2016	104447.2		
marie doruskova	902	25.08.2016	unprocessed	01.08.2016	172640		
marie doruskova	912	06.09.2016	unprocessed	24.09.2016	92366.56		
marie doruskova	922	21.09.2016	unprocessed	30.09.2016	24601.72		
marie doruskova	932	21.09.2016	unprocessed	30.09.2016	74245.6		

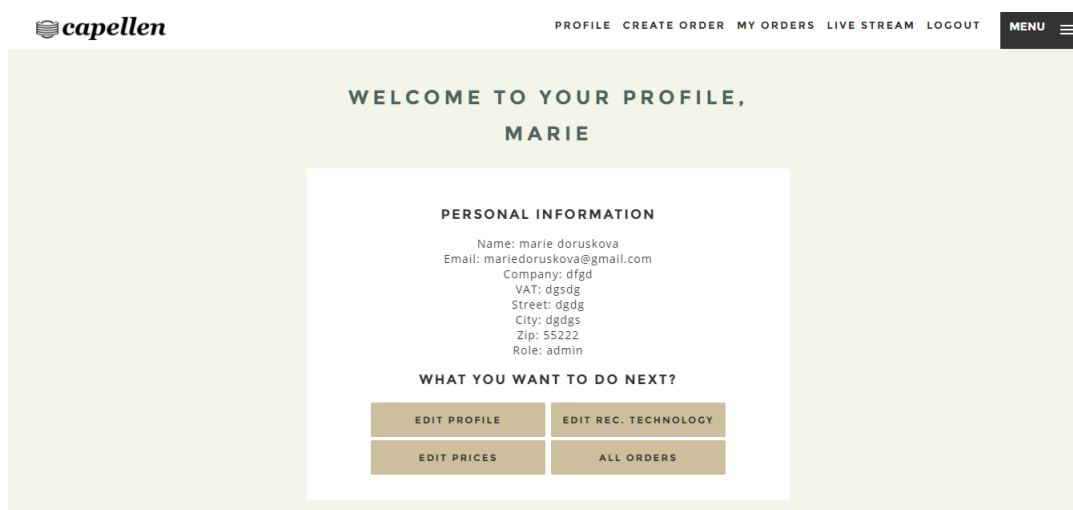
Obrázek 8: Přehled objednávek uživatele

3.5.4 Administrativní rozhraní

Administrátor má ve svém profilu navíc několik možností na rozdíl od běžného uživatele. Administrátor má možnost navíc editovat ceny dostupných služeb. Po rozkliknutí této možnosti si administrátor může vybrat službu a přidat k ní novou cenu. Musí uvést také dobu platnosti dané ceny z důvodu, aby bylo možné generovat

fakturu s historickými daty. Dále může editovat nahrávací technologie a zobrazovat všechny objednávky, které byly provedeny všemi zákazníky. Při zobrazení všech objednávek lze vygenerovat faktury, které neobsahují cenu každé služby, nýbrž pouze celkovou cenu objednávky. Tyto informace o cenách jsou pro majitele firmy důvěrné a jsou zahrnuty ve vzorci pro kalkulaci služeb při objednávce.

Vzhled rozhraní byl vytvořen tak, aby se administrátor nebo uživatel mohli rychle orientovat na stránce. Rozhraní je sestaveno pouze z horizontálního menu, kde na levé straně je zobrazeno logo společnosti a na pravé straně navigace jsou položky s odkazy na různé typy podstránky příslušící administrátorovi. Pod touto částí je přivítání, blok s informacemi o profilu a výběr nejdůležitějších funkcí profilu (viz obr. 9). V této části se tedy zobrazuje hlavní obsah. Pod každým obsahem je sekce s kontaktem na firmu Capellen Music Production. Tato sekce byla vyžádána zadavatelem.



Obrázek 9: Rozhraní administrátora

4 Implementace

4.1 Použité technologie

4.1.1 Nette Framework

Framework Nette slouží pro tvorbu webových aplikací a v současné době je jeho vývoj veden pod *Nette Foundation*. Tento framework je vybaven tzv. laděnkou *Tracy*, která dokáže chyby logovat, rychle je odhalit a vývojář je tedy schopen snadno danou chybu opravit. Dále využívá šablonovacího systému *Latte*, který nabízí širokou škálu možností pro zpříjemnění práce vývojáři (Nette, 2016).

Nette využívají zejména tuzemské společnosti, kterými je například *uloz.to*, *Mladá fronta nebo Slevomat* a mnoho dalších. Za nevýhodu lze považovat nedostatečnou dokumentaci, která obsahuje pár příkladů a někdy vývojář narazí i na chyby. Věc, která vnahrazuje tento nedostatek, je dobrá komunikace na fóru tohoto frameworku. Dochází i k pravidelným setkáním všech fanoušků programování webových aplikací. Této akci se říká Poslední sobota. Nette je na svém vzestupu a dokazuje to i 3. umístění v anketě magazínu *SitePoint* za „Best PHP Framework for 2015“ (Nette, 2016). Pro vývoj webové aplikace pro Capellen Music Production byla použita verze 2.3.5.

4.1.2 MySQL

MySQL je open source relační databázový systém typu DBMS (*Database Management System*) založený na SQL (*Structured Query Language*). Podporuje instalaci na všechny platformy a lze pomocí MySQL vytvořit širokou škálu aplikací. Technologie, která se obvykle kombinuje pro vývoj webových aplikací a MySQL je její součástí, se nazývá LAMP (*Linux, Apache, MySQL, PHP*) (Taušová, 2014). Tento typ databáze byl zvolen zejména z těchto důvodů:

- je zdarma a veden jako opensource,
- jednoduchost instalace a užívání,
- kompatibilní s jinými platformami,
- rychlost a široká použitelnost.

4.1.3 Nette Componette: Addons, plugins and components

Framework Nette disponuje možností rozšířit funkcionalitu webové aplikace pomocí open source projektu zvaném *Componette*. Tato stránka nabízí mnoho rozšíření pro řešení webových aplikací a usnadňuje vývojáři tvorbu aplikace. Je zde přehledně popsáno, co daný plugin dělá, kolik lidí si ho stáhlo, jaké má nevyřešené problémy. Obsahuje také návod na instalaci většinou prostřednictvím *Composeru* a informuje

o následném zprovoznění komponenty pomocí presenteru. Webová aplikace potřebovala využít několik komponent, kterými jsou: *MetisFW/PayPal*, *Kdyby/Autowired*, *Joseki/PdfResponse*, *Nextras/Forms*.

Po vyzkoušení několika komponent, které se v průběhu zprovoznění PayPal platební brány projeví jako závadné, byla vybrána komponenta PayPal od MetisFW z důvodu velkého počtu stáhnutí a také kvůli nulovým otevřeným problémům s touto komponentou. Tyto skutečnosti se potvrdily a mohl být zprovozněn online platební portál pro zaplacení objednávky. Pro funkčnost tohoto portálu bylo třeba ještě komponenty *Kdyby/Autowired*, jelikož *MetisFW/PayPal* předpokládá automatické užití továrny. Při nepoužití dochází k chybnému typu parametru kvůli *Dependency injection*. Tato továrna byla aplikována v BasePresenteru.

Při výběru tvorby faktury byla využita komponenta *Eciovni*, která se projevila jako velmi nevhodná pro tvorbu PDF výstupu, protože nelze libovolně vložit do šablony co potřebujete a musíte se řídit předepsanou šablonou, proměnnými a bez velkého zásahu do vnitřního kódu komponenty není vývojář schopen vytvořit vlastní fakturu. Pro generaci faktury bylo tedy využito komponenty *Joseki/PdfResponse*, která umožňuje výstup ve formátu PDF. Faktura je vytvořena díky šablonovacímu systému Latte, stylu CSS a HTML.

Poslední rozšíření, které bylo nainstalováno pomocí Composeru náleželo *Nextras/Forms*, protože pro webovou aplikaci bylo třeba vytvořit rozklikávací kalendář tzv. *DatePicker* pro výběr data nahrávání v objednávkovém formuláři a následně i ve výběru platnosti ceny v rozhraní administrátora.

4.1.4 Použité grafické prvky

Pro vytvoření hlavní webové prezentace bylo při vývoji využito několika grafických prvků a implementací, které byly převzaty od jiných tvůrců. Tyto prvky byly zejména v JavaScriptu, CSS a HTML.

Responzivní menu bylo vytvořeno na základě volně dostupného návodu a souborů ze stránky *CodyHouse* (Romano, 2014). Pro slider se použila knihovna *Slick*, která zajišťuje plynulý přechod mezi obrázky v záhlaví (Slick, 2014). Na konci stránky dochází i k přechodu textu, tam byl použit jQuery plugin *Owl Carousel* (Owl Carousel, 2013), protože lépe dokázal zobrazit přechod různých textů. Multimediální galerie zajišťuje zobrazení responzivního vyskakovacího okna, které čte obsah videí YouTube a Vimeo. Tato funkce byla zajištěna pomocí pluginu *Magnific Popup* (Semenov, 2014). Pro implementaci bylo nutné využití JavaScriptu a je tedy důležité uvědomit si, že zdroj webové stránky ve skutečnosti nezobrazuje nový element, který byl vložen, ale pouze ukazuje odkaz na soubor JavaScriptu (Ruppert, Moore, Duckett, 2014).

Webová stránka je responzivní pomocí media query v CSS šabloně. Media query dává CSS souboru logické podmínky na základě vlastností výřezu prohlížeče a může vykreslit odlišnosti v závislosti na vlastnostech okna prohlížeče (LaGrone, 2013).

4.2 Vývojové prostředí

4.2.1 WampServer

WampServer je vývojové prostředí pro *Windows* vedené pod open-source *GPL* licenci, pomocí kterého lze vytvořit vlastní lokální server na svém počítači. Vývojář může díky tomuto prostředí vytvářet jednoduše webové aplikace za pomoci *Apache serveru*, *jazyku PHP* a databázového serveru *MySQL* (WampServer, 2016).

Pro účely webové aplikace bylo využito serveru Apache s podporou jazyka PHP a byla vytvořena databáze v MySQL. PHP nabízí programátorům báječnou paletu nástrojů pro zpracování vstupních a výstupních operací týkajících se souborového systému (Gilmore, 2011). K databázi se přistupuje pomocí phpMyAdminu, který je napsán v jazyce PHP a podporuje širokou škálu operací na MySQL a MariaDB (PhpMyAdmin, 2016). Veškeré operace s tabulkami databáze byly provedeny pomocí tohoto nástroje, jak už v rámci WampServeru tak mimo tento server, když byla aplikace napojena na *IBM Bluemix*.

4.2.2 NetBeans

NetBeans je integrované open-source vývojové prostředí (IDE), které umožňuje vývojáři lehce a rychle tvořit aplikace zejména v *Javě*, *ale i v C/C++*, *Groovy*, *PHP*, *HTML5*, *CSS*, *JavaScriptu*. Pomocí NetBeans lze vytvořit jak desktopové, tak mobilní či webové aplikace (NetBeans, 2016). Mezi jeho vlastnosti patří například oprava chyb zvýrazněním syntaxe a nabídnutí možnosti vyřešení dané chyby nebo automatické doplnění kódu, které urychluje vývojáři práci. NetBeans disponuje i možností stáhnout si různá rozšíření (pluginy). Pro vývoj webové aplikace bylo využito NetBeans IDE 8.0.2.

4.2.3 IBM Bluemix

Bluemix je komplexní inovativní cloudová platforma pro vývoj, provoz a správu všech typů aplikací. Je založen na standardech *Cloud Foundry*, *Docker* a *OpenStack*. Vývojář si může při tvorbě projektu vybrat z několika programovacích jazyků a přistupovat k němu přes browser, příkazovou řádku (CLI) nebo přímo z vývojových nástrojů (IDE, např. Eclipse) (IBM, 2016). V případě webové aplikace pro Capellen Orchestra byl vybrán PHP projekt a k nahrávání souborů docházelo nejdříve pomocí CLI, ale jelikož je tato příkazová řádka poměrně nepohodlná a příkazy zdlouhavé, došlo k připojení verzovacího nástroje Git. Následně pomocí základních příkazů pro Git byly soubory přidány či aktualizovány. Ke konci byla připojena k projektu hotová databáze využívající phpMyAdmin. Tato služba byla využívána i jako platforma pro nasazení aplikace, ale proběhl zde i vývoj aplikace.

Během vývoje a nasazení aplikace na tuto službu bylo zjištěno pár skutečností, které nakonec vedly k rozhodnutí upustit od této platformy. Důvody jsou následující:

- časté padání této služby a nedostupnost služby v odpoledních hodinách, někdy i dopoledních,
- pomalé nahrávání souborů,
- častá aktualizace přístupových klíčů omezuje dobu užívání.

4.3 Vybrané funkce systému

4.3.1 Generace faktury

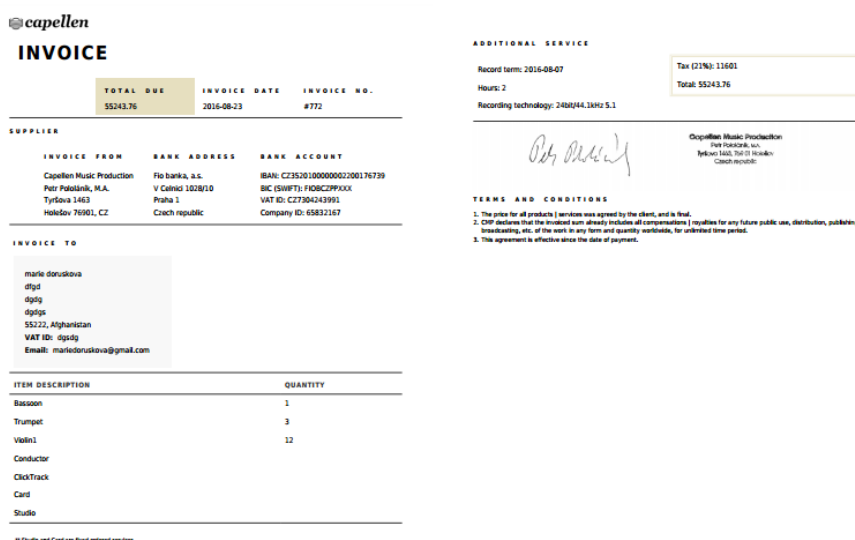
Při vytvoření objednávky má uživatel možnost vygenerovat si fakturu, na které má veškeré údaje o objednané službě. Jelikož zadavatel nechce zobrazovat jednotlivé ceny služeb z důvodů konkurence, jsou ve faktuře vypsány pouze jednotlivé položky, jejich množství, konečná cena a další doprovodné informace, které do faktury náležitě patří (viz obr. 10).

Generaci zajišťuje funkce `actionGenerate($id)` a implementace je následující:

```
$template = $this->createTemplate();
$template->setFile( dirname(__DIR__) . '/presenters/templates
/Eciovni/generate.latte ');

//zde se ukladaji polozky z~databaze do $template

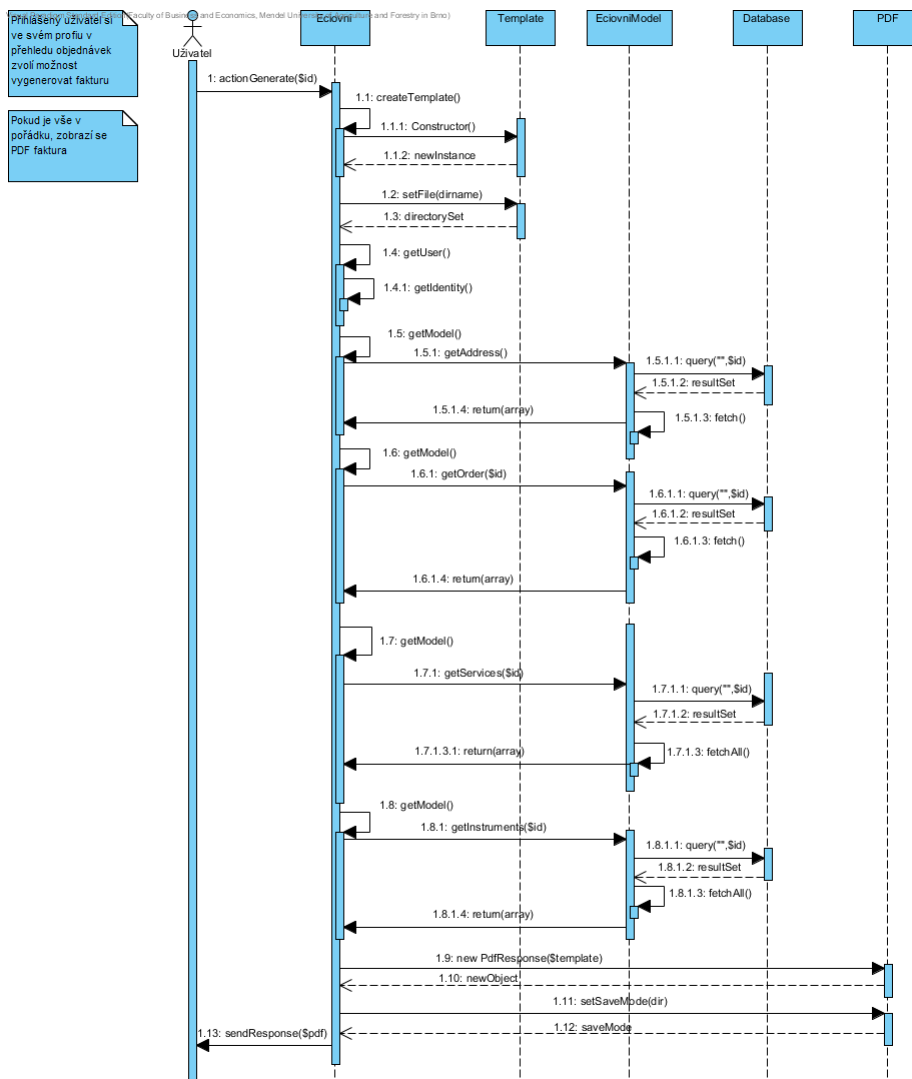
$pdf = new \Joseki\Application\Responses\PdfResponse($template);
$pdf->setSaveMode(\Joseki\Application\Responses\PdfResponse::INLINE);
$this->sendResponse($pdf);
```



Obrázek 10: Vzhled faktury

Sekvenční diagram generování faktur

Sekvenční modely jsou dynamické modely, které pro každý režim interakce popisují posloupnost interakcí, k nimž dochází mezi objekty (Sommerville, 2013). Pro případ užití generování faktury byl tedy vytvořen sekvenční diagram, který je znázorněn na obrázku (viz obr. 11). Tento diagram ukazuje již detailní posloupnost funkcí, které byly postupně implementovány a upřesňovány.



Obrázek 11: Sekvenční diagram pro generaci faktury

4.3.2 Online platba faktury

Zákazník po vytvoření objednávky má možnost fakturu vygenerovat a zaplatit online pomocí platební brány PayPal. Tato brána funguje díky komponentě *MetisFW/PayPal* a *Kdyby/Autowired*. K objednávkám a jejich placení se zákazník může dostat

i zpětně ve svém profilu v záložce My orders, pokud chce objednávku zaplatit v pozdějším termínu. Po vybrání možnosti zaplatit fakturu online v přehledu objednávek, je uživatel znovu tázán, zda chce opravdu zaplatit pomocí online PayPal brány. Následně je přesměrován na stránku PayPalu, kde vidí částku, kterou má zaplatit a má možnost přihlásit se do svého účtu PayPal nebo zaplatit pomocí debetní či kreditní karty spojenou s revolvingovým úvěrem. Tato skutečnost je zachycena na obrázku obr. 12. Aby došlo k platbě na účet majitele společnosti, musely být zjištěny pomocí developerského portálu PayPalu dva klíče tzv. *OAuth keys (ClientID a Secret)*, které toto propojení účtu zajišťují.

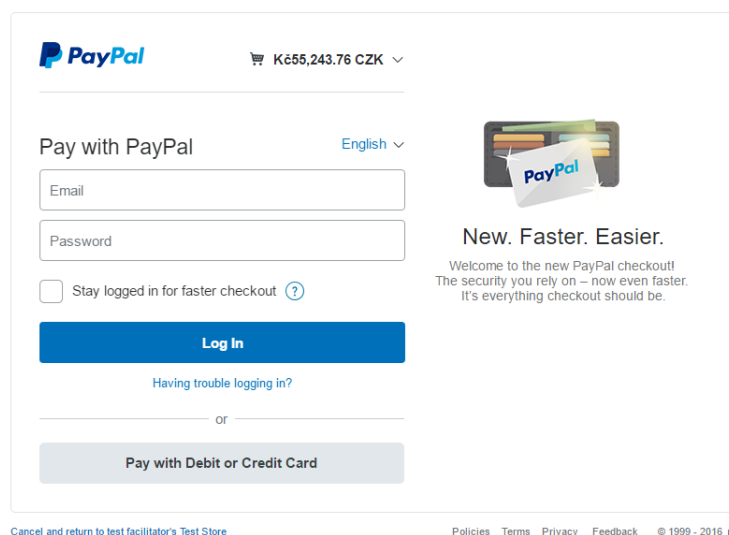
Níže je uvedena krátká ukáзка implementace tlačítka pro PayPal pomocí funkce `createComponentPayPalPaymentButton(SimplePaymentOperationFactory $factory)`.

```
$price = $this->getModel()->getOrder($this->getOrderId());
$operation = $factory->create('Orchestration', $price['total_price']);
$control = new PaymentControl($operation);

$control->onCheckout[] = function(PaymentControl $control, Payment $created){
};

$control->onSuccess[] = function(PaymentControl $control, Payment $paid){
};

$control->onCancel[] = function(PaymentControl $control){
};
return $control;
```



Obrázek 12: Online možnost platby za pomoci PayPal

Služba PayPal byla zadavatelem vyžádána zejména z důvodu, že se jedná o pohodlnou, rychlou a celosvětovou službu. A jelikož jeho klientela pochází spíše ze zahraničí (zejména Čína), pak je PayPal nejlepší způsob pro provedení transakce. Co se týče bezpečnosti této služby, pak lze říci, že stejně jako u každé bankovní společnosti existuje určité riziko některého z kyberútoků. Riziko může uživatel co nejvíce snížit zejména silným heslem, neklikáním na podvodné e-maily nebo nepropůjčováním účtu cizím lidem.

4.3.3 LiveStream

Součástí webové aplikace měl být i LiveStream, který by živě na webu vysílal nahrávání z kongresového centra ve Zlíně. Tuto funkci zajišťovala firma LIVEBOX, a.s. Po složitém vyžádání všech informací od této společnosti byla tato firma vyhodnocena jako nevhodná neboť má zjevné problémy v oblasti komunikace se zákazníkem a dodáním dostatečných materiálů ke zprovoznění funkcionality streamu. Při komunikaci dochází k naprosté ignoraci požadavků zadavatele, mylnému zasílání e-mailů a k velice dlouhým časovým prodlevám. Do dnešní doby nebyl odpovídající člověk schopen odepsat na příslušné problémy streamu, tudíž nelze zjistit, zda funguje či nikoliv.

Živé vysílání funguje za pomoci funkce *actionStream()*, kde implementace je následující.

```
$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, "http://api.smartcams.cz/v1/capellen.js");
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLOPT_POSTFIELDS,
"ver=1.0&key=zde_je_uveden_klic&video_id=1&ip=89.185.244.200&
nonce=1478966042&sig=zde_je_podepisujici_klic");
```

Společnost podala údaje o klíči, podepisujícím klíči a jejich SMS kódů. Tyto informace vloženy do připraveného API od společnosti a následně došlo k vygenerování klíče a podepisujícího klíče. Tyto výsledné klíče byly vloženy do funkce *curl_setopt* za třetí parametr do pozice *key* a *sig*. Ostatní informace byly také z tohoto API získány. Pro zobrazení přehrávače streamu bylo využito HTML a JavaScriptu, kde v jedné z funkcí bylo definováno číslo kamery. Tyto informace byly získány z dokumentace společnosti.

4.4 Diagram tříd

Pro znázornění struktury systému byl vytvořen tzv. *Class Diagram* neboli diagram tříd pro modely a presentery. Class Diagram popisuje atributy a operace třídy a také omezení uložená v systému. Tyto diagramy tříd jsou široce používány v modelování objektově orientovaných systémů, protože jsou jedinými UML diagramy, které mohou být mapovány přímo s objektově orientované jazyky (Tutorialspoint, 2016).

V každém presenteru a modelu jsou vyznačeny operace, které byly třeba k implementaci aplikace. Všechny presentery se nachází v namespace `App\Presenters` a dědí z `BasePresenteru`, který dědí ze třídy `Presenter`, která se nachází v namespace `Nette\Application\UI`. Modely byly následně napojeny k presenterům pomocí agregace a byl jim přidělen vlastní package pro přehlednost a systematičnost. `AclModel` je již obsažen v `BasePresenteru`, proto je spojen agregací k `BasePresenteru`. `UserManager` je volán pomocí *Dependency injection* a dědí z třídy `Object`, která se nachází v namespace `Nette`. Diagram tříd modelů a presenterů `Nette` je k dispozici v příloze viz obr. 16.

4.5 Bezpečnost

4.5.1 Autorizace na základě rolí

`Nette` pro zabezpečení aplikací vytvořilo třídu zvanou *Permission*, která poskytuje přístup uživatelů a jejich práva v podobě ACL (*Access Control List*) modelu. Vývojář tak může nadefinovat role (např. `guest`, `member`, `admin`), zdroje a vyjmenovat jaké pravomoce jsou jednotlivým rolím uděleny. Pomocí tohoto modelu lze udělovat či zakazovat přístupy k jednotlivým zdrojům dat uložených v aplikaci. Krátká ukázka kódu, kde se přidělují pravomoce uživateli a administrátorovi, je zobrazena níže.

```
$this->acl->allow('member', Permission::ALL);
$this->acl->deny('member', 'admin', 'editService');
$this->acl->deny('member', 'admin', 'editPrices');
$this->acl->deny('member', 'admin', 'viewOrders');
$this->acl->allow('admin', Permission::ALL);
```

Z kódu je zřejmé, že administrátor má přístup ke všemu, co je ve stránce obsaženo. Z těchto důvodů, místo vyjmenovávání jednotlivých presenterů, byla vybrána možnost `Permission::ALL`. Běžný uživatel má zakázaný přístup do šablon `editService`, `editPrices` a `viewOrders`. Tato část se nachází v presenteru `Admin`. Co se týče návštěvníka webu, ten není sice v kódu obsažen, ale má nejvíce omezené možnosti přístupu. Hlavní funkce umístěná v `BasePresenteru`, která rozhoduje o připuštění do systému se jmenuje `isAllowed()`.

4.5.2 Bezpečnostní rizika

- **Cross-Site Scripting (XSS)**

Je metodou narušení webových stránek tím způsobem, že útočník podstrčí svůj vlastní kód a může stránku pozměnit nebo získat citlivé informace o návštěvnicích. Framework `Nette` tuto hrozbu řeší automatickým ošetřením výstupů pomocí technologie *Context-Aware Escaping* (`Nette Foundation`, 2016). Útočník může do webové stránky vložit svůj HTML kód nebo svůj Javascript a tím přeměrovat uživatele na infikovanou stránku.

Spuštěním daného JavaScriptu se umožňuje útočníkovi provést například *krádež cookie*, *keylogging* nebo *phishing*. Lze rozlišit několik typů XSS, a to zejména

Persistent XSS, Reflected XSS nebo DOM-based XSS. U Persistent XSS škodlivý řetězec pochází z databáze webových stránek. Původ škodlivého řetězce u Reflected XSS pochází z požadavku oběti a u DOM-based XSS pochází zranitelnost kódu ze strany klienta (Kallin, Valbuena, 2016).

- **Cross-Site Request Forgery (CSRF)**

Je útokem, který přiměje uživatele navštívit stránku, která skrytě vykonává útok na webovou aplikaci, kde je uživatel zrovna přihlášen. Proti tomuto útoku se lze bránit generováním a ověřováním autorizačního tokenu. Pro úplnou ochranu formuláře stačí zadat příkaz `$form->addProtection()`; (Nette Foundation, 2016).

Pro příklad je uvedeno chatovací fórum, kde útočník zašle zprávu, která obsahuje grafickou značku nebo HTML element obrázku. Nicméně zdroj obrázku obsahuje odkaz, který provádí akci na bankovním účtu oběti. Takže místo souboru s obrázkem byl zaslán odkaz, který provádí bankovní transakci (Acunetix, 2016).

- **SQL Injection**

Je útok, ve kterém je do neošetřených vstupů aplikace vložen nebo aplikován pozměněný SQL dotaz útočníka, který se následně zpracuje a provede (Clarke, 2012). Tímto chováním dosahuje útočník vlastního prospěchu a může získat i citlivá data o návštěvnicích. Proti tomuto útoku se lze předem ochránit zejména tak, že se nevloží proměnná přímo do SQL dotazu. Místo proměnné se vloží symbol otazníku, který poté bude nahrazen odpovídajícím parametrem. Pro snadnější pochopení je níže připravena ukázka nejdříve špatného, a pak správného přístupu.

Špatný přístup

```
$stmt = $this->db->query("SELECT * FROM users JOIN addresses USING (address_id) WHERE user_id = $id);
```

Správný přístup

```
$stmt = $this->db->query("SELECT * FROM users JOIN addresses USING (address_id) WHERE user_id = ?", $id);
```

5 Testování

V běžném vývoji aplikace dochází po implementaci fáze testování. Tato fáze je důležitá zejména proto, aby se předešlo chybám v ostrém provozu. I když byla webová aplikace v průběhu svého vývoje testována po každé naimplementované funkci, může i tak dojít k drobným chybám. Tento fakt může narušit ucelenost systému a působit pro uživatele nedůvěryhodně. Je tedy vhodné celou aplikaci před nasazením do provozu důkladně otestovat a případné chyby včas napravit.

5.1 Uživatelské testování

Aplikace byla otestována samotným vývojářem, kde z prvních úkonů byla provedena registrace související s následnou kontrolou správnosti uložení dat do databáze. Pak došlo k přihlášení do systému, kde byla vybrána možnost editace profilu. Tyto změny byly provedeny správně. Dále byla vyzkoušena stěžejní funkce týkající se tvorby objednávky pomocí dostupného formuláře. Po dokončení vyplnění tohoto formuláře došlo k přesměrování na stránku, která dává uživateli dvě možnosti, a to vygenerovat fakturu a případně ji zaplatit ihned pomocí online platební brány PayPal.

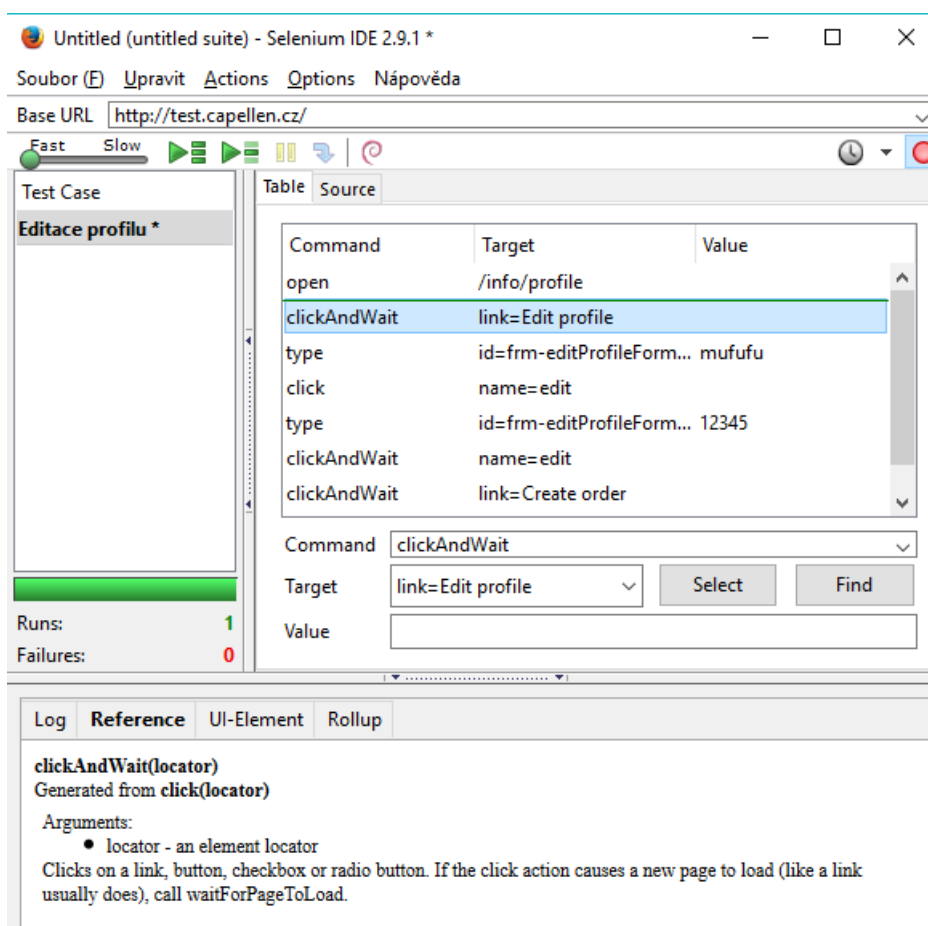
Byla vybrána možnost generace faktury, kde v okně prohlížeče se objeví faktura s údaji o objednávce. V průběhu byl kontrolován i osobní e-mail, kde přicházely zprávy týkající se registračních údajů nebo objednávky. Mezi poslední krok patřilo testování platby PayPalu z přehledu objednávek. Během prohlížení a přesměrování na stránku PayPal nebyly zjištěny žádné problémy. V administrativním profilu bylo nutné otestovat funkci editaci ceny a nahrávacích technologií. Obě funkce nevykazují chybu, která by vedla k nefunkčnosti a nepoužitelnosti systému. Testování streamingu nemohlo proběhnout vzhledem k nekomunikaci se společností zprostředkující tuto službu. Tento problém musí být vyřešen samotným zadavatelem. Následně byla aplikace předložena samotnému zadavateli, aby byl schopen aplikaci otestovat a ohodnotit její funkce. Kromě úprav v textu byl s aplikací spokojen.

5.2 Selenium IDE

Jedná se o integrované vývojové prostředí pro Selenium skripty. Je implementováno jako doplněk pro prohlížeč Mozilla. *Selenium IDE* zahrnuje celé Selenium jádro, což umožňuje rychle nahrávat a přehrávat testy ve skutečném prostředí, ve kterém se bude uživatel pohybovat. S podporou automatického doplňování a schopnosti rychle se pohybovat mezi příkazy, je Selenium IDE ideálním prostředím pro tvorbu Seleniumových testů. Doplněk je napsán v jazyce JavaScript a lze si s ním usnadnit práci při testování aplikace. (Selenium, 2016) Na obrázku 13 je znázorněna ukázka pro seleniumový test u editace profilu. Mezi výhody Selenium IDE patří:

- snadné nahrávání a přehrávání,
- selekce inteligentního pole užívá ID, jména nebo XPath podle potřeby,

- ukládá testy jako HTML, Ruby skripty nebo jako další formáty,
- podpora pro Selenium soubor user-extensions.js.



Obrázek 13: Ukázka Selenium testu

6 Nasazení

Mezi poslední krok pro úspěšné dokončení vývoje webové aplikace, je její reálné nasazení do určitého prostředí. V průběhu vývoje byla aplikace nasazena na platformu Bluemix, aby zadavatel viděl pokroky v tvorbě a případně zdůraznil nedostatky, které pozoruje. Tato platforma byla využita ale i jako vývojové prostředí, které bylo s těžejší. Pomocí využití příkazového řádku Git byla aplikace postupně aktualizována.

6.1 Výsledné prostředí

Zadavatel si během vývoje aplikace zaplatil webhosting a doménu, na kterou si přál aplikaci přenést zejména z důvodů samostatnosti a nezávislosti. Pro nasazení aplikace do provozu si majitel společnosti zaplatil hosting a doménu u společnosti *Relaxhosting.cz*. Tato služba poskytuje tyto služby:

- neomezená velikost dat,
- neomezený počet FTP a FTPS účtů,
- denní záloha FTP, zálohy dostupné 7 dní zpětně s přímým přístupem k zálohám přes FTP,
- neomezená velikost databáze. (Relaxhosting, 2016)

Podstatnou nevýhodou poskytované služby je použití zastaralejší verze PHP 5.3.8, což způsobilo i problémy s nasazením nové aplikace na tento prostor. Tento fakt nyní způsobuje i velké zpomalení aplikace na tomto prostoru z důvodu zastaralého PHP. Databáze pro tuto aplikaci byla zatím přesunuta na prostor PHPAdmin u Bluemixu, jelikož zadavatel nenastavil databázi na webhostingu. Do budoucna by měla databáze fungovat pomocí PHPAdmin na tomto prostoru. Tuto službu si zadavatel vybral sám, tudíž bylo nutné se této situaci přizpůsobit. Aplikaci lze nalézt na adrese: <http://test.capellen.cz/>

7 Závěr

Pro společnost Capellen Music Production byla vyhotovena webová aplikace, která splňuje všechny funkční a nefunkční požadavky, které byly probrány se zadavatelem během konzultací. Zadavateli by tato aplikace měla pomoci při propagaci svých služeb a podpořit tak zadavatelovo podnikání. Vzhled současné stránky odpovídá současným trendům v oblasti webového designu a prostředí pro uživatele je jasné a přehledné.

Uživatel, který se na stránku zaregistruje a přihlásí do svého účtu, by měl být schopen pomocí aplikace vytvořit objednávku prostřednictvím jednoduchého formuláře, případně si vytisknout fakturu ve formátu PDF a následně ji proplatit přes bankovní účet či využít svého účtu PayPal. Mezi další úkony pro běžné uživatele aplikace může patřit editace svého profilu při případné změně svých osobních údajů nebo vstup do přehledu všech objednávek, kde si lze vygenerovat fakturu zpětně i z historických údajů o cenách služeb zadavatele. Pro administrátora jsou specifické určité funkcionality, například editace cen nebo přehled všech objednávek uživatelů.

Problém, který by mohl při využívání aplikace vzniknout, je možná nesoučinnost již zavedených zahraničních klientů. Jelikož z bezpečnostních důvodů nebylo možné udržet jejich dosavadní hesla, muselo být přistoupeno k variantě, kdy takový klient si musí nechat vytvořit nové heslo. Je pouze tedy otázkou, zda všichni budou ochotni si toto heslo pomocí funkce zapomenutého hesla změnit, a tak využít i služby zadavatele. Tato skutečnost je pouze spekulativní a je možné, že tento problém vůbec nenastane. Je ale důležité uvažovat i nad takovou možností. I když je tato aplikace natolik specifická, že jí nelze použít pro jiné společnosti, může se v budoucnosti dále rozšiřovat o další funkční bloky a zdokonalovat je. Jedná se například o přehled objednávek, kde lze vytvořit v budoucnosti větší administrativní pravomoc týkající se například přidělením přístupu k streamu konkrétním uživatelům nebo přehledu zasílaných vyhotovených souborů pro zákazníky. Aplikace by se dala dále rozšířit o archiv veškeré audiovizuální tvorby, která by tvořila tzv. organizér tvorby pro zadavatele. Co se týče tvorby, mohla by zde být i externí fotogalerie napojená na webovou stránku. Tato funkce by značně ulehčila organizaci všech fotografií z různých koncertů a událostí. Dále by se dalo vytvořit určité diskuzní fórum pro dirigenty a další personál, aby došlo k rozšíření komunikace. Toto fórum či chatovací okno by mohlo být zavedeno u stream přenosu. Tato možnost již byla diskutována, ale služba poskytující stream nebyla dostatečně schopná a ochotná poskytnout takovou službu, tudíž by muselo dojít k samostatné implementaci nebo ke změně poskytovatele streamu.

8 Reference

- ACUNETIX. *CSRF Attacks, XSRF or Sea-Surf – What They Are and How to Defend Against Them* [online]. [cit. 2015-12-23]. Dostupné z: <http://www.acunetix.com/websitesecurity/csrf-attacks/>.
- BART. *Co je Joomla?* [online]. 2012 [cit. 2015-12-05]. Dostupné z: <http://www.joomlaportal.cz/dotazy/co-je-joomla>.
- CLARKE, J. *SQL injection attacks and defense*. [online]. Waltham, Mass.: Elsevier, 2012. 576 s. ISBN 9781597499736.
- DVOŘÁK, T. *Textpattern: zapomenutý redakční systém?* [online]. 2013 [cit. 2016-10-09]. Dostupné z: <https://www.tomas-dvorak.cz/posts/textpattern-zapomenuty-redakcni-system/>.
- GILMORE, W. *Velká kniha PHP 5 a MySQL: kompendium znalostí pro začátečníky i profesionály*. Brno: Zoner Press, 2011. 736 s. ISBN 978-80-7413-163-9.
- GRUNDL, D. *Nette Framework: MVC & MVP* [online]. 2009 [cit. 2016-12-11]. Dostupné z: <https://www.zdrojak.cz/clanky/nette-framework-mvc-mvp/>.
- IBM *PaaS - IBM Bluemix* [online]. [cit. 2016-11-07]. Dostupné z: <http://www-05.ibm.com/cz/cloud/web.html>.
- ITNETWORK *MVC architektura* [online]. [cit. 2016-11-01]. Dostupné z: <http://www.itnetwork.cz/navrhove-vzory/mvc-architektura-navrhovy-vzor>.
- KALLIN, J., VALBUENA L. I. *Part Two: XSS Attacks* [online]. [cit. 2016-12-23]. Dostupné z: <https://excess-xss.com/#xss-attacks>.
- KARTHIKEYAN *Advantages And Disadvantages Of Using Drupal CMS For Your Website* [online]. 2016 [cit. 2016-10-05]. Dostupné z: <http://www.digisecrets.com/web/advantages-and-disadvantages-of-using-drupal-cms-for-your-website/>.
- KUDLÁČEK, L. *WordPress: podrobný průvodce tvorbou a správou webů*. Brno: Computer Press, 2010. 264 s. ISBN 978-80-251-2734-6.
- KVASNICOVÁ, J.. *Open Source – ano nebo ne – 2.část* [online]. 2014 [cit. 2016-10-05]. Dostupné z: <http://blog.byznysweb.cz/2014/04/open-source-ano-nebo-ne-2-cast/>.
- LAGRONE, B. *HTML5 and CSS3 responsive web design cookbook: learn the secrets of developing responsive websites capable of interfacing with today's mobile Internet devices*. Birmingham: Packt Pub., 2013. 204 s. ISBN 978-1-84969-544-2.
- NETBEANS *NetBeans IDE - Overview* [online]. [cit. 2016-11-01]. Dostupné z: <https://netbeans.org/features/index.html>.

- NETTE FOUNDATION *Nette* [online]. [cit. 2016-11-02]. Dostupné z: <https://nette.org/cs/>.
- NETTE FOUNDATION *Zabezpečení před zranitelnostmi* [online]. [cit. 2016-11-02]. Dostupné z: <https://doc.nette.org/cs/2.4/vulnerability-protection>.
- OWL CAROUSEL *Owl Carousel* [online]. 2013 [cit. 2016-11-02]. Dostupné z: <http://owlgraphic.com/owlcarousel/>.
- PHPMYADMIN *PhpMyAdmin* [online]. [cit. 2016-11-01]. Dostupné z: <https://www.phpmyadmin.net/>.
- POLZER, J. *Vytvořte si vlastní webové stránky bez znalosti programování* [online]. 2012 [cit. 2015-12-05]. Dostupné z: <http://www.cnews.cz/vytvorte-si-vlastni-webove-stranky-bez-znalosti-programovani>.
- REDDING, J. *Beginning Drupal*. Indianapolis, Ind.: Wiley Pub., Inc., 2010. 484 s. ISBN 978-0-470-43852-7.
- RELAXHOSTING *Parametry a podrobná specifikace webhostingu* [online]. [cit. 2016-12-22]. Dostupné z: <https://www.relaxhosting.cz/neomezeny-webhosting.php#section-3>.
- ROMANO, C. *Secondary Expandable Navigation. CodyHouse - Free HTML, CSS, JS nuggets* [online]. 2014 [cit. 2016-11-02]. Dostupné z: <https://codyhouse.co/gem/secondary-expandable-navigation/>.
- RUPPERT G., MOORE J., DUCKETT, J. *JavaScript & jQuery: interactive front-end web development..* Indianapolis, IN: Wiley, 2014. 640 s. ISBN 978-1-118-53164-8.
- SELENIUM *Selenium IDE Plugins* [online]. [cit. 2016-12-11]. Dostupné z: <http://www.seleniumhq.org/projects/ide/>.
- SEMENOV, D. *Magnific Popup* [online]. 2014 [cit. 2016-11-02]. Dostupné z: <http://dimsemenov.com/plugins/magnific-popup/>.
- SHIMA *Největší nevýhody WordPress zvláště pro důležité weby* [online]. 2015 [cit. 2016-10-05]. Dostupné z: <http://n-host.cz/2015/10/nejvetsi-nevyhody-wordpress-zvlaste-pro-dulezite-weby>.
- SLICK *Slick - the last carousel you'll ever need* [online]. 2014 [cit. 2016-11-02]. Dostupné z: <http://kenwheeler.github.io/slick/>.
- SOMMERVILLE, I. *Softwarové inženýrství*. Brno: Computer Press, 2013. 680 s. ISBN 978-80-251-3826-7.
- TAUŠOVÁ, V. *MySQL* [online]. 2014 [cit. 2016-12-27]. Dostupné z: <http://www.pslib.cz/filip.dittrich/Viki/mysql.html>.

- TOMLINSON, T. *Beginning Drupal 8*. Berkeley: Apress, 2015. 287 s. ISBN 978-1-4302-6581-8.
- TUTORIALSPPOINT *UML - Class Diagram* [online]. [cit. 2016-12-26]. Dostupné z: https://www.tutorialspoint.com/uml/uml_class_diagram.htm.
- WAMPSEVER *WampServer* [online]. [cit. 2016-11-01]. Dostupné z: <http://www.wampserver.com/en/>.
- WILLIAMS, B., DAMSTRA, D., STERN, H. *Professional WordPress: Design and Development*. Second edition. Indianapolis, Ind: Wiley Pub., Inc., 2013. 456 s. ISBN 978-1-118-44227-2.

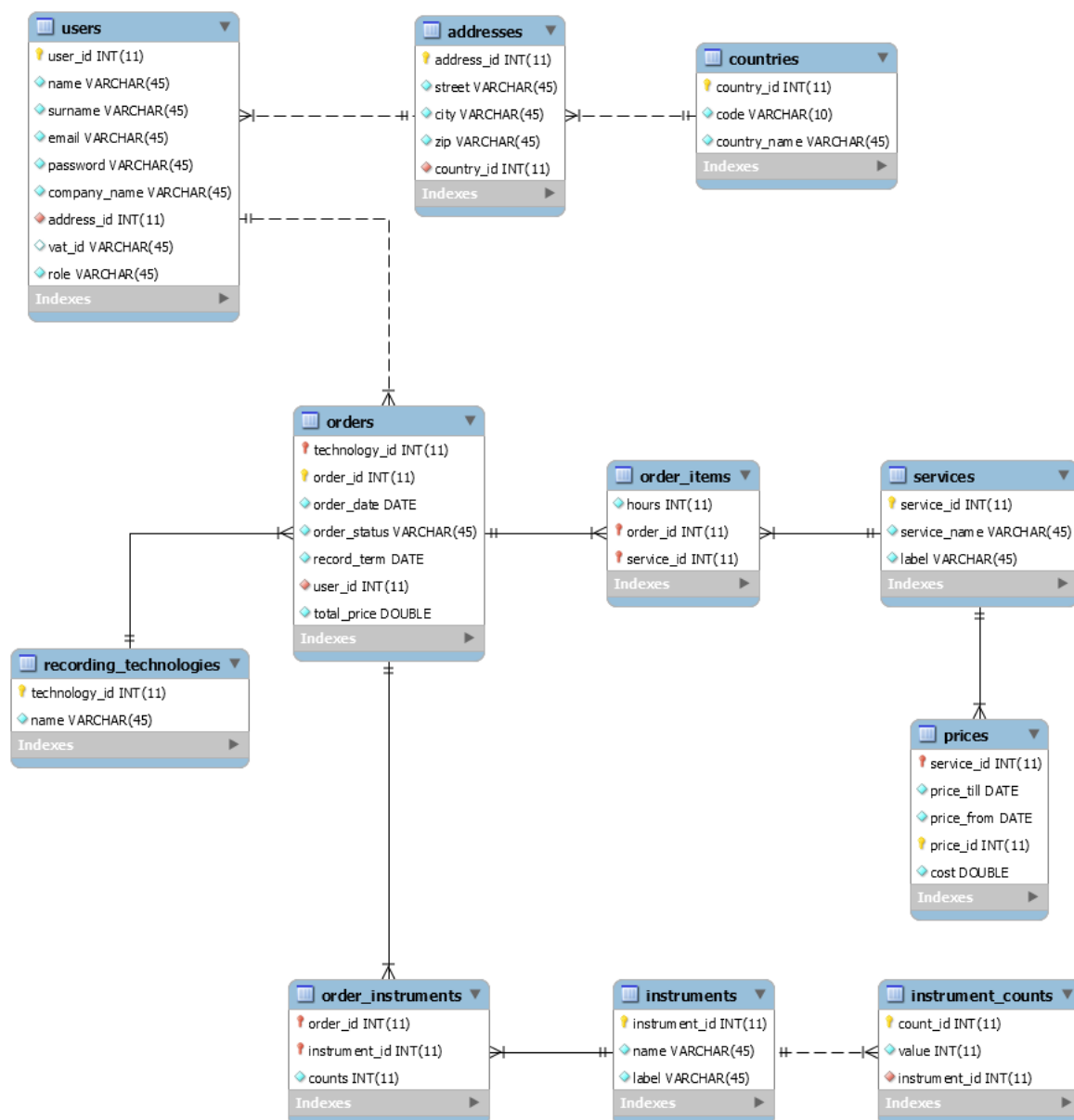
Přílohy

A Původní systém



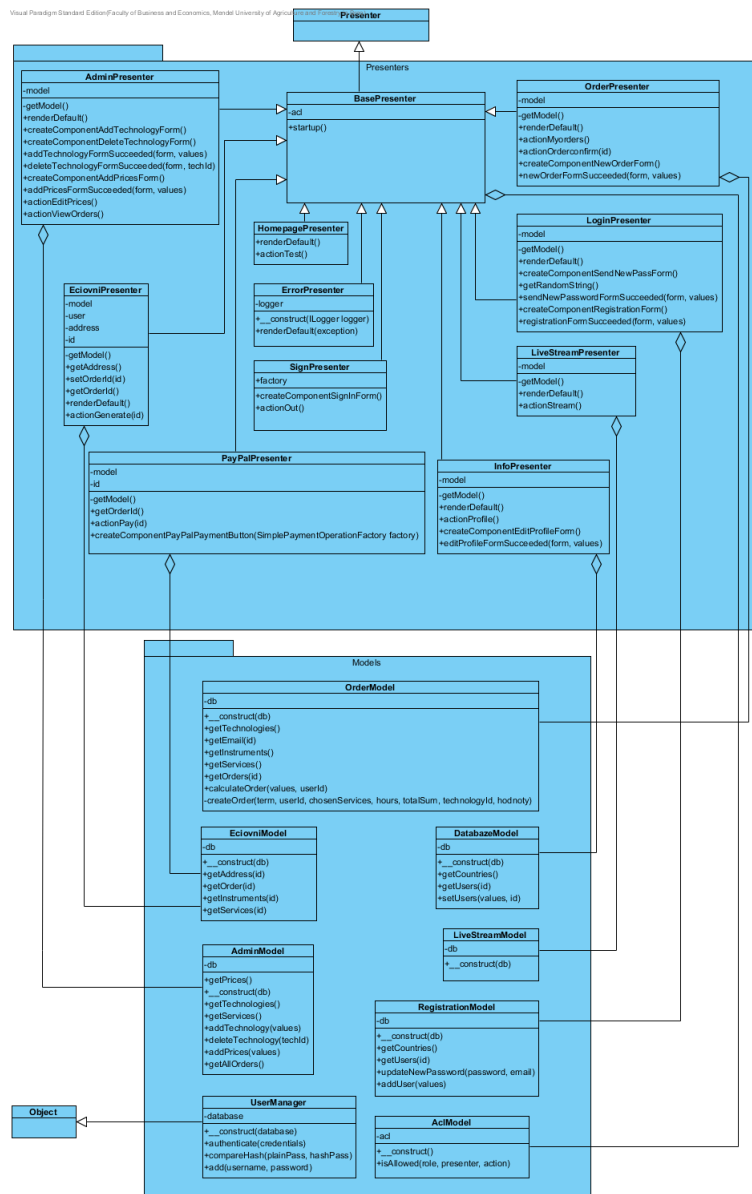
Obrázek 14: Původní vzhled aplikace

B Fyzický model



Obrázek 15: Fyzický model nové databáze

C Diagram tříd



Obrázek 16: Diagram tříd modelů a presenterů