

Webová aplikace pro nabídku a poptávku krátkodobého pronájmu

Josef Obrtlík

Bakalářská práce
2019



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
akademický rok: 2018/2019

ZADÁNÍ BAKALÁŘSKÉ PRÁCE (PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Josef Obrtlík**
Osobní číslo: **A16026**
Studijní program: **B3902 Inženýrská informatika**
Studijní obor: **Informační technologie v administrativě**
Forma studia: **prezenční**

Téma práce: **Webová aplikace pro nabídku a poptávku krátkodobého pronájmu**
Téma anglicky: **Web Application for Supply and Demand Short-term Lease**

Zásady pro vypracování:

1. Provedte rešerši stávajících řešení.
2. Vypracujte stručný rozbor technologií, které budou použity k návrhu webové aplikace.
3. Provedte rozbor a analýzu požadavků na zvolené řešení.
4. Popište strukturu a princip frameworku Symfony.
5. Realizujte navrženou webovou aplikaci pro nabídku a poptávku krátkodobého pronájmu bytů/domů ve vlastnictví soukromých osob.
6. Věnujte pozornost zabezpečení webové aplikace.



Rozsah bakalářské práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. WELLING, Luke a Laura THOMSON. *Mistrovství PHP a MySQL*. Brno: Computer Press, 2017, 799 s. ISBN 978-80-251-4892-1.
2. SKLAR, David. *PHP 7: praktický průvodce nejrozšířenějším skriptovacím jazykem pro web*. Brno: Zoner Press, 2018, 368 s. Encyklopedie Zoner Press. ISBN 978-80-7413-363-3.
3. NIXON, Robin. *Learning PHP, MySQL & JavaScript: with jQuery, CSS & HTML5. Fourth edition*. Beijing: O'Reilly, 2014, xxvii, 780. ISBN 978-1-491-91866-1.
4. BANCER, Wojciech. *Symfony2 Essentials*. UK: Packt Publishing Limited, 2015. ISBN 978-1-784-39876-7.
5. SALEHI, Sohail. *Mastering Symfony: orchestrate the design, development, testing, and deployment of web applications with Symfony*. Birmingham: Packt Publishing, 2016. *Community experience distilled*. ISBN 978-1-784-39426-4.

Vedoucí bakalářské práce: **Ing. Petr Navrátil, Ph.D.**

Ústav řízení procesů

Datum zadání bakalářské práce: **30. listopadu 2018**

Termín odevzdání bakalářské práce: **15. května 2019**

Ve Zlíně dne 7. prosince 2018

doc. Mgr. Milan Adámek, Ph.D.
děkan



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

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 22. května 2019

Josef Obrtlík, v. r.

ABSTRAKT

Cílem bakalářské práce je vytvoření funkční webové aplikace pro nabídku a poptávku krátkodobého ubytování. V teoretické části jsou představeny podobné stávající řešení, dále věci potřebné ke správné implementaci projektu, a také všechny technologie, se kterými se v projektu pracuje. Poté je popsán Framework Symfony a to jeho princip, popis zpracování webového požadavku, struktura, práce se vzhledem aplikace, práce s databází a nástroj, který poskytuje k ladění aplikace. V praktické části je popsáno funkční řešení dané aplikace.

Klíčová slova:

Symfony, PHP, HTML, CSS, Javascript, ORM, MVC, Git

ABSTRACT

Main goal of this bachelor thesis is to create functional web application for supply and demand of short-term lease. In theoretical part there are presented similar existing solutions, then there are things needed for correct implementation of the project and also all the technologies that the project works with. Then the Framework Symfony is described, namely its principle, description of web request processing, its structure, how Symfony works with the view of the application, also how it works with a database and a tool that is provided by Symfony to debug the application. Practical part describes functional solution of this application.

Keywords:

Symfony, PHP, HTML, CSS, Javascript, ORM, MVC, Git

Rád bych poděkoval vedoucímu své bakalářské práce panu Ing. Petru Navrátilovi, Ph.D. za odborné vedení a konzultace.

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

OBSAH

ÚVOD	10
I TEORETICKÁ ČÁST	11
1 O APLIKACI	12
1.1 PODOBNE APLIKACE	12
1.1.1 Airbnb.cz	12
1.1.2 LevneUbytovani.net	14
1.1.3 Ubytovani.net	15
1.1.4 Booking.com	17
2 POTŘEBY PROJEKTU WEBOVÉ APLIKACE	19
2.1 DOMÉNA	19
2.2 HOSTING	19
2.3 WEBOVÝ SERVER NGINX.....	20
2.3.1 Jak NGINX pracuje?	20
2.4 COMPOSER	20
2.5 CMDER	21
2.6 SYSTEM SPRÁVY VERZÍ – GIT	22
2.6.1 GitLab	22
2.6.2 GitKraken.....	23
3 POUŽITÉ TECHNOLOGIE	24
3.1 HTML.....	24
3.2 CSS.....	24
3.3 PHP.....	25
3.4 JAVASCRIPT	26
3.5 MARIADB	26
3.5.1 HeidiSQL	27
3.6 PHPSTORM	28
4 FRAMEWORK SYMFONY	29
4.1 PRINCIP MVC	30
4.1.1 Model	30
4.1.2 View	30
4.1.3 Controller	30
4.1.4 MVC diagram.....	31
4.2 REQUEST A RESPONSE.....	31
4.2.1 Původní HTTP	31
4.2.2 Moderně v Symfony.....	31
4.2.2.1 Symfony Request objekt	32
4.2.2.2 Symfony Response objekt	32
4.2.2.3 Front Controller	32
4.2.2.4 Průběh Request-Response.....	33
4.2.2.5 Pracovní tok Symfony:	33
4.3 STRUKTURA	33
4.3.1 bin.....	34

4.3.2	config.....	34
4.3.3	public.....	34
4.3.4	src	34
4.3.5	templates	34
4.3.6	var.....	34
4.3.7	vendor.....	34
4.4	PHP ŠABLONY TWIG	35
4.5	ORM.....	36
4.6	SYMFONY PROFILER.....	37
II PRAKTICKÁ ČÁST		38
5	ANALÝZA A NÁVRH ŘEŠENÍ.....	39
5.1	POŽADAVKY DEMO APLIKACE	39
5.1.1	Uživatel	39
5.1.1.1	Registrace uživatele	39
5.1.1.2	Přihlášení uživatele	39
5.1.1.3	Úprava profilu.....	39
5.1.1.4	Zobrazení profilu uživatele	39
5.1.1.5	Změna hesla uživatele.....	40
5.1.1.6	Odhlásit se.....	40
5.1.1.7	Uživatelský panel - přehled	40
5.1.2	Nabídka	40
5.1.2.1	Zobrazení nabídek přihlášeného uživatele.....	40
5.1.2.2	Přidání nabídky	40
5.1.2.3	Vlastnosti nabídky	40
5.1.2.4	Úprava nabídky.....	40
5.1.2.5	Deaktivace nabídky.....	40
5.1.2.6	Aktivace nabídky	40
5.1.2.7	Vyhledávání nabídek	41
5.1.2.8	Zobrazení jednotlivé nabídky	41
5.1.3	Rezervace	41
5.1.3.1	Zadání vytvoření rezervace nabídky.....	41
5.1.3.2	Rekapitulace vytvoření rezervace nabídky.....	41
5.1.3.3	Zobrazení rezervací uživatele	41
5.1.3.4	Zobrazení rezervací návštěvníků	41
5.1.3.5	Archiv rezervací.....	41
5.1.3.6	Blokace peněz na rezervaci.....	41
5.1.3.7	Historie transakcí souvisejících s rezervacemi	41
5.1.3.8	Akce s rezervací.....	42
5.1.4	Hodnocení	42
5.1.4.1	Přidání hodnocení rezervované nabídky.....	42
5.1.4.2	Zobrazení hodnocení návštěvníků	42
5.1.4.3	Zobrazení mých hodnocení.....	42
5.1.4.4	Úprava hodnocení	42
5.1.4.5	Zbývá ohodnotit.....	42
5.1.5	Zpráva	42
5.1.5.1	Vytvoření nové zprávy.....	42
5.1.5.2	Zobrazení konverzací.....	42
5.1.5.3	Zobrazení jednotlivé konverzace	42
5.1.5.4	Odpověď v konverzaci	42

5.1.6	Peněženka.....	43
5.1.6.1	Nabití peněženky	43
5.1.6.2	Výběr z peněženky.....	43
5.1.6.3	Historie transakcí	43
5.2	POŽADAVKY NA REÁLNOU APLIKACI.....	43
5.2.1	Nastavení změny statusu rezervace podle času.....	43
5.2.2	Implementace reálného peněžního systému.....	43
5.2.3	Ověření telefonu a e-mailu uživatele	43
5.2.4	Rozhraní pro administrátora.....	43
5.2.5	Poplatek z uskutečněné rezervace pro majitele aplikace	43
5.3	NÁVRH UŽIVATELSKÉHO ROZHRAŇÍ.....	43
5.4	NÁVRH DATABÁZOVÉ STRUKTURY	44
5.4.1	Relační vztahy:.....	45
5.4.2	User	46
5.4.3	Conversation a Message.....	47
5.4.4	Offer	47
5.4.5	Image.....	48
5.4.6	Reservation a Reservation_history.....	48
5.4.7	Review.....	49
5.4.8	Transaction.....	49
6	IMPLEMENTACE	50
6.1	ZALOŽENÍ PROJEKTU	50
6.2	ÚVODNÍ STRÁNKA	50
6.3	ŘEŠENÍ UŽIVATELSKÉHO PŘÍSTUPU	51
6.4	UŽIVATELSKÁ SEKCE	51
6.5	PŘIDÁNÍ NABÍDKY	52
6.5.1	Vykreslení formuláře v Twig template	53
6.5.2	Poloha nabídky.....	54
6.5.3	Tvoření atributu slug	54
6.5.4	Obrázky u nabídky	54
6.6	VYHLEDÁVÁNÍ	55
6.7	PENĚŽENKA.....	57
6.8	REZERVACE.....	58
6.8.1	Vytvoření žádosti o rezervaci.....	58
6.8.1.1	Podmínky:.....	59
6.8.2	Akce rezervace	59
6.8.2.1	Storno rezervace	59
6.8.2.2	Zamítnutí rezervace	60
6.8.2.3	Schválení rezervace	60
6.8.2.4	Převzetí klíčů	60
6.8.2.5	Dokončení rezervace.....	60
6.9	ZPRÁVY.....	60
6.10	HODNOCENÍ NABÍDEK	62
7	ZABEZPEČENÍ	63
7.1	OMEZENÍ PŘÍSTUPU	63
7.1.1	Security voters.....	63

7.2	OCHRANA FORMULÁŘŮ	64
7.3	HASHOVÁNÍ UŽIVATELSKÉHO HESLA	64
ZÁVĚR		65
SEZNAM POUŽITÉ LITERATURY		66
SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK		71
SEZNAM OBRÁZKŮ		72
PŘÍLOHY		73

ÚVOD

Práce řeší zpracování webové aplikace pomocí PHP Frameworku Symfony. Webová aplikace bude sloužit k nabídce a poptávce krátkodobého ubytování v domech či bytech. V České republice je pouze jedna velice podobná aplikace, ale není vytvořena českými vývojáři, proto autor věří, že aplikace bude důvěryhodnější pro českou komunitu a bude tak úspěšnější.

Téma si autor práce zvolil sám, neboť jej velice zajímají dnešní webové technologie, a proto se snaží, aby jeho práce byla co nejvíce aktualizována a moderně zpracována. Proto využívá nejnovějších technologií, které jsou webovým vývojářům nyní dostupné.

V teoretické části se nejprve autor zabývá stručným popisem aplikace a popisem stávajících řešení. Druhá kapitola popisuje všechny potřebné části k implementaci projektu, do kterých patří: doména, hosting, webový server, composer, cmdr a git. Následuje kapitola s popisem technologií použitých k vytvoření této webové aplikace. Dále přichází řada na samotný Framework Symfony, který je detailně popsán. Začíná modelem MVC, na kterém je Symfony stavěno; dále prací s webovými požadavky a odpověďmi na ně; adresářovou strukturou Frameworku. V kapitole je také popsáno, jak Symfony pracuje s View z modelu MVC pomocí Twig šablon; jak přistupuje k datům z databáze a nakonec jak napomáhá k ladění napsaného kódu.

V praktické části je nejprve popsána analýza požadované aplikace. V této analýze je zmíněna představa, jak by tato aplikace měla fungovat. Dále jsou popsány požadavky na demo aplikaci, a také nástin požadavků týkajících se reálné aplikace určené k ostrému provozu. Následuje krátký popis představy uživatelského rozhraní. V kapitole Návrh databázové struktury se detailně provede rozbor databázového schématu. Následující kapitola Implementace pojednává o praktickém provedení vývoje aplikace. Podkapitoly se zaměřují na stěžejní komponenty. Implementace jsou popisovány jako postupy řešení s ukázkovým kódem. Postupuje se od začátku tvoření projektu. V průběhu postupu tvoření se vynechávají již zmíněné části. Praktická část je zakončena kapitolou Zabezpečení, která pojednává o použitých bezpečnostních metodikách.

I. TEORETICKÁ ČÁST

1 O APLIKACI

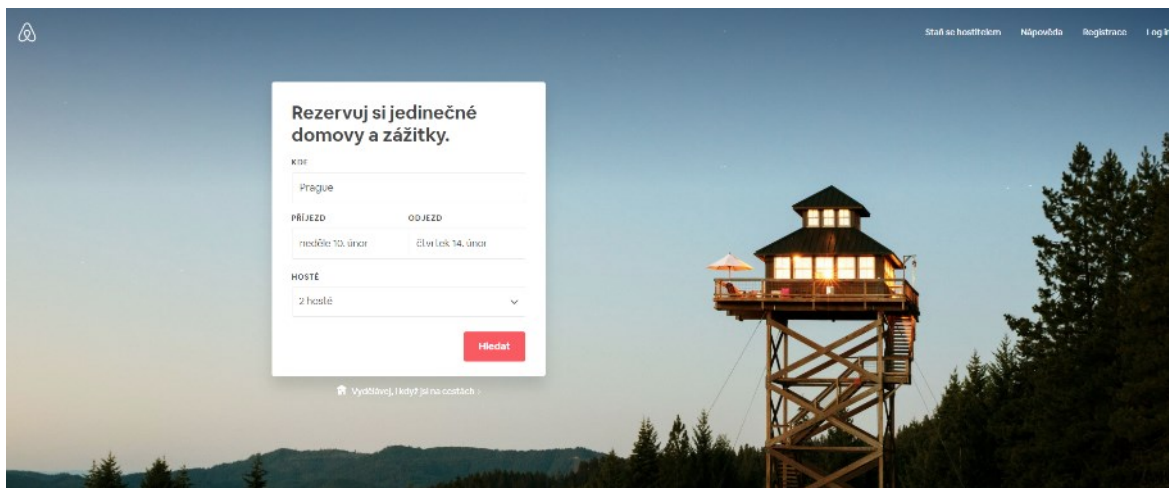
Webová aplikace, která má pracovní název „Ubytuj se“ umožňuje poptávajícímu nalézt krátkodobou možnost ubytování v prostorách soukromých bytů či domů a nabízejícímu (majiteli nemovitosti) možnost krátce pronajímat své byty či domy za určitou finanční odměnu.

1.1 Podobné aplikace

V České republice je pouze jedna podobná služba, která se do ČR rozšířila z USA a tou je Airbnb. Nicméně existuje několik aplikací, které fungují alespoň jako vyhledávače či srovnávače cen krátkodobého ubytování. Těch se v ČR objevuje několik, bohužel většina těchto aplikací je pouze přeložených ze zahraničních jazyků do jazyka českého a ne vždy korektně.

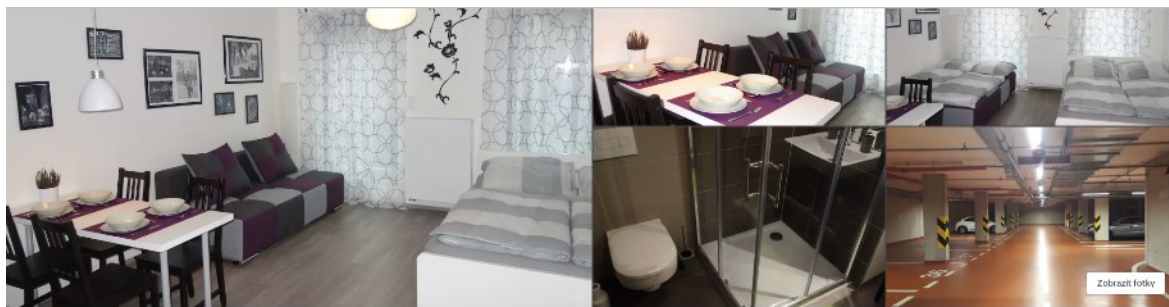
1.1.1 Airbnb.cz

Aplikace je inspirující, neboť je vcelku propracovaná a má i slušnou uživatelskou základnu, navíc má velice moderní vzhled. Nabízí uživatelům velice snadné a intuitivní vyhledávání ubytování a také možnost nabídnout své ubytovací prostory jiným uživatelům. Aplikace funguje celosvětově, tudíž chystáte-li se do zahraničí, můžete využít jejích služeb.



Obrázek 1: Airbnb [1]

Zobrazení vyhledávání i jednotlivých nabídek ubytování je velice přehledné a poutavé. Určitě si každý zákazník lehce vybere, neboť má mnoho detailních informací o daném ubytování jako např. sprcha, topení, Wi-Fi, osušky, fén, ...



CELÝ BYT
COZY STUDIO for 2 - 4 people - FREE PARKING INSIDE
 Praha 3

Studio 1 koupelna

Magda A Milan je Superhost (superhostitelem)
 Superhosts (superhostitelé) jsou zkušení hostitelé s vysokým hodnocením, kteří si dávají záležet na tom, aby hosté pobyt užili.

- Parkování na pozemku v ceně**
 Toto je jadro z mála míst v oblasti, která má toto vybavení.
- Zařívě osvět**
 13 posledních hostů říká, že toto místo září čistotou.

ONLY 10 MINUTES by direct bus lines to OLD TOWN and 6 minutes to the Main Bus station and metro stations Florenc (lines B and C).

405 Kč za noc
 4.9 (27)

Termín
 10.02.2019 → 14.02.2019

Hosté
 2 hosté

405 Kč x 4 noci	1 620 Kč
Poplatek za úklid	141 Kč
Služby poplatek	263 Kč
Celkem	2 024 Kč

[Rezervovat](#)

Obrázek 2: Airbnb – nabídka ubytování [1]

Velkou výhodou této služby je skvělá aplikace na mobily, ve které se snadno vyhledává ubytování, ale také i nabízí vlastní nemovitosti. V mobilní aplikaci je přehledný výpis vašich statistik z pronajímaného ubytování. Airbnb je v porovnání s „Ubytuj se“ velmi podobné, jenže „Ubytuj se“ má tu výhodu, že je vyroben a cílen pro český trh a tudíž mu lidé budou více důvěřovat.

1.1.2 LevneUbytovani.net

Jedná se o pouhý vyhledávač ubytování. Vytvořen Čechy a je funkční od roku 2000, což je na webu hodně znát. Vzhledově není aktualizovaný do moderní podoby. Nabízí ubytování v ČR a na Slovensku, ale také i na Ukrajině a v Polsku. Stránka funguje spíše jako nástěnka s inzeráty a celkový počet možností k ubytování je zde 698 na celou ČR, což je hodně málo.

The screenshot shows the website interface for finding accommodation. At the top, there is a navigation bar with the URL [HTTP://WWW.LEVNEUBYTOVANI.NET](http://www.levneubytovani.net) and a logo for 'Seznam levného ubytování' with flags for Morava, Čechy, and Slovensko. Below the navigation bar, there are several menu items: 'Nabídka', 'Language', 'Přihlášení', 'Důraz na kvalitu', 'Facebook', and 'Spolupracuji s'. The main content area is divided into two sections: 'Hledání ubytování dle kritérií' and 'Hledání podle zadaných parametrů'. The 'Hledání dle kritérií' section includes a search box for 'Fulltextové vyhledávání' and a 'Hledej' button. Below it, there are checkboxes for 'Hledat v' (názvech, obcích, rekreačních oblastech, umístěních, popisesh ubytování) and a note: 'Pozn.: Hledáte-li všechna zařízení z konkrétních hor, stačí do textového pole zadat jejich název. Např. Krkonoše, Jeseníky, Vysoké Tatry.' The 'Hledání podle zadaných parametrů' section includes filters for 'Země' (Čechy, Morava a Slezsko, Slovensko, Ukrajina, Polsko), 'Cena' (0 - 50 Kč, 50 - 100 Kč, 100 - 150 Kč, 150 - 200 Kč, 200 Kč a více), 'Druh ubytování' (rekreační středisko, chata, penzion, ubytování v soukromí, kemp, hotel, ubytovna, horská chata, lázeňské zařízení, tábor), and 'Seřadit podle' (název, ceny, aktualizace). There are also dropdown menus for 'Min. počet lůžek' and 'Min. počet lůžek bez bariér', and checkboxes for 'včetně objektů, kde údaj není znám'. A 'Hledej' button is at the bottom of this section. A note at the bottom of the main content area reads: 'Pozn.: Vyhledáno je vždy maximálně 100 zařízení. Případně více upřesněte svůj dotaz.' The footer contains the copyright information: 'Copyright © 2000 - 2019 Robert Vystrčil (fotograf)'.

Obrázek 3: LevneUbytovani.net [2]

Výpisy jednotlivých ubytování jsou nepřehledné. Sice je zde možnost uvedení názoru k ubytování, ale názory zde žádné nejsou, z čehož vyplývá, že stránku mnoho lidí nepoužívá. Aplikace „Ubytuj se“ je v porovnání s touto stránkou lepší, jak vzhledem, tak i ve funkčnosti a přehlednosti nabízených ubytování.

Penzion Chmelnice, Praha 3 - Jarov
 Praha

zobraz telefonní číslo penzionchmelnice@volny.cz

Umístění: V Praze ve čtvrti Jarov u hlavní ulice Koněvova. Dopravu do centra zajišťují tramvajové linky, 1 s nočním provozem (stanice 200 m, do centra Prahy 20 minut). Do 500 m jsou domácí potřeby, potraviny, lékárny, zdravotní středisko, levná jídelna a pošta. Nejbližší turistickou atrakcí je věž Žižkovského vysílače.

Ubytování: 2- až 4-lůžkové pokoje různých úrovní ubytování (ubytovna, penzion). Většina pokojů je spojená po dvou do buňky se společnou koupelnou a kuchyňkou, některé pokoje mají koupelnu a kuchyňku vlastní. Penzion nabízí více soukromí, v nejvybavenější variantě mají pokoje kuchyňku vybavenou, koupelnu zrekonstruovanou, TV, mýdlo a ručníky na každém pokoji. V areálu penzionu je restaurace, bowling, solárium, herna.

Cena: Ubytovna: 1-7 nocí 232 Kč/os/noc (děti 3-12 let 116 Kč/os/noc), 8-14 nocí 206 Kč/os/noc, 15 a více nocí 174 Kč/os/noc, v případě žádosti o soukromí je nutné zaplatit i část neobsazených lůžek na pokoji. **Penzion:** pokoj 1-lůžkový 410-620 Kč/noc, 2-lůžkový 580-840 Kč/pokoj/noc, 3-lůžkový 750-1.160 Kč/pokoj/noc (podle druhu pokoje), ceny platí při ubytování na 1-7 nocí, delší pobyty se slevou. V ceně jsou lůžkoviny, veškeré energie, DPH. Rekreační poplatek 21 Kč/os/noc (dospělí). Ceny platí celoročně mimo Silvestra.

Adresa: Penzion Chmelnice (pension Chmelnice) a ubytovna Chmelnice, Mgr. Jan Rajmon, Na Chmelnici 1957/9, 130 00 Praha 3 - Jarov

WWW: www.levneubytovani.penzionpraha.cz

Poznámky: Placená inzerce.

Aktualizace: 2. května 2018

Obrázek 4: LevneUbytovani – nabídka ubytování [2]

1.1.3 Ubytovani.net

Asi nejpovedenější český vyhledávač ubytování. Je přehledný a umožňuje i registrace uživatelů a jejich rezervace. Vzhledově je již mnohonásobně modernější než předchozí český zástupce: LevneUbytovani.

ubytovani.NET Registrace Pro ubytovatele

Vyhledávač ubytování v ČR a na Slovensku

Česká republika **Slovensko** Last Minute!

1 Kraj 2 Horská oblast nebo lokalita 3 Tematické ubytování [Rozšířené vyhledávání](#)

Vyhledat ubytování

98% spokojených klientů
Naším cílem jsou spokojení zákazníci, kteří se k nám budou dlouhodobě vracet.

13 let na trhu
Jsme jeden s nejdéle úspěšně působících serverů na internetu, v oblasti vyhledávání ubytování.


Chytrý věrnostní program
Připravujeme pro vás chytrý věrnostní program na míru, jak pro ubytovatele, tak pro naše věrné klienty.


Obrázek 5: Ubytovani.net [3]

Výpisy nabízených ubytování jsou pěkné a moderní. Nachází se zde detailnější informace o daném ubytování, např.: fotky, mapa, popis, cena, koupelna ...

Sport-hotel Praha

U Klikovky č.p. 10, 15000 Praha 5 | km od centra Praha 5





Upravit hotel

Sport-hotel Praha
GPS: 50.0741435,14.3684353

Adresa: U Klikovky č.p. 10, 15000 Praha 5
Telefon: +420 257 210 146
Web: <http://sport-hotel-praha.trade.cz>

Dotaz / Rezervace

KAPACITA
75 lůžek

UBYTOVÁNÍ JIŽ OD
300 Kč

Obecný

Sport-hotel Praha *** se nachází na v klidné části města, na pokraji městského parku a jen pár kroků od obchodního a kulturního centra Anděl.

Bližkost metra a městské hromadné dopravy zaručuje pohodlné a rychlé spojení do historického srdce Prahy. V objektu hotelu je možno využít uzavřeného parkování Vašeho automobilu, vyčleněného prostoru pro Bus.

Vybavení pokojů: koupelna se sprchou a WC na každém pokoji televizor + satelit minibar a osobní bezpečnostní trezor.

Vybavení pokojů kategorie B koupelna se sprchou a WC na patře (4 x WC , 4 x sprcha) společenská hala s televizorem + sat.

Ceník

kategorie A

600 Kč/ dvoulůžkový pokoj (1 osoba)

900 Kč/ dvoulůžkový pokoj

1200 Kč/ třílůžkový pokoj

2200 Kč/ apartmán

kategorie B

400 Kč/ dvoulůžkový pokoj (1 osoba)

600 Kč/ dvoulůžkový pokoj

Zajímavá místa v okolí



Štefánikova hvězdárna

Štefánikova hvězdárna v Praze na vrchu Petřín v nejjihnějším cípu Hradčan, určená široké veřejnosti ...

Zobrazit

Obrázek 6: Ubytovani.net – nabídka ubytování [3]

1.1.4 Booking.com

Jak z názvu aplikace vypovídá je ze zahraniční. Jedná se o aplikaci, která vyhledává ubytování hlavně v hotelech, ale také i apartmánech (spíše hostely a motely, než klasické domy a byty). Vyhledávání je dobře zpracováno a aplikace je přeložena do českého jazyka. V Česku je docela úspěšná a má relativně velkou uživatelskou základnu, nabízí také mobilní aplikaci. Za určité poplatky umožňuje nabídnout zobrazení vlastních apartmánů k pronájmu.

Booking.com Kč [Registrace ubytování](#) [Registrujte se](#) [Přihlášení](#)

[Ubytování](#) [Lety](#) [Pronájem aut](#) [Letištní taxi](#)

Využijte nabídek pro všechny sezóny
Od útulných venkovských chalup až po stylové apartmány

[Hledat](#)

Cestuji pracovně

Vysoké slevy po celém světě. [Najděte si dnešní nabídky](#)

Jakékoliv ubytování hledáte, my ho pro Vás máme...

Kategorie	Podobných nabídek
Apartmány	700 669 apartmánů
Rezorty	21 060 rezortů
Vily	356 097 vil
Horské chaty	12 056 horských chat
Chalupy	115 440 chalup
Luxusní kemp	7 417 luxusních kempů

Nabídněte své ubytování na Booking.com

Pronajměte své ubytování a využijte vydělané peníze, na co chcete.

[Zobrazit více](#)

Zaregistrujte se k odběru tajných nabídek

Přihlaste se k odběru zpravodaje a ceny budou hned nižší!

[Zaregistrovat se](#)

Obrázek 7: Booking.com [4]

Celkový vzhled stránky s výpisem detailů jednotlivých nabídek ubytování se jeví tak, že autoři chtějí poskytnout co nejvíce informací. Takový výpis je ale spíše na škodu, neboť hledat informace ve třech sloupcích působí nekonzistentně a hlavně zmatečně. Levý sloupec slouží pro zobrazení recenzí jiných uživatelů. Prostřední sloupec k popisu ubytování a pravý k: doplňkovým službám; co služby zahrnují a hlavně rezervaci daného ubytování.

The screenshot shows a Booking.com listing for 'Flora house' in Prague. The page is divided into three columns. The left column contains a search filter with fields for destination (Praha), dates (11. února 2019 to 15. února 2019), and number of guests (2 adults, 0 children, 1 room). The middle column shows the apartment name 'Flora house', address '25 Sudomerska, Praha, 130 00, Česká republika', and several photos of the interior. The right column displays a rating of 8.3, a review snippet, and a list of amenities including 'WiFi zdarma' (8.8) and 'Výhled'.

Obrázek 8: Booking – nabídka ubytování [4]

2 POTŘEBY PROJEKTU WEBOVÉ APLIKACE

V kapitole jsou popsány projektové náležitosti, které jsou použity ke tvorbě webové aplikace. Patří zde doména, na které bude webová aplikace spuštěna; hosting, na kterém budou uloženy data webové aplikace; webový server, který zajistí kompilaci PHP kódu a budou se z něj odesílat odpovědi na webové požadavky uživatelů; composer, který zajistí všechny náležitosti Symfony Frameworku a jejich aktualizaci na nejnovější verze; cmdr, díky kterému jsme schopni ze začátku vytvořit kostru Symfony projektu, dále vytvářen Entity, aktualizovat databázi, mazat cache paměť a další; Git, který umožňuje spravovat verze, tudíž ukládat aktuální verze, případně se navracet k verzím starším.

2.1 Doména

Webová doména, častěji pouze doména představuje název webové stránky či URL, někdy je nazývána také „host name“ (název hostitele). Název hostitele spíše podvědomě reprezentuje numerický význam, čímž je IP adresa, která se těžce pamatuje. Doména umožňuje návštěvníkům najít webové stránky, případně se k nim vrátit mnohem snadněji. Dále umožňuje inzerentům dát webové stránce nezapomenutelné jméno, které si návštěvníci zapamatují a díky tomu se budou vracet. Webové domény jsou flexibilní a umožňují, aby bylo napojeno více IP adres ke stejné doméně, čímž webová stránka poskytuje několik různých stránek, ale stále zůstává na snadno zapamatovatelné adrese. [5]

Domény jsou omezeny na určité znaky z ASCII tabulky a to písmena A až Z a čísla 0 až 9, v některých případech je povoleno používat i pomlčku. Názvy všech domén mají zpravidla nějaký specifický význam pro lidi, kteří za ně platí. Každá webová doména je zakončena tečkou a dvou či tří písmenovou referencí, která udává buďto zemi původu, jako např. „pl“ pro Polsko nebo typ webové stránky jako je například velmi populární „com“ pro „commercial“ (komerční) webové stránky. [5]

2.2 Hosting

Web hosting je služba která umožňuje organizacím a jednotlivcům zveřejnit webovou stránku na Internet. Poskytovatel hostingové služby je firma, která poskytuje technologie a služby potřebné k tomu, aby mohla být viděna vaše webová stránka na Internetu. Webové stránky jsou uváděné, či ukládané na speciálních počítačích, kterým se říká servery. Když si chce internetový uživatel prohlédnout vaši webovou stránku, vše co potřebuje je zadat vaši

webovou adresu nebo doménu v jeho prohlížeči. Jeho počítač se připojí na váš server a vaše stránky mu budou doručeny přes prohlížeč. [6]

Většina hostingových společností vyžaduje, abyste vlastnili doménu k hostingu na jejich serverech. Jestliže doménu nevlastníte, hostingová společnost Vám pomůže s její koupí. [6]

2.3 Webový server NGINX

NGINX je open-source webový server. Od prvopočátku je velice úspěšný. [7]

Mezi vysoce úspěšné společnosti, které používají NGINX patří Autodesk, T-Mobile, GitLab, DuckDuckGo, Microsoft, IBM, Google, Adobe, VMWare, Xerox, LinkedIn, Cisco, Facebook, Twitter, Apple, Intel, a mnoho dalších. [7]

NGINX byl původně založen Igorem Sysoevem, kdy bylo v říjnu roku 2004 jeho první oficiální vydání. Igor původně vytvořil tento software jako odpověď na C10k problém, což je problém ohledně potíží s výkonem při manipulaci s 10 000 souběžnými připojeními. [7]

Protože kořeny tohoto softwaru jsou uzpůsobeny k optimalizaci výkonu, NGINX často předčí jiné populární webové servery v benchmark testech, zejména v situacích se statickým obsahem a/nebo vyššími souběžnými webovými požadavky. [7]

2.3.1 Jak NGINX pracuje?

NGINX je postaven k tomu, aby nabídl nízké využití paměti a vysoký počet souběžných připojení. Raději než, aby vytvořil nový proces pro každý webový požadavek, NGINX použije asynchronní událostmi řízený přístup, kde jsou webové požadavky řešeny v jediném vlákně. [7]

Pomocí NGINX serveru může jeden hlavní proces řídit několik pracovních procesů. Hlavní proces udržuje pracovní procesy, zatímco pracovní procesy řeší aktuální zpracovávání webových požadavků. Jelikož je NGINX asynchronní, každý webový požadavek může být zpracován pracovním procesem souběžně bez blokace jiných webových požadavků. [7]

2.4 Composer

Composer je manažer balíků, které používají moderní PHP aplikace. Composer se používá ke správě závislostí v naší Symfony aplikaci a k instalaci Symfony komponent, které jsou potřebné k celkovému běhu Frameworku v PHP projektu. [8, 9]

Composer není správce balíčku ve stejném smyslu jako Yum nebo Apt (v Linuxu). Sice pracuje s „balíčky“ nebo knihovnamy, ale spravuje je v každém projektu zvlášť, kde je instaluje do složky (např. vendor) v projektu. Ve výchozím stavu neinstaluje nic globálně. Proto je to správce závislostí. [10]

Předpokládejme:

1. Máme projekt, který se spoléhá na nějaký počet knihoven.
2. Některé z těchto knihoven spoléhají na jiné knihovny.

Composer:

1. Umožňuje nám deklarovat knihovny, na kterých nám projekt spočívá.
2. Zjistí, jaké verze kterých balíčků (knihoven) jsou k dispozici, či je třeba nainstalovat a nainstaluje je do našeho projektu. [10]

2.5 Cmder

CMDER je bezplatný přenosný emulátor konzole (příkazového řádku) pro Windows. Konzolovým emulátorem se míní to, že zaobaluje funkci příkazového řádku spolu s větším množstvím funkcí a intuitivního zážitku. [11]

Seznam funkcí, které přináší CMDER se stále rozšiřuje a vývojáři uvádí, že jejich nástroj byl vytvořen z čiré frustrace z absence více uživatelsky přívětivého příkazového řádku ve Windows. Tento nástroj je velmi užitečný, neboť nabízí rozšířené funkce, klávesové zkratky a aliasy. [11]

CMDER je široce přizpůsobitelný a nabízí dlouhý list nastavení, které lze změnit a přizpůsobit si je. Vzhled lze snadno modifikovat změnou barvy, pozice a fontů. Je zde také možnost použití obrázku na pozadí konzole. [11]

CMDER podporuje docela dost klávesových zkratk, které si lze taktéž přizpůsobit. Lze si zde nastavit kopírovat/vyjmout z konzole, což je v jiných programech obtížné. Kromě toho nástroj může i zablokovat klávesy, či vytvářet makra. [11]

2.6 Systém správy verzí – Git

Git je VCS (rozumějme: Version Control System – systém správy verzí), díky kterému můžeme vytvářet projekty různých velikostí s rychlostí a efektivitou. Pomáhá spravovat kód, komunikovat a spolupracovat se členy vývojářského týmu na různých softwarových projektech. Git umožňuje vrátit se zpět do předchozího stavu projektu nebo se podívat na celou evoluci projektu od jeho založení. Lze na něj pohlížet, jako na stroj času, který umožňuje vrátit se zpět časem v našem projektu, kdykoliv chceme. Díky práci s Gitem získáváme tři výhody při práci na projektech:

1. Lehčí správa velkých projektů
2. Pomáhá nám vyhnout se přepsání pokroku a práce týmu.
3. S Gitem si zkopírujete kód a historii do vašeho zařízení, pročež můžete v klidu pracovat ve svém prostředí bez rušení, či nějakých hranic. Je to mnohem jednodušší a lehčí. [12, 13]

Git je v dnešní době nejrozšířenější systém správy verzí a je také nejmodernější. Neboť je vyspělý a aktivně spravovaný open source projekt, původně vytvořen v roce 2005 Linusem Torvaldsem, slavným zakladatelem jádra operačního systému Linux. Nezměrné množství softwarových projektů spoléhá na Git, díky jeho správě verzí, včetně komerčních projektů tak jako i open source projektů. Vývojáři, kteří pracovali s Gitem již v minulosti většinou patří mezi talentované vývojáře software. [14]

Nabízí distribuovanou architekturu, neboť Git je příkladem DVCS (rozumějme: Distributed Version Control System – Distribuovaný systém správy verzí), tudíž v Gitu každá vývojářská kopie kódu je také repositářem, který obsahuje celou historii všech změn.

Spolu s distribuovaností je Git uzpůsoben k velkému výkonu, zabezpečení a flexibilitě. [14]

2.6.1 GitLab

GitLab je první jednotná aplikace pro celý životní cyklus DevOps (Development Operations – Operace vývoje), který zahrnuje pět fází: vývoj, testování, integrace, nasazení, monitoring. GitLab poskytuje bezkonkurenční viditelnost, vyšší úroveň účinnosti a komplexní řízení. Toto umožňuje, aby se životní cyklus softwaru prováděl 200 % krát rychleji, což radikálně zlepšuje rychlost podnikání. [15]

GitLab.com

- GitLab.com hostuje vaše (privátní) softwarové projekty zdarma.
- Nabízí bezplatné veřejné a soukromé úložiště, sledování problémů a podporu wiki.
- Nabízí GitLab Enterprise Edition a GitLab CI/CD.
- Žádná instalace není zapotřebí, stačí si zaregistrovat bezplatný účet.
- Bezplatný účet na GitLab.com podporuje Support Tracker – webovou podporu, která vám zodpoví vaše dotazy. [12]

2.6.2 GitKraken

Zatímco Git může být výzvou pro nezasvěcené, uživatel může zvolit skvěle designované grafické uživatelské prostředí Gitu, čímž je GitKraken. Díky němu bude schopný práce s Gitem téměř okamžitě. [13]

GitKraken nepodporuje dočasné přecházení na staré commity (verze), ale umožňuje zobrazení změn v celém adresáři a samozřejmě i v jednotlivých souborech, což je mnohem jednodušší, než si pamatovat tucty argumentů pro příkaz „git log“. [13]

Používání GitKrackenu umožňuje vizuální zobrazení, díky čemuž se snadněji naučí některé základní principy Gitu, což napomáhá ke kompetentnější a sebevědomější spolupráci s Gitem. [16]

3 POUŽITÉ TECHNOLOGIE

Tato kapitola se zaměřuje na popis technologií, které jsou využity ke tvorbě aplikace.

3.1 HTML

HTML (Hypertext Markup Language – Hypertextový značkovací jazyk) je textově založený přístup k popisu struktury obsahu v HTML souboru. Značky sdělují prohlížečům, jak se má zobrazit daný text, obrázek a jiné formy multimédií na webové stránce. [17]

HTML je formálně doporučeno World Wide Web Consortiumem (W3C) a toto doporučení je dodržováno všemi hlavními webovými prohlížeči, včetně prohlížečů na stolních počítačích a mobilních zařízeních. Nejnovější a aktuálně používanou verzí je HTML5. [17]

Role HTML je informovat webový prohlížeč o tom, jak se má obsah strukturovaného HTML souboru zobrazit. Mezi nejčastěji používané značky HTML patří `<h1>`, která popisuje nadpis nejvyšší úrovně; `<h2>`, která značí nadpis druhé úrovně; `<p>` k popisu odstavce; `<table>`, popis tabulky; a ``, která popisuje číslovaný seznam informací. [17]

3.2 CSS

CSS (Cascading Style Sheets – Kaskádové styly) jsou standardem a preferovaným mechanismem pro formátování HTML stránek. [18]

Kaskádové styly poskytují centralizované informace o různých fontech, barvách, barvách pozadí, kurzívy, zdůrazňování HTML elementů na webové stránce. Kaskádové styly mohou také vzhledově upravovat různé části stránky, jako například hlavičku, patičku, tělo, články a sekce. Což je velmi nápomocné, pokud jejich zobrazení je rozvrženo odlišným stylem pro stolní počítače, tablety nebo chytré telefony. [18]

Kaskádová povaha CSS souborů je přičítána skutečnosti, že informace o stylování webové stránky, lze definovat ze tří různých míst, také známých jako úrovně stylů. [18]

Preferovaný způsob je vložení informací o stylech do samostatného souboru s koncovkou „.css“, který se do webové stránky vloží pomocí HTML značky `<link>`. Webová stránka může být provázána s nulou, jedním anebo i více css soubory (pomocí vícenásobného použití značky `link`). Druhá úroveň stylů je definována pomocí HTML značky `<style>` ve hlavičce HTML souboru, což se při větším objemu stává nepřehledné. Třetí úroveň se definuje přímo v HTML souboru, u daného element pomocí atributu `style`. [18]

Syntaxe CSS je relativně jednoduchá. Jméno elementu, který má být stylován, v CSS nazýván jako selektor je následován složenými závorkami, ve kterých jsou různé atributy, jako např. font-size a background-color a k nim jsou přiřazeny hodnoty. W3C definuje standardní CSS atributy. [18]

CSS selektory mohou být HTML značky, atribut elementu – class, či dokonce stavy daných elementů, jako např. disabled u vstupního pole formuláře, nebo hover (přejetí myši nad odkazem). Možnost přizpůsobit styl komponent v určitých stavech poskytuje grafickým designérům velkou flexibilitu v definování jak se má webová stránka zobrazit. [18]

3.3 PHP

PHP je populární skriptovací open source jazyk všeobecně použitelný, ale zvláště vhodný pro vývoj webových stránek, může být vestavěn do HTML souborů. Je rychlý, flexibilní a pragmatický, PHP pohání vše od vašich blogů po nejpůlárnější webové stránky světa. Nejnovější verzi PHP je nyní PHP 7.3.2, tato verze bude použita při tvorbě aplikace. [19, 21]

PHP se používá pro tvoření dynamických webových stránek. HTML a CSS pouze poskytují strukturu a stylování. PHP je jazyk, který je kompilován na straně serveru, tudíž mu webové prohlížeče nerozumějí. PHP vám může sdělit čas, přeložit informaci, vypočítat něco, připojit k databázi a mnoho dalšího. Je to open source jazyk, tudíž nepotřebujete žádnou licenci, abyste jej mohli využívat pro vývoj. Může generovat HTML, využívat cyklů za různých podmínek, dotazovat se uživatelů a podporuje mnoho podobných programovacích funkcí. [20]

Co odlišuje PHP od klientských skriptovacích jazyků jako JavaScript je to, že jeho kód je prováděn na straně serveru, což generuje HTML, které je poté odesláno klientovi. Klient tedy pak dostane výsledek proběhlého skriptu, ale neví, jak vypadal kód, který výsledky předčil. Dokonce můžete nakonfigurovat váš webový server tak, aby zpracoval všechny vaše HTML soubory pomocí PHP, a pak skutečně neexistuje žádný způsob, jak by uživatelé mohli říct, „co máte v rukávu“. [21]

Nejlepší věcí na používání PHP je to, že je velice jednoduchý pro začátečníky, ale nabízí také mnoho pokročilých funkcí pro zkušené programátory. [21]

3.4 JavaScript

JavaScript byl původně vytvořen aby „oživil webové stránky“. Programy v tomto jazyce se nazývají skripty. Mohou být napsány přímo v HTML souboru webové stránky a provedou se automaticky po načtení stránky. Skripty jsou poskytovány a prováděny jako čistý text. Nepotřebují žádnou speciální úpravu nebo kompilaci, pro jejich spuštění. [22]

Moderní JavaScript je „bezpečný“ programovací jazyk. Neposkytuje nízko-úrovňový přístup k paměti nebo CPU, protože byl původně vytvořen pro webové prohlížeče, které tuto funkci nevyžadují. Možnosti JavaScriptu hodně záleží na prostředí, ve kterém běží. Například, Node.JS podporuje funkce, které umožňují JavaScriptu číst/zapisovat libovolné soubory, provádět webové požadavky, atd. JavaScript v prohlížeči může provádět všechno, co se týče manipulace s webovými stránkami, interakcí s uživatelem a webovým serverem. [22]

Například, JavaScript v prohlížeči je schopen:

- Přidat nový HTML kód do stránky, změnit existující obsah, změnit styly.
- Reagovat na uživatelské akce: kliknutí myši, pohyb ukazatele, zmáčknutí kláves.
- Posílat požadavky přes síť na vzdálené servery, stahovat a nahrávat soubory (takzvané AJAX a COMET technologie).
- Získávat a nastavovat cookies, pokládat otázky návštěvníkům, zobrazovat zprávy.
- Pamatovat si data na straně klienta (“local storage”). [22]

Tyto tři věci jsou úžasné na JavaScriptu:

- Plná integrace s HTML/CSS.
- Jednoduché věci, jsou prováděny jednoduše.
- Podpora všech důležitých prohlížečů a je ve výchozím stavu povolen. [22]

JavaScript je jedinou prohlížečovou technologií, která kombinuje tyto tři věci. To činí JavaScript unikátním. Proto je nejrozšířenějším nástrojem pro vytváření prohlížečového rozhraní. [22]

3.5 MariaDB

MariaDB je systém řízení báze (DBMS) s open source licencí, který je vhodnou kompatibilní náhradou pro široce používanou databázovou technologii MySQL. MariaDB bylo založeno jako větev MySQL od tvůrců, kteří hráli klíčovou roli v budování původní databáze

v MySQL, navrhli tak MariaDB v roce 2009 jako reakci na akvizici MySQL společností Oracle Corp. [23]

Většina vývojové práce na open source databázi se soustředila na dosažení shody funkcí mezi MariaDB a MySQL. MariaDB Corp., hnací síla MariaDB, říká, že „MariaDB je uzpůsoben všem praktickým účelům“, databáze je navíc binárně kompatibilní s MySQL. Z čehož vyplývá, že většina uživatelů může přepnout mezi těmito dvěma technologiemi jednoduše pouhým odinstalováním MySQL a nainstalováním MariaDB na jeho místě. [23]

MariaDB je poskytován jako open source software pod druhou verzí GNU General Public Licence (GPL). Mezi uživateli MariaDB jsou například Wikipedia, Facebook a Google, stejně jako databáze Singapurské banky a Institut pro měření a hodnocení zdravotní péče v Seattlu, a další. [23]

MariaDB se vyvíjí průběžně a aktualizace jsou k dispozici konečným uživatelům velmi rychle, je zde také detailní funkce sledování chyb. MariaDB je optimalizována pro výkon a je mnohem výkonnější u velkých množství dat. Migrování z jiných databázových systémů na MariaDB je tudíž velmi přínosné. [24]

3.5.1 HeidiSQL

HeidiSQL, původně znám jako MySQL-Front, je bezplatný open source klient, či frontend aplikace pro MySQL (a jeho větve jako MariaDB či Percona Server). [25]

HeidiSQL je bezplatný software, který cílí k tomu, aby byl snadný k naučení. „Heidi“ vám umožní prohlížet a upravovat data a struktury z počítačů, které běží na databázových systémech MariaDB, MySQL, Microsoft SQL nebo PostgreSQL. Heidi bylo založeno v roce 2002 Ansgarem, vrchol vývoje byl pak mezi roky 2009 až 2013, HeidiSQL patří celosvětově mezi nejpoblárnější nástroj pro MariaDB a MySQL. [26]

Ke správě databáze pomocí HeidiSQL se uživatelé musí přihlásit do lokální nebo vzdáleného MariaDB serveru se správnými údaji, čímž vytváří „session“. V této session mohou uživatelé spravovat MariaDB databázi přes připojení k MariaDB serveru, až jsou hotovi, od serveru se odpojí. Soubor funkcí HeidiSQL je dostačující pro většinu běžných a pokročilých databázových, tabulkových a záznamových operací, ale stále zůstává v aktivním vývoji, aby byl schopný plné funkčnosti, jaká je v databázovém rozhraní SQL. [25]

3.6 PhpStorm

PhpStorm je inovativní integrované vývojové prostředí (IDE) založené na jazyce Java, IDE je vyvinuté firmou JetBrains pro PHP a webové vývojáře. Software využívá nástroj IntelliJ IDEA, který umožňuje vývojářům psát kód ve více jazycích zároveň, včetně CSS, HTML5, JavaScript. Je to kompaktní verze IntelliJ s přidanou podporou PHP. PhpStorm optimalizuje pomoc při kódování pomocí špičkových kódů, hloubkového pochopení kódu a podporou hlavních IDE frameworků a PHP nástrojů. [27]

Přináší působivou sadu funkcí a umožňuje přidávat další rozšíření funkcí (plugins) pro zvýšení produktivity a pohodlí programátora. Pracuje perfektně s Frameworky Magento, Drupal, Symfony, Zend, Wordpress, Joomla, CakePHP a Laravel. PhpStorm využívá průkopnických technologií front-end vývoje a využívá robustní refactoring, testování a nástrojů ladění. [27]

PhpStorm je spolehlivý, inteligentní kódovací asistent, který zahrnuje všechny CSS, HTML a JavaScript výhody programu Webstorm od JetBrains. Podporuje také všechny inovativní technologie webového vývoje včetně CSS, SASS, HTML, SCSS, CoffeeScript, LESS, Jade templates, ECMAScript, a další. Kromě toho je dodáván s elegantním editorem Javascriptu a live editací, která umožňuje okamžitě zobrazit všechny provedené změny. [27]

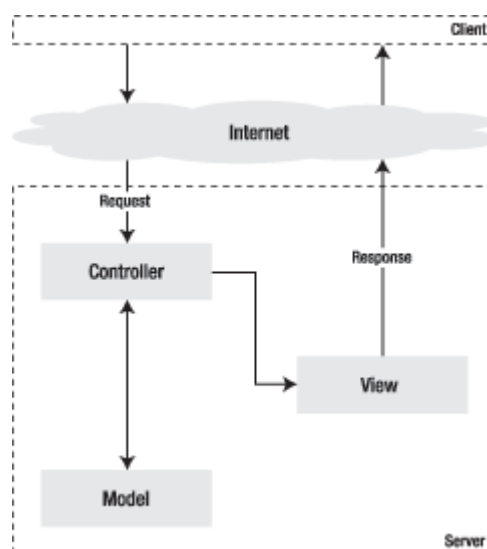
4 FRAMEWORK SYMFONY

Obecně Framework, se skládá z:

- „Krabice nástrojů“ – sada předpřipravených, rychle integrovaných softwarových komponent. To znamená, že napíšete méně kódu s menším rizikem chybovosti. Také to znamená větší produktivitu a schopnost věnovat více času věcem, které mají větší přidanou hodnotu, jako správa hlavních zásad, vedlejších účinků apod.
- Metodologie – jedná se o „montážní diagram“ pro aplikace. Strukturovaný přístup se může zdát nejprve omezující, ale ve skutečnosti dovoluje vývojářům pracovat efektivně na nejsložitějších aspektech úkolu a používat osvědčené postupy, které zaručují stabilitu, udržitelnost a zdokonalitelnost aplikace, kterou vytváříme. [28]

Symfony je Framework webových PHP aplikací. Přidává novou vrstvu nad jazykem PHP, která poskytuje nástroje, které urychlují vývoj komplexních webových aplikací. K porozumění a práci s tímto Frameworkem stačí být obeznámen se základními koncepty moderního programování, jmenovitě: Objektově orientované programování (OOP), Objektově relační mapování (ORM) a Rapid application development (RAD). Jediným požadovaným technickým zázemím je znalost PHP. [29]

Symfony je založeno na klasickém vzoru webového návrhu známém jako architektura MVC, která se skládá ze tří úrovní (Model, View, Controller) [31]:



Obrázek 9: Model MVC [30]

4.1 Princip MVC

MVC znamená Model-View-Controller. MVC je aplikační designový model, který se skládá ze tří propojených částí. Zahrnuje Model (data), View (uživatelské rozhraní) a Controller (procesy, zpracovávající vstup). [32]

MVC model je běžně používán pro vývoj moderních uživatelských rozhraní. Poskytuje základní kusy pro designování programů stolních počítačů či mobilů, ale i webových aplikací. Pracuje skvěle s objektově orientovanými jazyky, protože části Model, View a Controller lze považovat za objekty a lze je opakovaně používat v rámci aplikace. [32]

MVC architektura odděluje obchodní logiku (Model) a zobrazení (View), což vede k větší udržitelnosti. Například jestliže by naše aplikace měla běžet na standardních webových prohlížečích a kapesních zařízeních, potřebuje pouze nový pohled (View), můžeme tedy zachovat původní Controller a Model. Controller pomáhá skrýt podrobnosti protokolu používaného pro požadavek (HTTP, konzolový režim, pošta a podobně) z Modelu a View. A Model představuje logiku dat, která činí View a Controller pro danou akci nezávislou, například typ databáze použitý pro danou aplikaci. [31]

4.1.1 Model

Představuje data, se kterými program pracuje. Může se například jednat o databázi, soubor, nebo nějaký objekt, jako např. ikona či charakter v počítačové hře. [32]

4.1.2 View

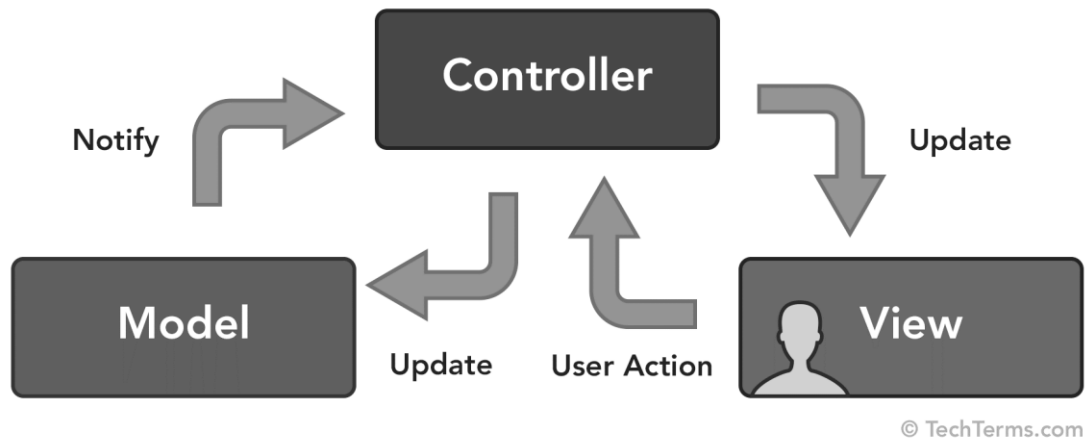
Vykresluje Model do webové stránky vhodný pro interakci s uživatelem. Například zobrazení okna, či tlačítka nebo textu v okně. Zahrnuje vše, co může uživatel vidět. [31, 32]

4.1.3 Controller

Controller aktualizuje obě části Model i View. Přijímá vstup a zpracovává odpovídající aktualizaci. Například Controller může aktualizovat model změněním atributů u postavy v počítačové hře. Může také upravit View, tím, že zobrazí tuto upravenou postavu. [32]

4.1.4 MVC diagram

Tyto tři části M-V-C jsou propojeny. View zobrazuje Model uživateli. Controller přijímá uživatelský vstup a upravuje Model a View odpovídajícím způsobem. Ačkoli MVC architektura není požadována v designu aplikací, mnoho programovacích jazyků a IDE podporuje MVC strukturu, čímž se stává běžně využívanou strukturou vývojáři. [32]



Obrázek 10: MVC diagram [33]

4.2 Request a Response

Jedná se o způsob komunikace mezi klientem a serverem.

4.2.1 Původní HTTP

Krok 1: Klient zašle Request (žádost) – každá webová konverzace začíná pomocí Request. Request je textová zpráva, vytvořená klientem (např. prohlížečem, aplikací telefonu, atd.) ve speciální formě známé jako HTTP. Klient zašle tento Request na server a čeká na odpověď. [34]

Krok 2 – Server odpoví pomocí Response (odpověď) – Jakmile server přijme Request, ví server přesně, které zdroje klient potřebuje (pomocí URL) a co chce klient provádět s těmito zdroji (pomocí Method). Například v případě GET Request, server připraví zdroje a vrátí je v HTTP Response. [34]

4.2.2 Moderně v Symfony

Symfony poskytuje alternativu k hrubému PHP přístupu pomocí dvou tříd, které umožňují interakci s HTTP Request a Response jednodušší cestou. [34]

4.2.2.1 Symfony Request objekt

Request třída je objektově orientovanou reprezentací HTTP Request zprávy. S tímto způsobem máte všechny informace z Request na dosah ruky. A jako bonus Request třída dělá spoustu práce na pozadí, o kterou se nikdy nebudeme muset starat. Například `isSecure()` metoda kontroluje tři různé hodnoty v PHP, které informují, zda se uživatel připojuje pomocí zabezpečeného připojení (např. HTTPS). [34]

4.2.2.2 Symfony Response objekt

Symfony také poskytuje Response třídu: PHP reprezentaci HTTP Response zprávy. Tato reprezentace umožňuje vaší aplikaci použít objektově orientované rozhraní pro konstrukci Response, která je potřeba k odeslání klientovi. [34]

4.2.2.3 Front Controller

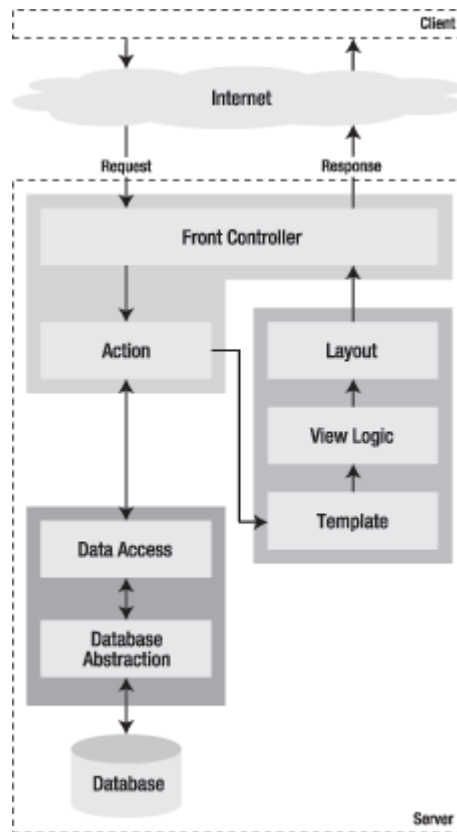
Tradičně byly aplikace tvořeny tak, že každá její stránka byla reprezentována jejím souborem (např. `index.php`, `contact.php`, atd.) [30]

Mnohem lepší řešení nabízí Front Controller: jediný PHP soubor, který se zabývá každou příchozí Request zprávou do vaší aplikace. [34]

4.2.2.4 Průběh Request-Response

- Klient (např. prohlížeč) zašle HTTP Request;
- Na každý Request se spustí stejný soubor (zvaný Front Controller);
- Front Controller načte Symfony a předá mu Request informace;
- Vnitřně Symfony používá Routes (URL adresy) a Controllers k vytvoření Response pro danou stránku;
- Symfony přetvoří váš Response objekt na textovou hlavičku a obsah (a to na HTTP Response), které jsou zaslány zpět klientovi. [34]

4.2.2.5 Pracovní tok Symfony:



Obrázek 11: Pracovní tok Symfony

[35]

4.3 Struktura

Zahrnuje rozdělení celého Frameworku do jednotlivých částí, pro které má Symfony vyhrazené jednotlivé adresáře.

4.3.1 bin

Nejdůležitější v tomto adresáři je bin/console, která za nás automatizovaně vykonává pomocí příkazů v příkazové řádce vytváření entit, mazání cache, spuštění serveru, ... [31, 36]

4.3.2 config

Obsahuje veškeré konfigurace Frameworku Symfony. Třeba konfigurace routes, services a packages. [31, 36]

4.3.3 public

Obsahuje veškeré veřejně dostupné soubory, jako: css, js, obrázky. [31, 36]

4.3.4 src

Zde se nachází veškerý PHP kód. [31, 36]

4.3.5 templates

Nachází se zde veškeré Twig šablony. [31, 36]

4.3.6 var

V této složce se nachází automaticky vytvořené soubory jako cache (var/cache/) či logy (var/logs/). [31, 36]

4.3.7 vendor

Obsahuje dodatečné knihovny třetích stran, které se snadno instalují pomocí Composeru. [31, 36]

4.4 PHP šablony Twig

Twig je flexibilní, rychlý a bezpečný šablonový nástroj pro PHP. [37]

Klíčové funkce:

- *Rychlost*: Twig kompiluje šablony až na prostý a optimalizovaný PHP kód. Režijní náklady v porovnání s klasickým PHP kódem byly sníženy na minimum.
- *Bezpečnost*: Twig lze spustit v „sandbox“ (bezpečném) módu pro vyhodnocení nedůvěryhodného kódu šablony. Tento mód umožňuje Twigu být použit jako šablonový jazyk pro aplikace, kde uživatelé mohou upravovat design šablony.
- *Flexibilita*: Součástí Twigu je flexibilní lexer (převádí source kód na malé části pro snadnější zpracování) a parser (lexerem převedený kód konvertuje na Abstraktní strom Syntaxe, který se dále kompiluje na PHP kód). Tyto součásti umožňují vývojáři definovat jejich vlastní značky a filtry a vytvářet jejich vlastní twig syntaxi. [37, 38]

Výhody Twigu:

Stručný popis: Jazyk PHP je podrobný a stává se ještě podrobnějším, pokud jde o výstupní escapování (převedení na string pro bezpečné vložení, do cílového výstupu) [39, 40]:

```
<?php echo $var ?>
```

```
<?php echo htmlspecialchars($var, ENT_QUOTES, 'UTF-8') ?>
```

Twig má v porovnání s PHP mnohem stručnější syntaxi, a proto jsou šablony více čitelné [39]:

```
{{ var }}
```

```
{{ var|escape }}
```

```
{{ var|e }}           {# zkrácená varianta escapování proměnné #}
```

Mnoho funkcí: Twig podporuje vše, co potřebujete ke snadné tvorbě efektivních šablon: dědičnost, bloky, automatické escapování výstupu a mnoho dalšího [39, 40]:

```
{% extends "layout.html" %}
```

```
{% block obsah %}
```

```
Obsah stránky...
```

```
{% endblock %}
```

Lze se snadno naučit: Syntaxe je velice jednoduchá k naučení a je optimalizována k tomu, aby web designer splnil svou práci rychle a bez problémů.

Jistěže PHP nabízí mnoho šablonových nástrojů, ale většina z nich se nezabývá těmito osvědčenými postupy:

Rozšiřitelnost: Twig je dostatečně flexibilní pro všechny vaše potřeby, dokonce i ty nejvíce komplexnější. Díky otevřené architektuře, můžete do něj implementovat vlastní jazykové konstrukce (značky, filtry, funkce a dokonce operátory), čímž vytvoříte vlastní Twig syntaxi.

Testovaná jednotka: Twig je plně testovanou jednotkou. Jeho knihovna je stabilní a připravena k použití ve velkých projektech.

Dokumentace: Twig je kompletně zdokumentován online.

Chybové zprávy: Kdykoli nastane problém v syntaxi v rámci šablony, Twig poskytne na výstupu nápomocnou zprávu s názvem souboru, číslem řádku kódu, kde chyba nastala, což velmi zlehčuje ladění chyb. [39]

4.5 ORM

Databáze fungují na relacích. PHP je objektově orientovaný jazyk. Abychom mohli přistoupit k databázi objektově orientovaným přístupem, je zapotřebí rozhraní, které přeloží objektovou logiku na relační logiku. Toto rozhraní se nazývá Objektově relační mapování (ORM). ORM je tvořeno objekty, které poskytují přístup k datům. [41]

Jednou z výhod objektově-relačně abstraktní vrstvy je to, že zabraňuje v užití syntaxe specifické pro danou databázi, neboť automaticky překládá volání modelu objektů na SQL dotazy optimalizované pro danou databázi. Proto je přechod na jiný databázový systém uprostřed projektu snadné. [41]

Abstraktní vrstva zapouzdřuje logiku dat. Zbytek aplikace nepotřebuje vědět cokoli o SQL dotazech. SQL dotazy pouštěné na databázi jsou snadné k nalezení. Vývojáři, kteří se specializují na databázové programování, rovněž jasně vědí, kde hledat. Používání objektů místo záznamů a tříd místo tabulek má další výhodu: své tabulky můžete stále snadno modifikovat. [41]

Symfony podporuje dva nejpoblárnější open source ORMy v PHP: Propel a Doctrine. Symfony integruje oba bez problémů. Při vytvoření nového projektu Symfony se naskýtá otázka výběru ORM mezi Propel či Doctrine. V této práci bude použit Doctrine ORM. [41]

4.6 Symfony Profiler

Jednou ze stěžejních funkcí Symfony je Web Debug Toolbar – lišta zobrazující obrovské množství ladících informací v dolní části stránky při vývoji. Instaluje se pomocí Composeru nainstalováním balíčku s názvem symfony/profiler-pack. [36]

Profiler je velmi mocný vývojářský nástroj, který poskytuje detailní informace o vykonání jakéhokoli webového požadavku. Nesmí být nikdy povolen v produkčním prostředí, neboť by to mohlo vést k závažným bezpečnostním slabinám projektu. [42]

II. PRAKTICKÁ ČÁST

5 ANALÝZA A NÁVRH ŘEŠENÍ

Základní myšlenkou bakalářské práce je navržení funkční webové aplikace pro jednoduchý systém ubytování. Systém by sloužil hostitelům ke krátkodobému pronájmu jeho nemovitostí a zájemci o ubytování ke snadnému nalezení krátkodobého ubytování. V aplikaci tyto dvě uživatelské role nejsou rozlišeny, proto uživatel může figurovat jako hostitel a ubytovaný (zákazník).

Neregistrovaný uživatel má v aplikaci limitované možnosti. Může vyhledávat a prohlížet jednotlivé nabídky. Registrovaný uživatel získává přístup k celému potenciálu aplikace. Což zahrnuje komunikaci s ostatními uživateli; vytváření rezervací a nabídek; hodnocení nabídek; správa uživatelského účtu a peněžního konta.

Systém rezervací, by měl zahrnovat vytvoření potenciální rezervace, která lze stornovat zákazníkem, ale také zamítnout či schválit hostitelem. Po schválení hostitelem se stává rezervace aktivní a nejde zrušit. V reálné aplikaci se zrušení aktivní rezervace řeší přes správce aplikace. V den počátku rezervace se předpokládá, že klient převezme klíče k ubytování a tím zahájí průběh rezervace. Posledním dnem rezervace je rezervace skončena a pro získání finanční částky za tuto rezervaci, musí hostitel potvrdit ukončení rezervace. Nepotvrdí-li má možnost kontaktovat správce v rámci jakýchkoli nesrovnalostí s rezervací.

5.1 Požadavky demo aplikace

5.1.1 Uživatel

5.1.1.1 *Registrace uživatele*

K používání aplikace se musí uživatelé registrovat.

5.1.1.2 *Přihlášení uživatele*

Uživatel se autorizuje přihlášením ke svému uživatelskému účtu.

5.1.1.3 *Úprava profilu*

Uživateli je umožněno měnit údaje o jeho osobě v uživatelském profilu.

5.1.1.4 *Zobrazení profilu uživatele*

Možnost zobrazení uživatelských profilů s příslušnými informacemi, nabídkami a s hodnocením těchto nabídek.

5.1.1.5 Změna hesla uživatele

Uživatel si může změnit heslo.

5.1.1.6 Odhlásit se

Schopnost odhlásit aktuálního uživatele a přejít do anonymního režimu prohlížení.

5.1.1.7 Uživatelský panel - přehled

Možnost zobrazení uživatelského panelu se souhrnnými informacemi aktuálně aktivních nabídek, celkovým počtem hodnocení, rezervací uživatelových nabídek a jeho celkový počet jeho rezervací. Dále lze zjistit poslední změnu stavů rezervací a historii transakcí.

5.1.2 Nabídka

5.1.2.1 Zobrazení nabídek přihlášeného uživatele

Možnost zobrazit v uživatelském panelu veškeré nabídky daného uživatele, jak aktivních tak i deaktivovaných.

5.1.2.2 Přidání nabídky

Vytvoření nové nabídky s veškerými vlastnostmi.

5.1.2.3 Vlastnosti nabídky

Nabídka obsahuje: Titulek ubytování, Místo, Typ ubytování, Rozsah ubytování, Maximální počet hostů, Počet pokojů pro hosty, Popis nabídky, Počet koupelen, Maximální délka pobytu, Domácí pravidla, Kontakty, Galerie, Vyzvednutí klíčů, Konec ubytování hostů a Ceník.

5.1.2.4 Úprava nabídky

Nabídku lze upravit.

5.1.2.5 Deaktivace nabídky

Neplatnou nabídku je možno deaktivovat.

5.1.2.6 Aktivace nabídky

Pokud nabídka nabude nové platnosti, lze ji opětovně aktivovat.

5.1.2.7 Vyhledávání nabídek

Hledáme-li ubytování, musí být umožněno vyhledávání podle našich preferencí. V aplikaci lze zvolit: Středový bod vyhledávání, do jakého okruhu lze nabídky vyhledat a v jakém termínu je nabídka dostupná, počet plánovaných hostů.

5.1.2.8 Zobrazení jednotlivé nabídky

Vyhledáme-li nabídku, musí zde být možnost zobrazení všech vlastností dané nabídky.

5.1.3 Rezervace

5.1.3.1 Zadání vytvoření rezervace nabídky

Pro vytvoření rezervace je dostupný patřičný formulář u zobrazení dané nabídky.

5.1.3.2 Rekapitulace vytvoření rezervace nabídky

Pro kontrolu správnosti zadaných údajů je možné využít rekapitulace nově vytvářené rezervace.

5.1.3.3 Zobrazení rezervací uživatele

V uživatelském panelu musí být zobrazení aktuálních rezervací uživatele.

5.1.3.4 Zobrazení rezervací návštěvníků

V uživatelském panelu lze zobrazit rezervace nabídek uživatele.

5.1.3.5 Archiv rezervací

Ukončená rezervace je archivována. V zobrazení rezervací je možné zobrazit archivované rezervace.

5.1.3.6 Blokace peněz na rezervaci

Při vytvoření nové rezervace se nájemníkovi zablokuje finanční částka určená pro tuto rezervaci.

5.1.3.7 Historie transakcí souvisejících s rezervacemi

Schopnost zobrazit historii transakcí souvisejících s rezervacemi.

5.1.3.8 Akce s rezervací

Status rezervace lze měnit a tím kontrolovat průběh rezervace.

5.1.4 Hodnocení

5.1.4.1 Přidání hodnocení rezervované nabídky

Možnost přidat hodnocení nabídky, kterou uživatel navštívil.

5.1.4.2 Zobrazení hodnocení návštěvníků

V uživatelském panelu lze zobrazit všechna hodnocení získaná od návštěvníků.

5.1.4.3 Zobrazení mých hodnocení

V uživatelském panelu lze zobrazit hodnocení udělená uživatelem.

5.1.4.4 Úprava hodnocení

Možnost upravit hodnocení.

5.1.4.5 Zbývá ohodnotit

Zobrazení nabídek, které byly uživatelem navštíveny, ale nebyly ještě hodnoceny.

5.1.5 Zpráva

5.1.5.1 Vytvoření nové zprávy

Uživatel může poslat zprávu jinému uživateli.

5.1.5.2 Zobrazení konverzací

V uživatelském panelu lze zobrazit konverzační vlákna s jinými uživateli.

5.1.5.3 Zobrazení jednotlivé konverzace

Zobrazení výpisu zpráv v konverzačním vlákně.

5.1.5.4 Odpověď v konverzaci

Schopnost odpovědět v daném konverzačním vlákně.

5.1.6 Peněženka

5.1.6.1 Nabití peněženky

Uživatel může přidávat peněžní částky na své konto.

5.1.6.2 Výběr z peněženky

Uživatel si může vybrat peněžní částky ze svého konta.

5.1.6.3 Historie transakcí

Historie transakcí souvisejících s nabitím a výběrem z peněženky.

5.2 Požadavky na reálnou aplikaci

5.2.1 Nastavení změny statusu rezervace podle času

Změna statusu u rezervace na základě času.

5.2.2 Implementace reálného peněžního systému

Možnost reálných plateb a výplat. Např. zavedení API pro platební karty.

5.2.3 Ověření telefonu a e-mailu uživatele

Ověření telefonního čísla a e-mailu uživatelů pro bezpečnější a důvěryhodnější chod aplikace.

5.2.4 Rozhraní pro administrátora

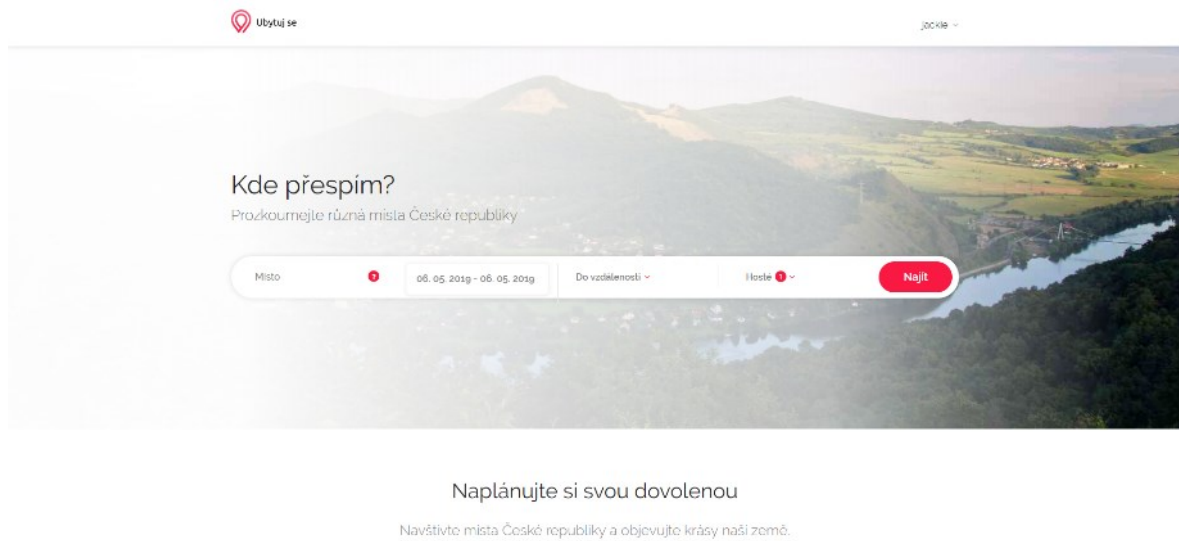
Sekce pro administrátora, ke snadnému řešení nesrovnalostí u rezervací, atp.

5.2.5 Poplatek z uskutečněné rezervace pro majitele aplikace

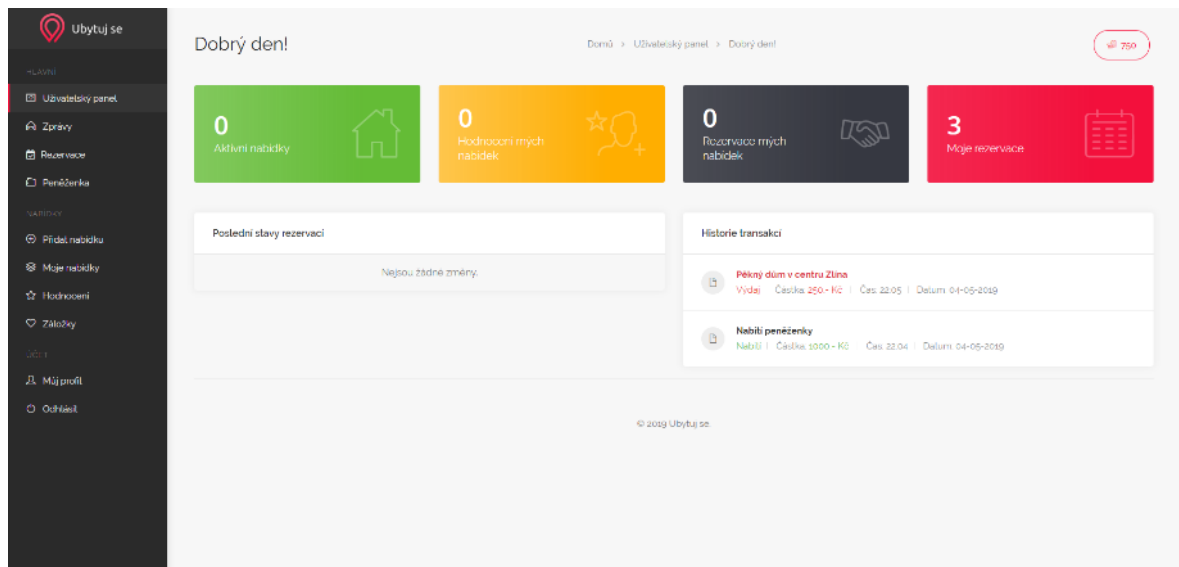
K vytvoření zisku aplikace, je potřeba zavést poplatek za uskutečněné rezervace.

5.3 Návrh uživatelského rozhraní

Uživatelské rozhraní je tvořeno šablonou s názvem Listeo, která je založena pomocí open-source CSS nástroje Bootstrap. Účel šablony se přímo nabízel k použití v aplikaci. Autor práce preferuje úplnou kontrolu nad kódem a díky této šabloně se mohl vzhled aplikace uzpůsobit potřebám aplikace.



Obrázek 12: Domovská stránka



Obrázek 13: Uživatelská sekce

5.4 Návrh databázové struktury

Návrh databázové struktury se mohl uskutečňovat v průběhu tvoření aplikace, díky systému ORM, který velice jednoduše umožňuje měnit strukturu databáze. Neboť jednotlivé tabulky jsou reprezentovány entitami ve zdrojovém adresáři Frameworku (App\Src\Entity). Přidání entity probíhá příkazem v Cmderu:

“php bin\console make:entity“

v projektovém adresáři a následném postupu v průvodci, který vyžaduje zadání jména entity a nastavení atributů entity. V tomto průvodci lze také nastavit relace s jinými entitami.

Vytvořenou entitu je potřeba zavést do databázového schématu. To lze dvěma způsoby: „*migrate*“ nebo „*update schema*“.

U migračního způsobu se vytvoří pomocí *migrations:diff* příkazu migrační soubor s SQL příkazy na změnu tabulek. Následně se tato migrace spustí, tím se příkazy v souboru provedou.

Způsob *update schema* se provádí pomocí příkazu:

```
“php bin\console doctrine:schema:update --force“
```

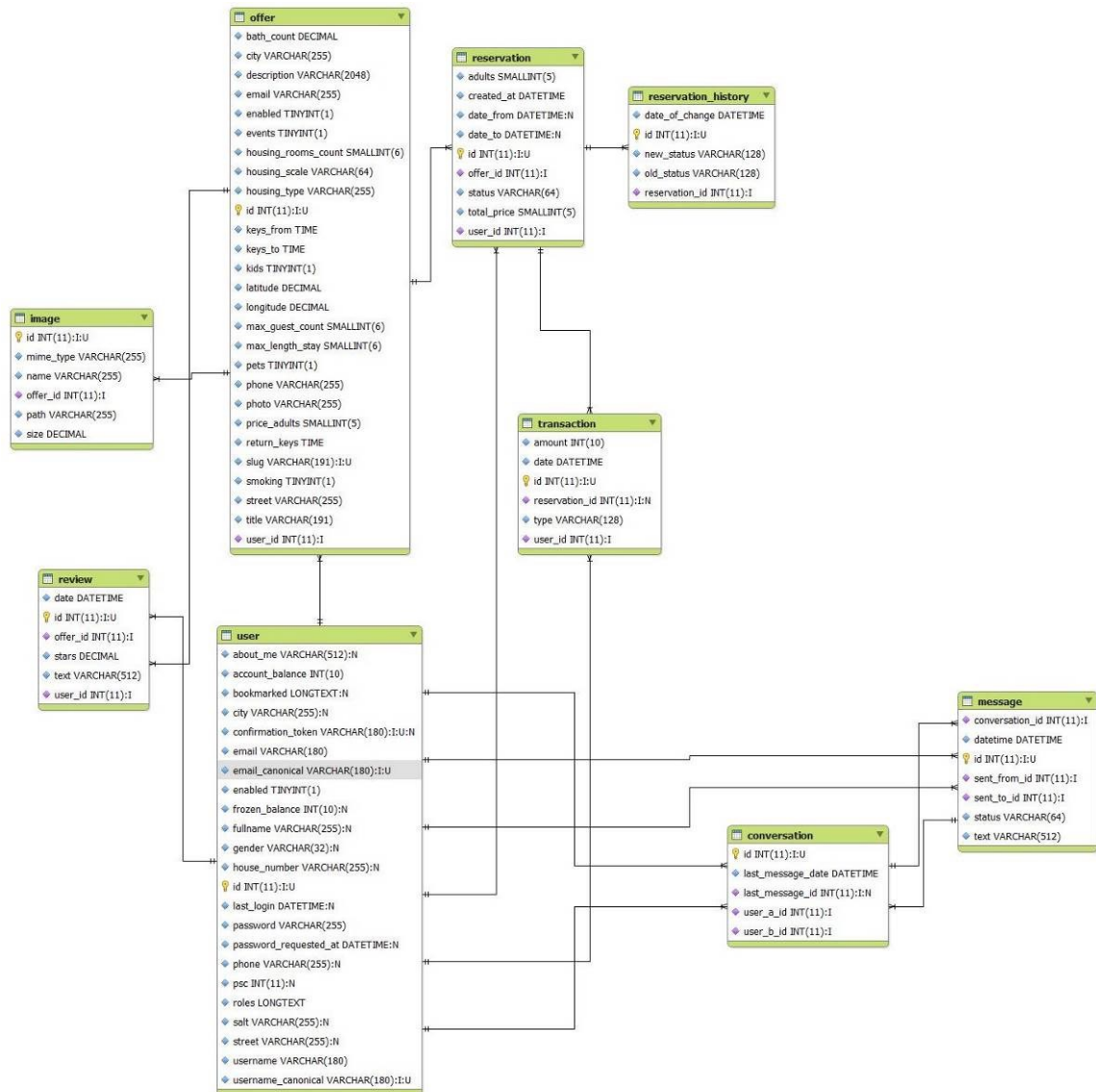
v projektovém adresáři pomocí *Cmderu*. Tento příkaz přetvoří aktuální databázové schéma na schéma, jež popisuje ORM Doctrine v projektu.

Konečné databázové schéma aplikace je vyobrazeno na obrázku (Obr. 14). Databázi tvoří tabulky: *Image*, *Offer*, *Reservation*, *Reservation_history*, *Transaction*, *Review*, *User*, *Conversation* a *Message*.

Databáze má nastaveno kódování *utf8mb4_general_ci*. Kóduje se způsobem UTF8, *mb4* znamená, že znak je kódován 4 bajty. *General* značí obecný jazyk kódování a *ci* – znamená: *case insensitive* – velikost písma se nerozlišuje.

5.4.1 Relační vztahy:

<i>Offer</i> : <i>User</i>	N : 1	Uživatel může mít N nabídek
<i>Offer</i> : <i>Reservation</i>	1 : N	Nabídka může mít N rezervací
<i>Offer</i> : <i>Review</i>	1 : N	Nabídka může mít N recenzí
<i>Offer</i> : <i>Image</i>	1 : N	Nabídka může mít N obrázků
<i>Reservation</i> : <i>User</i>	N : 1	Uživatel může mít N rezervací
<i>Reservation</i> : <i>Reservation_history</i>	1 : N	Rezervace může mít N záznamů v historii
<i>Reservation</i> : <i>Transaction</i>	1 : N	Rezervace může mít N záznamů transakcí
<i>Review</i> : <i>User</i>	N : 1	Uživatel může mít N recenzí
<i>Transaction</i> : <i>User</i>	N : 1	Uživatel může mít N transakcí
<i>User</i> : <i>Conversation</i>	1 : N	Uživatel může mít N konverzací
<i>Conversation</i> : <i>Message</i>	1 : N	Konverzace může mít N zpráv
<i>User</i> : <i>Message</i>	1 : N	Uživatel může mít N zpráv



Obrázek 14: Databázová struktura

5.4.2 User

Uživatelské rozhraní je v aplikaci implementováno pomocí FOSUserBundle, což je Symfony balíček, který za nás řeší registrace a přihlašování uživatelů. Může také řešit editaci účtu, změnu hesla a další akce související s uživatelskými účty. FOSUserBundle ve výchozím stavu implementuje tabulku user s atributy: id, username, username_canonical, password, password_requested_at, enabled, last_login, roles, salt, email, email_canonical, confirmation_token. Kanonická pole jsou pole daných atributů převedeny do lowercase (malých znaků), tyto pole se využívají pro vyhledávání. Zajišťují, že se nenaleznou duplikáty díky rozlišení velkých a malých písmen. Tabulka user byla pro potřeby aplikace rozšířena o

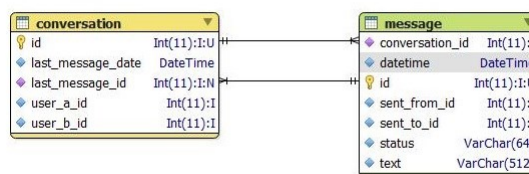
pole: about_me, account_balance, bookmarked, city, frozen_balance, fullname, gender, house_number, phone, psc, street.

Column Name	Data Type
about_me	VarChar(512):N
account_balance	Int(10)
bookmarked	LongText:N
city	VarChar(255):N
confirmation_token	VarChar(180):I:U:N
email	VarChar(180)
email_canonical	VarChar(180):I:U
enabled	TinyInt(1)
frozen_balance	Int(10):N
fullname	VarChar(255):N
gender	VarChar(32):N
house_number	VarChar(255):N
id	Int(11):I:U
last_login	DateTime:N
password	VarChar(255)
password_requested_at	DateTime:N
phone	VarChar(255):N
psc	Int(11):N
roles	LongText
salt	VarChar(255):N
street	VarChar(255):N
username	VarChar(180)
username_canonical	VarChar(180):I:U

Obrázek 15: Tabulka User

5.4.3 Conversation a Message

K řešení posílání zpráv mezi uživateli byly vytvořeny dvě tabulky: conversation a message. Conversation zaznamenává konverzační vlákno (kanál) dvou uživatelů. Message představuje jednotlivou zprávu. Conversation tabulka obsahuje vlastní ID, uživatele A a B, čas poslední zprávy a relační odkaz na poslední zprávu. Tabulka Message obsahuje vlastní ID, datum a čas, do které konverzace patří, status (přečtená / nepřečtená), obsah zprávy, komu byla poslána a od koho byla poslána.



Obrázek 16: Tabulky Conversation a Message

5.4.4 Offer

Tabulka Offer zaznamenává nabídku a její vlastnosti. Obsahuje popis nabídky, titulek, ke kterému uživateli je přiřazena, zda je aktivní, relativní cestu prvního obrázku a další vlastnosti. Za zmínku stojí atribut slug, který je automaticky generován při vytvoření nabídky. Generuje se z atributu title a představuje jednoznačné textové označení nabídky.

offer	
bath_count	Decimal
city	VarChar(255)
description	VarChar(2048)
email	VarChar(255)
enabled	TinyInt(1)
events	TinyInt(1)
housing_rooms_count	Smallint(6)
housing_scale	VarChar(64)
housing_type	VarChar(255)
id	Int(11):I:U
keys_from	Time
keys_to	Time
kids	TinyInt(1)
latitude	Decimal
longitude	Decimal
max_guest_count	Smallint(6)
max_length_stay	Smallint(6)
pets	TinyInt(1)
phone	VarChar(255)
photo	VarChar(255)
price_adults	Smallint(5)
return_keys	Time
slug	VarChar(191):I:U
smoking	TinyInt(1)
street	VarChar(255)
title	VarChar(191)
user_id	Int(11):I

Obrázek 17: Tabulka Offer

5.4.5 Image

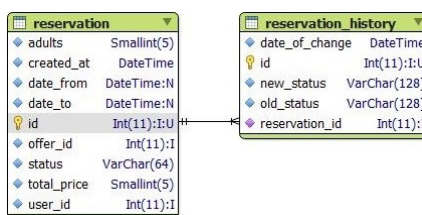
Tabulka Image obsahuje název, příponu, velikost a cestu k obrázku nabídky.

image	
id	Int(11):I:U
offer_id	Int(11):I
path	VarChar(255)
name	VarChar(255)
mime_type	VarChar(255)
size	Decimal

Obrázek 18: Tabulka Image

5.4.6 Reservation a Reservation_history

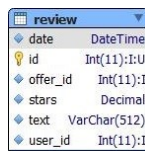
Tabulka Reservation představuje rezervaci uživatele určité nabídky. V tabulce je zaznamenáno: kolik hostů bude ubytováno; kdy byla rezervace vytvořena; od kdy do kdy je plánována; ke které nabídce a uživateli patří; celková cena a aktuální status rezervace. Tabulka reservation_history zaznamenává veškeré změny statusu u rezervace a kdy tyto změny nastaly.



Obrázek 19: Tabulky Reservation a Reservation_history

5.4.7 Review

Tabulka Review slouží k zaznamenání hodnocení nabídky. Atributy tabulky jsou: datum hodnocení, počet hvězdiček, text hodnocení, který uživatel hodnotil, a ke které nabídce hodnocení náleží.

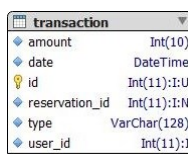


review	
date	DateTime
id	Int(11):I:U
offer_id	Int(11):I
stars	Decimal
text	VarChar(512)
user_id	Int(11):I

Obrázek 20: Tabulka Review

5.4.8 Transaction

Transaction tabulka zaznamenává pohyb peněžních prostředků uživatele. V záznamu je určena: částka, datum transakce, typ transakce, kterému uživateli byl změněn stav peněženky a obsahuje také atribut: reservation_id, který může být nulový. Transakce může souviset s nabitím či výběrem z konta uživatele, ale také může souviset s rezervací. Souvisí-li, je v transakci uvedena.



transaction	
amount	Int(10)
date	DateTime
id	Int(11):I:U
reservation_id	Int(11):I:N
type	VarChar(128)
user_id	Int(11):I

Obrázek 21: Tabulka Transaction

6 IMPLEMENTACE

V kapitole jsou popsány veškeré hlavní implementace komponent této aplikace. K efektivnímu programování využíváme generování kódu pomocí příkazů v Cmderu, generují se nové Controllery, Entity, Formuláře a Security Votery. Využitím tohoto způsobu si ulehčíme mnoho práce, neboť generování vytváří mnoho řádků kódu, které bychom museli psát ručně. Všechny příkazy v Cmderu jsou zadávány v projektovém adresáři aplikace.

6.1 Založení projektu

Nejprve se založí projekt v GitLabu, poté se v Cmderu spustí příkaz:

```
“composer create-project symfony/website-skeleton”
```

Nakonec se nastaví nginx config pro správnou funkci lokálního webserveru.

6.2 Úvodní stránka

Vytvoříme si nový Controller v Cmderu, pomocí příkazu:

```
“php bin\console make:controller”
```

nazveme jej HomepageController. Tento nový Controller vykresluje pouze prázdnou vítací stránku díky příkazu:

```
return $this->render('homepage/index.html.twig');
```

Šablonový soubor, který je vykreslen pomocí funkce render(), začíná rozšířením pomocí příkazu:

```
{% extends 'dashboard/base.html.twig' %}
```

Base.html.twig soubor udává základní strukturu stránky. Do base souboru bylo vloženo základní zobrazení struktury aplikace ze šablony Listeo. V homepage/index souboru byla vytvořena úvodní stránka aplikace ve stylu šablony Listeo.

V HomepageControlleru byly dále vytvořeny dvě funkce: `faq()` a `howItWorks()`, u kterých je nastavena route (URL adresa) pomocí anotace. Obě funkce vykreslují vlastní šablonový soubor, který je rovněž rozšířen: `base.html.twig` souborem.

Zápis anotace:

```
/**  
 * @Route("/jak-to-funguje", name="how-it-works")  
 */
```

6.3 Řešení uživatelského přístupu

K řešení uživatelského přístupu byla použita externí knihovna FOSUserBundle. Která se nejdříve nainstaluje do projektu pomocí Composeru. V Cmderu se provede příkaz:

```
“composer require friendsofsymfony/user-bundle”
```

Následně se tato knihovna nastaví v: fos_user.yaml souboru, ve kterém se určí: jaký databázový systém se používá. Pro obsažení dodatečných atributů uživatelské musíme vytvořit novou entitu User a určit její rozšíření základní uživatelské entity knihovny FOSUserBundle. Pro upravení těchto atributů, musíme vytvořit formulář. Tomuto formuláři nastavíme rozšíření základního formuláře pro editaci v knihovně FOSUserBundle.

```
# app/config/packages/fos_user.yaml
fos_user:
  db_driver: orm
  firewall_name: main
  user_class: App\Entity\User
  profile:
    form:
      type: App\Form>EditProfileType
```

Uživatelskou entitu: User – vytvoříme pomocí Cmderu:

```
“php bin\console make:entity”
```

V této entitě importujeme FOS\UserBundle\Model\User. Třída se tak touto komponentou rozšíří. Podobným způsobem vytvoříme editační formulář. Místo make:entity v příkazu použijeme make:form. Nazveme jej EditProfileType. Do formuláře vložíme funkci getParent(), která navrací ProfileFormType z balíčku FOSUserBundle.

Ještě je nutné nastavit základní bezpečnostní konfigurační soubor security.yaml. V tomto souboru se určí poskytovatel registrace a přihlášení uživatelů: fos_userbundle. Aktualizujeme databázi.

Pro upravení vzhledu vykreslování přihlašovacího, registračního a editačního formuláře uživatele, zkopírujeme vše ze složky “vendor/friendsofsymfony/user-bundle/Resources/views“ a vložíme do “App/templates/bundles/FOSUserBundle“. Následně upravíme dle svého.

6.4 Uživatelská sekce

Pro uživatelskou sekci byl vytvořen nový Controller s názvem DashboardController. Veškeré templates související s uživatelskou sekci jsou vloženy ve složce “templates/dashboard“. Pro uživatelskou sekci byl rovněž vytvořen base.html.twig soubor, který definuje zá-

kladní kostru uživatelské sekce. Na úvodní stránce uživatelské sekce, jsou zobrazeny základní souhrnné informace týkající se uživatelského účtu. DashboardController zpracovává plánované Záložky a Úpravu profilu, kde ale volá FOSUserBundle ke zpracování nově zadaných uživatelských dat.

6.5 Přidání nabídky

První charakteristickou komponentou této aplikace, která byla nutná implementovat je vytvoření nabídky. Pro přehlednost v kódu byl vytvořen nový OfferController, který má na starosti veškerou logiku související s nabídkami. Obsahuje metodu addOffer(), která řeší akci přidání nabídky. Nejprve je potřeba vytvořit novou entitu s názvem Offer. Abychom mohli k nabídkám přidat obrázky, je nutné ještě vytvořit entitu Image, ve které záznamy reprezentují jednotlivé obrázky. U každého záznamu v tabulce Image je sděleno, ke které nabídce patří. Aktualizujeme databázi. Dále je potřeba vytvořit nový formulář, pomocí příkazu v Cmderu:

“php bin\console make:form”

V průvodci sdělíme, že je formulář navázán na nově vytvořenou entitu Offer.

Formulář pro nabídku je tvořen pomocí Symfony komponenty: FormBuilderInterface. V následujícím kódu byly vynechány některé atributy, pro kratší zobrazení kódu. Kód slouží jako nástin mapování atributů entity ve formulářích. V tomto případě mapování atributů entity Offer ve formuláři OfferType.

```
public function buildForm(FormBuilderInterface $builder, array $options)
{
    $builder
        ->add('title', TextType::class)
        ->add('images', FileType::class, [
            'required' => false,
            'mapped' => false,
            'multiple' => true,
            'error_bubbling' => true,
            'attr' => [
                'accept' => 'image/*',
            ]
        ])
        ->add('phone', TextType::class)
        ->add('email', EmailType::class)
        ->add('longitude', HiddenType::class)
        ->add('latitude', HiddenType::class)
        ->add('city', HiddenType::class)
        ->add('street', TextType::class)
        ->add('submit', SubmitType::class, [
            'label' => 'Nabídnout'
        ]);
}
```

Nyní můžeme začít tvořit metodu `addOffer()` v `OfferControlleru`. Nejprve vytvoříme nový objekt entity `Offer`, a poté vytvoříme objekt formuláře typu `OfferType`, který ukládá získaná data do nového objektu entity `Offer`.

Pro zpracování formuláře musíme nejdříve zpracovat `Request` objekt a zkontrolovat zda byl formulář odeslán a obsahuje korektně zadaná data. U nabídky se nastaví uživatel, který nabídku vytvořil a platnost zveřejnění nabídky. Nová nabídka se zapíše do databáze a nastaví se informační hláška o vytvoření nabídky. Nastává přesměrování na výpis nabídek uživatele.

```
$offer = new Offer();
$form = $this->createForm(OfferType::class, $offer);
$form->handleRequest($request);

if ($form->isSubmitted() && $form->isValid()) {
    $offer->setEnabled(true);
    $offer->setUser($user);

    $entityManager = $this->getDoctrine()->getManager();
    $entityManager->persist($offer);
    $entityManager->flush();

    $this->addFlash(
        'notice',
        'Nabídka byla vytvořena.'
    );

    return $this->redirectToRoute('my-offers');
}

return $this->render('dashboard/offer/add-offer.html.twig', [
    'form' => $form->createView(),
    'offer' => $offer
]);
}
```

Funkce `addOffer()` končí vykreslením dané template, kde jako parametr předáme formulářový objekt, u kterého zavoláme metodu `createView()` pro vytvoření zobrazení formuláře na stránce.

6.5.1 Vykreslení formuláře v Twig template

Vykreslení formuláře v Twig template začíná definováním začátku `form_start`, Symfony jej zpracuje a vytvoří HTML značku `<form>`. V závorkách se uvede, o jaký formulář se jedná. `Form_widget` určuje, o jaký prvek z formuláře se jedná. Symfony jej zpracuje a vytvoří HTML značku `<input>`. Lze nadefinovat i specifické HTML atributy, jako `name`, `class`, atd. `Form_label` definuje HTML značku `<label>`. Automaticky je propojena s prvkem formuláře, na který je navázána. Formulář se ukončuje pomocí `form_end`, který automaticky dokončí formulář – vykreslí všechny zbývající prvky z formuláře a definuje CSRF token.

```
{{ form_start(form) }}  
{{ form_widget(form.pets, {'attr': {'name': 'rules'}}) }}  
{{ form_label(form.pets, 'Vhodné pro domácí mazlíčky') }}  
{{ form_end(form) }}
```

6.5.2 Poloha nabídky

Pro určení polohy nabídky je v aplikaci použito API od Mapy.cz [43]. Toto API je zcela zdarma a je možné jej využít i pro komerční zaměření. Velmi velkou výhodou tohoto API je, že umí vyhledávat pomocí čísel popisných, a tím se poloha domů určuje s téměř 100% přesností. U vytvoření nabídky se využívá z tohoto API – Našeptávač polohy, díky kterému se získávají souřadnice zeměpisné šířky a délky. Tyto se následně uloží v nabídce do databáze.

6.5.3 Tvoření atributu slug

Jak již bylo zmíněno v kapitole 5.4.4, v nabídce se generuje atribut slug z atributu title. Generování probíhá díky externí knihovně Symfony s názvem Doctrine Extensions a její části: Sluggable. Nastavuje se velice jednoduše, instalací:

```
“composer require gedmo/doctrine-extensions“
```

Následně se u entity Offer importuje: Gedmo\Mapping\Annotation jako Gedmo. Použije se u atributu slug v anotaci, kde se sdělí pole, ze kterého se tento atribut tvoří.

6.5.4 Obrázky u nabídky

Obrázky v OfferType formuláři nejsou namapovány v entitě, je třeba je načíst z Request objektu. V aplikaci je omezen maximální počet nahraných obrázků na 5, proto se kontroluje délka pole s obrázky. V “App/Config/services.yaml“ si nastavíme globálně parametry pro relativní a absolutní cestu ke složce public/uploads/images a v OfferControlleru nastavíme do proměnných tyto cesty, ke kterým přidáme atribut slug, pro unikátní identifikaci složky dané nabídky. Tuto složku vytvoříme a začneme procházet pole obrázků získaných z Request objektu. Procházíme cyklem foreach a vždy vytvoříme nový objekt entity Image. Nastavíme všechny atributy, zapíšeme do databáze a přesuneme obrázek do složky dané nabídky.

```
$i = 0;
foreach ($images as $image) {
    $file = new Image();
    /** @var File $image */
    $imageExtension = $image->guessExtension();
    $imageSize = $image->getSize();
    $fileName = $offer->getSlug() . '[' . $i . ']';
    $fullFileName = $fileName . '.' . $imageExtension;

    $image->move($uploadDir, $fullFileName);

    $file->setOffer($offer);
    $file->setSize($imageSize);
    $file->setMimeType($imageExtension);
    $file->setName($fileName);
    $file->setPath($uploadDirSmall . '/' . $fullFileName);

    $entityManager->persist($file);
    $i++;
}
$entityManager->flush();
```

6.6 Vyhledávání

Po vytvoření nabídky je další důležitou komponentou vyhledávání nabídek. Pro tuto komponentu je třeba vytvořit nový formulář s názvem SearchType. Vyhledávání nabídek je komponentou přístupnou i anonymním uživatelům, proto je obsažena v HomeController. Vyhledávání zpracovává funkce search(), které jsou předány data z formuláře na úvodní stránce. V případě druhého vyhledávání v řadě z formuláře na stránce vyhledávání.

Vyhledávat nabídky lze podle několika parametrů a to: Místa (povinné), datum námi plánovaného pobytu, počtu hostů a vzdálenost vyhledávání. U místa se opět využívá Našeptávače z API Mapy.cz, z něhož se získá zeměpisná šířka a délka.

Vytváření entit pomocí příkazů v Cmderu nám generuje také repositáře k daným entitám, ve kterých jsou základní vyhledávací dotazy dané entity. Tyto dotazy jsou tvořeny pomocí DQL (Doctrine Query Language) – jazyku ORM: Doctrine. V repositáři lze tvořit dotazy dvěma způsoby: DQL nebo zadáním Nativního SQL dotazu. Výsledkem použití nativního SQL dotazu je pole objektů, které ale není správně zmapováno. Jediným přístupným atributem je ID. Použitím DQL metody získáme správně zmapované pole objektů se všemi atributy přístupnými.

Pro velkou složitost vyhledávacího dotazu, byl dotaz rozdělen na dvě části (dotazy). První dotaz vyhledává nabídky podle vzdálenosti od vyhledávaného místa, počtu hostů a počtu strávených dní (některé nabídky mohou mít limitovaný počet strávených dní). Druhý dotaz vyhledává podle dostupnosti nabídky v zadaném termínu. Poté se na výsledky těchto dvou

dotazů použije matematická operace průnik, čímž získáme nabídky odpovídající parametrům vyhledávání. První dotaz byl tvořen pomocí DQL:

```
public function findSearchData($longitude, $latitude, $radius, $guests,
    $stay, $enabled)
{
    return $this->createQueryBuilder('o')
        ->select('o')
        ->addSelect('111.111 * degrees(acos(least(cos(radians(o.latitude))
* cos(radians(:latitude)) * cos(radians(o.longitude - :longitude)) +
sin(radians(o.latitude)) * sin(radians(:latitude)), 1.0))) AS ratio')
        ->where('(111.111 * degrees(acos(least(cos(radians(o.latitude)) *
cos(radians(:latitude)) * cos(radians(o.longitude - :longitude)) +
sin(radians(o.latitude)) * sin(radians(:latitude)), 1.0))) <= :radius')
        ->andWhere('o.enabled = :enabled')
        ->andWhere('o.max_guest_count >= :guests')
        ->andWhere('o.max_length_stay >= :stay')
        ->setParameter('longitude', $longitude)
        ->setParameter('latitude', $latitude)
        ->setParameter('radius', $radius)
        ->setParameter('enabled', $enabled)
        ->setParameter('guests', $guests)
        ->setParameter('stay', $stay)
        ->orderBy('ratio')
        ->getQuery()
        ->getResult();
}
```

Druhý dotaz byl tvořen pomocí nativního SQL dotazu:

```
public function findAvailable($arrival, $departure)
{
    $rsm = new ResultSetMapping();
    $rsm->addEntityResult(Offer::class, 'o');
    $rsm->addFieldResult('o', 'id', 'id');

    $query = $this->_em->createNativeQuery('select * from offer
where offer.id not in (select r.offer_id from reservation r where
(r.date_from >= :arrival AND r.date_from <= :departure) OR (r.date_to >=
:arrival AND r.date_to <= :departure))', $rsm);
    $query->setParameter('arrival', $arrival);
    $query->setParameter('departure', $departure);

    return $query->getResult();
}
```

Provedením obou dotazů, se získají dvě pole: offersSearchData a offersAvailable.

```
$offersSearchData = $offerRepository->findSearchData($longitude, $latitude,
    $distance, $adults, $stay, 1);
$offersAvailable = $offerRepository->findAvailable($arrival, $departure);
```

Průnik těchto dvou polí za pomoci funkce array_intersect(), objekty v polích se porovnávají pomocí hash funkce. Následně se pole znovu naindexuje.

```
$offers = array_uintersect($offersToArray, $offersAvailable, static
function ($a, $b) {
    return strcmp(spl_object_hash($a), spl_object_hash($b));
});
```

```
$offers = array_values($offers);
```

V prvním dotazu pro získání relativní vzdálenosti nabídky od místa vyhledávání byl použit matematický vzorec pro výpočet vzdálenosti dvou bodů na kouli. Jedná se o Haversine vzorec. [44]

```
111.111 *
DEGREES(ACOS(LEAST(COS(RADIANS(a.Latitude))
* COS(RADIANS(b.Latitude))
* COS(RADIANS(a.Longitude - b.Longitude))
+ SIN(RADIANS(a.Latitude))
* SIN(RADIANS(b.Latitude)), 1.0)))
```

Konstanta 111.1111 ve vzorci představuje počet kilometrů na stupeň zeměpisné šířky. [44]

6.7 Peněženka

Platební systém je v aplikaci důležitou komponentou, díky které jsou uživatelé motivováni pronajímat své nemovitosti. V této demo aplikaci byl zřízen provizorní systém, který umožňuje upravovat stav konta uživatele bez omezení. V reálné aplikaci by měl být implementován reálný platební systém, umožňující navyšování finančního konta uživatele pomocí platebních karet, atp. Výběry z konta uživatele by se schvalovaly administrátorem nebo zautomatizovaly.

Provizorní platební systém, který byl v demo aplikaci zřízen, je ovládán pomocí WalletControlleru. Obsahuje dvě hlavní funkce: `walletAdd()` a `walletWithdraw()`. Funkce `add` přidává peníze na konto a `withdraw` peníze z konta vybírá. Funkce vytváří formulář s částkou, která se testuje, zda je validní a následně se upraví uživatelův stav konta. Stav konta je zobrazen v celé uživatelské sekci, proto je uložen v `Session`. `Session` se aktualizuje s každou změnou konta. Veškeré změny konta vytváří nový záznam o transakci do tabulky `Transaction`.

```
$form = $this->createForm(WalletType::class);
$form->handleRequest($request);

if ($form->isSubmitted() && $form->isValid()) {
    $now = new \DateTime('now');
    $amount = $form->getData();

    $transaction = new Transaction();
    $transaction->setUser($user);
    $transaction->setDate($now);
    $transaction->setType('Nabití');
    $transaction->setAmount($amount['amount']);

    $accBal = $user->getAccountBalance();
    $user->setAccountBalance($accBal + $amount['amount']);

    $session = $this->get('session');
    $session->set('accBalance', $this->getUser()->getAccountBalance());
}
```

6.8 Rezervace

6.8.1 Vytvoření žádosti o rezervaci

Na stránce zobrazení nabídky se vytvoří formulář rezervace, ve kterém se zadává termín rezervace a počet hostů. Po odeslání následuje potvrzení této rezervace. Žádost o rezervaci se vytváří pomocí `bookOffer()` funkce, která je definována v `OfferControlleru`. V této funkci se vypočítá celková cena rezervace, nastaví se rezervaci související informace: žadatel o rezervaci, nabídka, čas vytvoření rezervace a status. Než se nová rezervace zapíše do databáze, je nutné ověřit, zda splňuje podmínky: maximální délky pobytu nabídky, dostatek finančních prostředků na účtu klienta, zda v termínu rezervace není jiná aktivní rezervace, správnost data rezervace (datum odjezdu musí být po datu příjezdu). Nakonec se ještě kontroluje maximální počet hostů, který nabídka povoluje. Jsou-li všechny podmínky rezervace splněny, vytvoří se záznam o rezervaci v tabulce `Reservation_history`. Zarezervuje se celková částka z konta klienta a rezervace se zapíše do databáze.

```

$reservation = new Reservation();
$form = $this->createForm(ReservationType::class, $reservation);
$form->handleRequest($request);

if ($form->isSubmitted() && $form->isValid()) {
    $reservation->setUser($user);
    $reservation->setOffer($offer);
    $reservation->setCreatedAt($today);
    $reservation->setStatus('Čeká');

    $daysInterval = date_diff($reservation->getDateFrom(), $reservation->getDateTo());
    $daysInterval = $daysInterval->format('%d') + ($daysInterval->format('%m') * 30);

    $reservation->setTotalPrice($offer->getPriceAdults() * $reservation->getAdults() * $daysInterval);

    $reservationHistory = new ReservationHistory();
    $reservationHistory->setDateOfChange($today);
    $reservationHistory->setOldStatus('Nevytvořeno');
    $reservationHistory->setNewStatus('Čeká');
    $reservationHistory->setReservation($reservation);

    $user->setFrozenBalance($reservation->getTotalPrice());
    $user->setAccountBalance($user->getAccountBalance() - $reservation->getTotalPrice());
    $userManager->updateUser($user);
}

```

6.8.1.1 Podmínky:

```

if ($daysInterval > $offer->getMaxLengthStay())
if ($user->getAccountBalance() < $reservation->getTotalPrice())
if ($reservation->getDateFrom() >= $reservation->getDateTo())
if ($reservation->getAdults() > $offer->getMaxGuestCount())

```

6.8.2 Akce rezervace

S rezervacemi souvisí akce, které aktualizují status rezervace a řídí její průběh. Tyto akce jsou řízeny pomocí ReservationController.

6.8.2.1 Storno rezervace

Uživatel zaslal novou žádost o rezervaci, ale mohl učinit chybu, či si rezervaci rozmyslel, proto je zde akce stornování rezervace. Ve funkci reservationStorno() se uživateli vrátí rezervované peníze. Vytvoří se nový záznam v Reservation_history tabulce a změní se status dané rezervace na Storno. Již není třeba volat metodu persist na \$reservation, neboť rezervace se v databázi již vyskytuje. Proto stačí zavolat pouze funkci flush().

```

$entityManager = $this->getDoctrine()->getManager();
$reservation->setStatus('Storno');
$entityManager->flush();

```

6.8.2.2 *Zamítnutí rezervace*

Hostiteli se žádost o rezervaci nemusí zamlouvat, proto ji může zamítnout. Funkce `reservationDeny()` funguje téměř totožně jako funkce `reservationStorno()`. Ve funkci se navrátí rezervovaná částka na konto klienta. Vytvoří se záznam v historii rezervací a nastaví se rezervaci status `Zrušeno`.

6.8.2.3 *Schválení rezervace*

Pokud hostitel schválí rezervaci, vytvoří se o tom záznam a automaticky se vytvoří záznam i v tabulce transakcí, kde zákazníkovi přibude výdajová položka.

6.8.2.4 *Převzetí klíčů*

V den počátku rezervace se při načtení stránky s rezervacemi objeví tlačítko u nájemce pro převzetí klíčů, které volá metodu `reservationKeys()`. Metoda změní status rezervace na: `Probíhá`.

6.8.2.5 *Dokončení rezervace*

Posledním dnem rezervace, se při načtení stránky s rezervacemi objeví u hostitele tlačítko pro dokončení, které volá funkci `reservationComplete()`. Tato funkce změní status rezervace ze: `Skončeno` na: `Dokončeno`. Vytvoří u hostitele nový záznam v tabulce transakcí s typem `Příjem`. Aktualizují se konta obou uživatelských stran a vytvoří se nový záznam v tabulce `Reservation_history`. Všechny tyto změny se zapíše do databáze.

6.9 Zprávy

Posíláme-li zprávu jinému uživateli poprvé, využíváme tlačítka `Poslat zprávu`, které se nachází na uživatelském profilu jiného uživatele, či v nabídce, kterou uživatel vytvořil. Po kliknutí toto tlačítko zobrazí modal dialog, ve kterém se vykreslí formulář `MessageType` obsahující text nové zprávy. Tlačítko je zobrazeno pouze, pokud je uživatel přihlášen.

```

<!-- Send message popup -->
<div id="small-dialog" class="zoom-anim-dialog mfp-hide">
  {{ form_start(form) }}
  <div class="small-dialog-header">
    <h3>Poslat zprávu</h3>
  </div>
  <div class="message-reply margin-top-0">
    {{ form_widget(form.text, {'attr': {'placeholder': 'Poslat zprávu
uživateli'}}) }}
    <input type="submit" class="button" value="Poslat zprávu">
  </div>
  {{ form_end(form) }}
</div>

{% if is_granted("IS_AUTHENTICATED_REMEMBERED") %}
  <a href="#small-dialog" class="send-message-to-owner button popup-with-
zoom-anim"><i
class="sl sl-icon-envelope-open"></i> Poslat zprávu</a>
{% endif %}

```

Zobrazení nabídky a uživatelského profilu řeší HomepageController, tudíž pro úsporu kódu můžeme vytvořit v HomepageControlleru funkci sendMessage(\$user, \$message, \$conversationRepository). Kde \$user je uživatel, kterému posíláme zprávu. Text zprávy je \$message a \$conversationRepository předáváme funkci, aby měla přístup k repositáři konverzací. Proto můžeme ve funkcích userProfile a offer použít stejný kód pro poslání zprávy:

```

$message = new Message();
$messageForm = $this->createForm(MessageType::class, $message);
$messageForm->handleRequest($request);

if ($messageForm->isSubmitted() && $messageForm->isValid()) {
    $this->sendMessage($user, $message, $conversationRepository);

    return $this->redirectToRoute('messages');
}

```

Funkce sendMessage() zkontroluje, zda uživatel neposílá zprávu sám sobě a nastaví zprávě status: Nepřečtená, čas poslání, kdo zprávu posílá a komu se zpráva posílá. Dále se musí zprávě nastavit vlákno konverzace. Nejprve se vyhledá existující vlákno v repositáři. Pokud konverzační vlákno neexistuje, vytvoří se nové. Existuje-li, nastaví se v nové zprávě. Dále se upraví poslední zpráva této konverzace a všechno se uloží do databáze.

```

$conversationA = $conversationRepository->findBy(['user_a' => $user,
'user_b' => $this->getUser()]);
$conversationB = $conversationRepository->findBy(['user_a' => $this-
>getUser(), 'user_b' => $user]);

if (empty($conversationA) && empty($conversationB)) {
    $conversation = new Conversation();
    $conversation->setUserA($user);
    $conversation->setUserB($this->getUser());
} else {
    if (empty($conversationA)) {
        $conversation = $conversationB[0];
    } else {
        $conversation = $conversationA[0];
    }
}

```

6.10 Hodnocení nabídek

Hodnotit lze pouze nabídky, u kterých má uživatel úspěšně proběhlou rezervaci. Proto bylo vytvořeno v hodnocení tlačítko: Zbývá ohodnotit...? Které nás přesměruje na funkci toReview(), ve které se naleznou v repositáři rezervací uživatelovi rezervace, které jsou již Skončené a Dokončené. Z repositáře těchto rezervací, se vytvoří pole s nabídkami. A porovná se s polem druhým, které obsahuje nabídky z repositáře hodnocených nabídek uživatele. Nabídky z prvního pole, které se nevyskytují v poli druhém je možno ohodnotit, a proto se zobrazí na stránce k ohodnocení. Přidání hodnocení přesměruje na formulář, který ukládá data do databáze. Hodnocení lze editovat na stránce s uživatelskými hodnoceními.

```

$reservations = $reservationRepository->findBy([
    'user' => $this->getUser(),
    'status' => ['Dokončeno', 'Skončeno'],
    ['date_to' => 'ASC']
]);

$countReservations = count($reservations);
$offers = [];
for ($i = 0; $i < $countReservations; $i++) {
    $offers[$i] = $reservations[$i]->getOffer();
}

$reviews = $reviewRepository->findBy(['user' => $this->getUser()]);

$countReviews = count($reviews);
$reviewedOffers = [];
for ($i = 0; $i < $countReviews; $i++) {
    $reviewedOffers[$i] = $reviews[$i]->getOffer();
}

if (empty($reviewedOffers)) {
    $result = $offers;
} else {
    // Ještě neohodnocené nabídky uživatelem
    $result = array_diff($offers, $reviewedOffers);
}

```

7 ZABEZPEČENÍ

7.1 Omezení přístupu

Přístup se zamezuje anonymním uživatelům do sekcí přístupných pouze pro registrované uživatele. Jasným příkladem je uživatelský panel, který je ovládán pomocí DashboardControlleru. Jedním ze způsobů zamezení přístupu je pomocí anotace. Na tento controller se zavolá Symfony komponenta: Security a nastaví se povolení přístupu pouze pro přihlášené uživatele. Pomocí této anotace můžeme zamezit přístup i k jakékoli funkci.

```
/**
 * Class DashboardController
 * @package App\Controller
 * @Security("is_granted('IS_AUTHENTICATED_REMEMBERED')")
 */
class DashboardController
```

7.1.1 Security voters

Bezpečnostní votery rovněž omezují přístup, ale lze je specifikovat na určitý typ akce. Například deaktivace nabídky, u tohoto typu akce se musí zkontrolovat, že uživatel požadující deaktivaci je opravdu vlastníkem této nabídky. Voter se generuje v Cmderu:

```
“php bin\console make:voter”
```

Nový voter se nazve OfferVoter a musí se ručně nastavit. Obsahuje dvě funkce supports() a voteOnAttribute(). Supports vyžaduje dva parametry – atribut a subjekt. Atribut je název akce př. OFFER_DEACTIVATE a subjekt je Entita, která je testována. Tato funkce kontroluje, zda se akce nachází v akcích definovaných voterem a zda je subjekt opravdu entita definovanou ve voteru. Funkce voteOnAttribute požaduje rovněž atribut a subjekt. K tomu si automaticky ještě vyžaduje token (je v něm uložen např. aktuálně přihlášený uživatel). Tato funkce rozhoduje o přístupu.

V případě deaktivace nabídky se ve Voteru nadefinuje atribut OFFER_DEACTIVATE a subjekt třídy Offer a dále podmínka, že aktuální uživatel musí být uživatelem z nabídky.

```
switch ($attribute) {
    case 'OFFER_ACTIVATE':
        return ($offer->getUser()->getId() === $user->getId());

    case 'OFFER_DEACTIVATE':
        return ($offer->getUser()->getId() === $user->getId());
}
```


7.2 Ochrana formulářů

Formuláře vytvářené pomocí Symfony Frameworku, přesněji jeho komponenty FormBuilderInterface jsou automaticky zabezpečeny CSRF Tokenem (Cross-site request forgery). Jedná se o náhodně vygenerovaný řetězec, který platí jen pro aktuálního uživatele a je po každé jiný. Symfony si tento řetězec vygeneruje a uloží, ale zároveň je vložen také do automaticky generovaného skrytého pole formuláře u uživatele. Při zpracování formuláře se porovnává totožnost obou tokenů. Při shodě se jedná o uživatele, kterému byl token předán. Při neshodě se jedná o cizího uživatele, tedy útočníka.

Vstupy ve formulářích jsou rovněž escapovány, a tím jsou ochráněny vůči SQL Injection. Proto např. nelze obejít přihlašování zadáním do polí 'OR 1 = 1;'.

7.3 Hashování uživatelského hesla

Díky balíčku FOSUserBundle, který se stará o uživatelský přístup, je při každé registraci uživatele heslo nového uživatele zahashováno a do databáze se ukládá pouze tento hash. Při přihlašování uživatele se tedy porovnávají hashe hesel. Jedná se o skvělé zabezpečení, pokud by byla databáze hacknuta. Ukradeny by byly pouze hashe hesel. Uživatelská hesla by byla ochráněna. Tento způsob také ochraňuje před zneužitím hesla ze strany administrátora.

ZÁVĚR

Při zpracování bakalářské práce se autor naučil mnoho z aktuálních metodik tvoření webových aplikací. Velice obohacující byla práce s nástrojem Git. Tento skvělý nástroj pro správu verzí je široce používán mezi developery aplikací. V praxi se s ním jistě setkáte, a proto je velkou výhodou s tímto nástrojem umět pracovat.

Také práce s objektově orientovaným programováním (OOP) dodává jiný rozměr při přemýšlení. Totéž platí i u objektově orientovaného relačního mapování, kdy se tabulky reprezentují objekty (entitami). Databázový systém Doctrine, s kterým úzce spolupracuje Symfony Framework, je založen na této objektové reprezentaci. Za pomoci systému Doctrine se můžete soustředit jen na návrh databáze a nemusíte se zabývat propojováním databáze s aplikací, či jakýmkoli jiným nastavováním, které souvisí s databázemi.

Samotný Framework Symfony je snadný na pochopení a použití. Spolu s IDE: PhpStorm, velice usnadňuje programování. Symfony řeší mnoho postupů místo programátora a dále jeho kód je minimalistický. PhpStorm hodně příkazů doplňuje za programátora a umí také generovat kód (getter, setter, ...). Tyto skvělé nástroje mnohonásobně zvyšují programátorovu produktivitu a šetří mnoho cenného času.

V teoretické části byly zpracovány následující části. Rozbor stávajících řešení, z nichž nejpodobnější řešení bylo Airbnb. Webové technologie, se kterými se pracovalo. A hlavně princip funkce a popis Frameworku Symfony, který lze označit jako základní kámen bakalářské práce.

Praktická část byla zaměřena na realizaci webové aplikace. Nejprve byla zpracována analýza a návrh řešení. Dále byla aplikace naprogramována, přičemž se v průběhu tvořilo databázové schéma.

Vytvořená aplikace je téměř připravena do ostrého provozu. Pokud se v aplikaci naprogramují zmíněné požadavky pro reálnou aplikaci z kapitoly 5.2, bude aplikace schopna fungovat v reálném prostředí a generovat zisk.

Aplikace byla nahrána na doménu: **ubytujse.obrtlik.cz**

SEZNAM POUŽITÉ LITERATURY

- [1] Rekreační pronájmy, domovy, zážitky a místa – Airbnb [online], 2019. Ringsend, Dublin 4: Airbnb [cit. 2019-02-22]. Dostupné z: <https://www.airbnb.cz/>
- [2] Nejvíce levného ubytování - ČR, SR, svět | LevnéUbytování.net [online], 2019. Brno-Bystrc: Robert Vystrčil [cit. 2019-02-22]. Dostupné z: <http://www.levneubytovani.net/>
- [3] Ubytovani.NET - Vyhledávač ubytování v ČR [online], 2019. Holešovice: SPA TRADE SERVICE CZ [cit. 2019-02-22]. Dostupné z: <https://www.ubytovani.net/>
- [4] Booking.com: 28 398 665 ubytovacích jednotek v hotelech a dalších typech ubytování po celém světě. Rezervujte hotel hned! [online], 2019. Amsterdam: Booking Holdings [cit. 2019-02-22]. Dostupné z: <https://www.booking.com>
- [5] What is a Website Domain? - Define a Website Domain, 2019. Boston SEO Marketing Services Company - Brick Marketing [online]. Boston Massachusetts: Brick Marketing [cit. 2019-02-22]. Dostupné z: <https://www.brickmarketing.com/define-website-domain.htm>
- [6] What is Web Hosting? - Web Hosting - Beginner's Guide – Website.com, 2019. Website.com [online]. British Columbia, Canada: Website.com Solutions [cit. 2019-02-22]. Dostupné z: <https://www.website.com/beginnerguides/webhosting/6/1/what-is-web-hosting?.ws>
- [7] What Is NGINX? A Basic Look at What It Is and How It Works, 2019. Kinsta Managed WordPress Hosting [online]. London: Kinsta [cit. 2019-02-22]. Dostupné z: <https://kinsta.com/knowledgebase/what-is-nginx/>
- [8] Introduction (Symfony Docs), 2019. Symfony, High Performance PHP Framework for Web Development [online]. Clichy Cedex: SensioLabs [cit. 2019-02-22]. Dostupné z: https://symfony.com/doc/current/create_framework/introduction.html
- [9] Installing Composer (Symfony Docs), 2019. Symfony, High Performance PHP Framework for Web Development [online]. Clichy Cedex: SensioLabs [cit. 2019-02-22]. Dostupné z: <https://symfony.com/doc/current/setup/composer.html>
- [10] Introduction - Composer, 2019. Composer [online]. Berlin: Nils Adermann, Jordi Boggiano [cit. 2019-02-22]. Dostupné z: <https://getcomposer.org/doc/00-intro.md>

- [11] CMDER is a Console Emulator for Windows systems, 2019. TheWindowsClub [online]. Indie: The Windows Club [cit. 2019-02-22]. Dostupné z: <https://www.thewindowsclub.com/cmdr-console-emulator-windows>
- [12] Simple words for a GitLab Newbie | GitLab, 2015. GitLab [online]. Ukraine: GitLab [cit. 2019-02-22]. Dostupné z: <https://about.gitlab.com/2015/05/18/simple-words-for-a-gitlab-newbie/>
- [13] 3 Reasons Students Should Use Git and GitKraken, 2017. Dev Blog by Axosoft [online]. Scottsdale, Arizona: Axosoft [cit. 2019-02-22]. Dostupné z: <https://blog.axosoft.com/3-reasons-students-git-gitkraken/>
- [14] What is Git: become a pro at Git with this guide | Atlassian Git Tutorial, 2019. Git Tutorials and Training | Atlassian Git Tutorial [online]. Amsterdam: Atlassian [cit. 2019-02-22]. Dostupné z: <https://www.atlassian.com/git/tutorials/what-is-git>
- [15] DevOps Tutorial: Complete Beginners Training, 2019. Meet Guru99 - Free Training Tutorials & Video for IT Courses [online]. India: Guru99 Tech Pvt [cit. 2019-02-22]. Dostupné z: <https://www.guru99.com/devops-tutorial.html#5>
- [16] Learning Git with GitKraken, 2017. Dev Blog by Axosoft [online]. Scottsdale, Arizona: Axosoft [cit. 2019-02-22]. Dostupné z: <https://blog.axosoft.com/learning-git-with-gui/>
- [17] What is HTML (Hypertext Markup Language)? - Definition from WhatIs.com, 2018. TheServerSide.com [online]. Newton, Massachusetts: TechTarget [cit. 2019-02-22]. Dostupné z: <https://www.theserverside.com/definition/HTML-Hypertext-Markup-Language>
- [18] What is CSS (cascading style sheets)? - Definition from WhatIs.com, 2018. TheServerSide.com [online]. Newton, Massachusetts: TechTarget [cit. 2019-02-22]. Dostupné z: <https://www.theserverside.com/definition/cascading-style-sheet-CSS>
- [19] PHP: Hypertext Preprocessor, 2019. PHP: Hypertext Preprocessor [online]. Dánsko: PHP Group [cit. 2019-02-22]. Dostupné z: <http://php.net>
- [20] What is PHP? - Quora, 2015. Quora [online]. Cape Town: Quora [cit. 2019-02-22]. Dostupné z: <https://www.quora.com/What-is-PHP>
- [21] PHP: What is PHP? - Manual, 2019. PHP: Hypertext Preprocessor [online]. Dánsko: PHP Group [cit. 2019-02-22]. Dostupné z: <http://php.net/manual/en/intro-what-is.php>

- [22] An Introduction to JavaScript, 2017. The Modern Javascript Tutorial [online]. Moscow, Russia: Kantor Ilya Alexandrovich [cit. 2019-02-22]. Dostupné z: <https://javascript.info/intro>
- [23] What is MariaDB? - Definition from WhatIs.com, 2018. SearchDataManagement [online]. Newton, Massachusetts: TechTarget [cit. 2019-02-22]. Dostupné z: <https://searchdatamanagement.techtarget.com/definition/MariaDB>
- [24] What is maria db? - Quora, 2018. Quora [online]. India: Quora [cit. 2019-02-22]. Dostupné z: <https://www.quora.com/What-is-maria-db>
- [25] What does heidisql mean - Definition of heidisql - Word finder, 2019. Word Finder [online]. San Francisco: Wikimedia Foundation [cit. 2019-02-22]. Dostupné z: <https://findwords.info/term/heidisql>
- [26] HeidiSQL - MariaDB, MySQL, MSSQL and PostgreSQL made easy, 2019. HeidiSQL [online]. Germany: Ansgar Becker [cit. 2019-02-22]. Dostupné z: <https://www.heidisql.com/>
- [27] PhpStorm Reviews: Overview, Pricing and Features, 2019. Trusted SaaS Software Reviews | FinancesOnline.com [online]. Wlynkowko, Poland: INTERCORP MEDIA sp. z o.o. sp. k. [cit. 2019-02-22]. Dostupné z: <https://reviews.financesonline.com/p/phpstorm>
- [28] Symfony at a Glance, 2019. Symfony, High Performance PHP Framework for Web Development [online]. Clichy Cedex: SensioLabs [cit. 2019-02-22]. Dostupné z: <https://symfony.com/at-a-glance>
- [29] Chapter 1 - Introducing Symfony (1_4), 2019. Symfony, High Performance PHP Framework for Web Development [online]. Clichy Cedex: SensioLabs [cit. 2019-02-22]. Dostupné z: https://symfony.com/legacy/doc/gentle-introduction/1_4/en/01-Introducing-Symfony
- [30] Model MVC, 2019. In: Symfony, High Performance PHP Framework for Web Development [online]. Clichy Cedex: SensioLabs [cit. 2019-02-22]. Dostupné z: https://symfony.com/legacy/images/book/1_4/F0201.png
- [31] Chapter 2 - Exploring Symfony's Code (1_4), 2019. Symfony, High Performance PHP Framework for Web Development [online]. Clichy Cedex: SensioLabs [cit. 2019-02-22]. Dostupné z: https://symfony.com/legacy/doc/gentle-introduction/1_4/en/02-Exploring-Symfony-s-Code

- [32] MVC (Model-View-Controller) Definition, 2018. The Tech Terms Computer Dictionary [online]. Wheaton: Sharpened Productions [cit. 2019-02-22]. Dostupné z: <https://techterms.com/definition/mvc>
- [33] MVC diagram, 2018. In: The Tech Terms Computer Dictionary [online]. Wheaton: Sharpened Productions [cit. 2019-02-22]. Dostupné z: https://cdn.techterms.com/img/lg/mvc_1321.png
- [34] Symfony and HTTP Fundamentals (Symfony Docs), 2019. Symfony, High Performance PHP Framework for Web Development [online]. Clichy Cedex: SensioLabs [cit. 2019-02-22]. Dostupné z: https://symfony.com/doc/current/introduction/http_fundamentals.html
- [35] The MVC pattern, 2019. In: Symfony, High Performance PHP Framework for Web Development [online]. Clichy Cedex: SensioLabs [cit. 2019-02-22]. Dostupné z: https://symfony.com/legacy/images/book/1_4/F0201.png
- [36] Create your First Page in Symfony (Symfony Docs), 2019. Symfony, High Performance PHP Framework for Web Development [online]. Clichy Cedex: SensioLabs [cit. 2019-02-22]. Dostupné z: https://symfony.com/doc/current/page_creation.html
- [37] Introduction - Documentation - Twig - The flexible, fast, and secure PHP template engine, 2019. Twig - The flexible, fast, and secure PHP template engine [online]. Clichy Cedex: SensioLabs [cit. 2019-02-22]. Dostupné z: <https://twig.symfony.com/doc/2.x/intro.html>
- [38] Twig Internals - Documentation - Twig - The flexible, fast, and secure PHP template engine, 2019. Twig - The flexible, fast, and secure PHP template engine [online]. Clichy Cedex: SensioLabs [cit. 2019-02-22]. Dostupné z: <https://twig.symfony.com/doc/2.x/internals.html>
- [39] Twig - The flexible, fast, and secure PHP template engine, 2019. Twig - The flexible, fast, and secure PHP template engine [online]. Clichy Cedex: SensioLabs [cit. 2019-02-22]. Dostupné z: <https://twig.symfony.com/>
- [40] Escape - Documentation - Twig - The flexible, fast, and secure PHP template engine, 2019. Twig - The flexible, fast, and secure PHP template engine [online]. Clichy Cedex: SensioLabs [cit. 2019-02-22]. Dostupné z: <https://twig.symfony.com/doc/2.x/filters/escape.html>

- [41] Chapter 1 - Introducing Symfony (1_4), 2019. Symfony, High Performance PHP Framework for Web Development [online]. Clichy Cedex: SensioLabs [cit. 2019-02-22]. Dostupné z: https://symfony.com/legacy/doc/gentle-introduction/1_4/en/01-Introducing-Symfony
- [42] Profiler (Symfony Docs), 2019. Symfony, High Performance PHP Framework for Web Development [online]. Clichy Cedex: SensioLabs [cit. 2019-02-22]. Dostupné z: <https://symfony.com/doc/current/profiler.html>
- [43] API Mapy.cz [online], 2019. Praha: Seznam.cz [cit. 2019-05-11]. Dostupné z: <https://api.mapy.cz/>
- [44] Find distance between two points using latitude and longitude in mysql - Stack Overflow, 2014. Stack Overflow - Where Developers Learn, Share, & Build Careers [online]. London: Stack Exchange [cit. 2019-05-12]. Dostupné z: <https://stackoverflow.com/questions/24370975/find-distance-between-two-points-using-latitude-and-longitude-in-mysql>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

AJAX	Asynchronous JavaScript and XML
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
CI/CD	Continuous Integration & Delivery
CPU	Central Processing Unit
CSS	Cascading Style Sheets
DBMS	Database Management System
DevOps	Development Operations
DVCS	Distributed Version Control System
DQL	Doctrine Query Language
GPL	General Public License
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IDE	Integrated Development Environment
MVC	Model-View-Controller
OOP	Object-oriented programming
ORM	Object-relational mapping
PHP	PHP: Hypertext Preprocessor
RAD	Rapid-application development
SQL	Structured Query Language
URL	Uniform Resource Locator
VCS	Version Control System
W3C	The World Wide Web Consortium

SEZNAM OBRÁZKŮ

<i>Obrázek 1: Airbnb [1]</i>	12
<i>Obrázek 2: Airbnb – nabídka ubytování [1]</i>	13
<i>Obrázek 3: LevneUbytovani.net [2]</i>	14
<i>Obrázek 4: LevneUbytovani – nabídka ubytování [2]</i>	15
<i>Obrázek 5: Ubytovani.net [3]</i>	15
<i>Obrázek 6: Ubytovani.net – nabídka ubytování [3]</i>	16
<i>Obrázek 7: Booking.com [4]</i>	17
<i>Obrázek 8: Booking – nabídka ubytování [4]</i>	18
<i>Obrázek 9: Model MVC [30]</i>	29
<i>Obrázek 10: MVC diagram [33]</i>	31
<i>Obrázek 11: Pracovní tok Symfony [35]</i>	33
<i>Obrázek 12: Domovská stránka</i>	44
<i>Obrázek 13: Uživatelská sekce</i>	44
<i>Obrázek 14: Databázová struktura</i>	46
<i>Obrázek 15: Tabulka User</i>	47
<i>Obrázek 16: Tabulky Conversation a Message</i>	47
<i>Obrázek 17: Tabulka Offer</i>	48
<i>Obrázek 18: Tabulka Image</i>	48
<i>Obrázek 19: Tabulky Reservation a Reservation_history</i>	48
<i>Obrázek 20: Tabulka Review</i>	49
<i>Obrázek 21: Tabulka Transaction</i>	49

PŘÍLOHY

- I. Zdrojove-kody.zip