

Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačních technologií



Diplomová práce

Tvorba webových aplikací v Nette Framework

Bc. Adam Kořenek

© 2018 ČZU v Praze

ZADÁNÍ DIPLOMOVÉ PRÁCE

Bc. Adam Kořenek

Informatika

Název práce

Tvorba webových aplikací v NETTE Framework

Název anglicky

Web application development using NETTE Framework

Cíle práce

Cílem této práce bude vytvořit informační systém na webu, který bude zprostředkovávat prodej náhradních dílů pro automobily a vytvořit tím ucelený pohled na vývoj webových aplikací s využitím PHP frameworku Nette. Hlavní motivací pro vývoj této webové stránky je nedostatek přehledných webových stránek, které by měly vhodné třídění inzerátů do kategorií, tak aby bylo snadné vyfiltrovat požadované díly. K tvorbě serverové části portálu se bude využívat programovací jazyk PHP, Framework Nette a pro uchování dat se využije databáze MySQL. Pro vývoj UI se použije HTML 5, CSS, Java Script, knihoven JQuery a front-end framework Bootstrap. Za pomoci uvedených technologií bude vytvořeno přehledné responzivní UI, které bude uživatelsky přívětivé jak na PC, tak tabletech či smartphonech.

Metodika

V rešeršní části této práce budou zanalyzovány veškeré technologie související s vývojem webového portálu, tak aby bylo možné je vhodně implementovat v praktické části. V praktické části bude provedena analýza funkčních požadavků z hlediska rolí jednotlivých uživatelů portálu. Na základě této analýzy se veškeré funkční požadavky implementují do UX návrhu wireframů, podle kterých se realizuje finální UI. Pro uchování dat byla zvolena relační databáze MySQL, následujícím krokem bude návrh této databáze s ohledem na normalizaci databázového modelu. Předposlední fází je implementace MVC architektury a naprogramování back-endu za pomoci Nette PHP frameworku. V poslední fázi se otestují veškeré funkční části portálu a odladí se poslední chyby. Na závěr práce bude provedeno zhodnocení výsledků, dosažení stanovených cílů a návrh případného rozšíření portálu do budoucna.

Doporučený rozsah práce

60 – 80 stran

Klíčová slova

webová aplikace, PHP, Nette, HTML, CSS, JavaScript, JQuery, Bootstrap

Doporučené zdroje informací

BÖHMER, M. *Návrhové vzory v PHP : [23 vzorových postupů pro rychlejší vývoj]*. Brno: Computer Press, 2012. ISBN 978-80-251-3338-5.

CASTRO, E. – HYSLOP, B. *HTML5 a CSS3 : názorný průvodce tvorbou WWW stránek*. Brno: Computer Press, 2012. ISBN 978-80-251-3733-8.

DELISLE, M. *phpMyAdmin : efektivní správa MySQL*. Brno: Zoner Press, 2004. ISBN 80-86815-09-9.

POKORNÝ, J. – GILMORE, W J. *Velká kniha PHP 5 a MySQL : kompendium znalostí pro začátečníky i profesionály*. Brno: Zoner Press, 2011. ISBN 978-80-7413-163-9.

VÁVRŮ, J. *jQuery Mobile*. Brno: Computer Press, 2013. ISBN 978-80-251-3811-3.

Předběžný termín obhajoby

2017/18 LS – PEF

Vedoucí práce

Ing. Václav Lohr, Ph.D.

Garantující pracoviště

Katedra informačních technologií

Elektronicky schváleno dne 29. 1. 2018

Ing. Jiří Vaněk, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 30. 1. 2018

Ing. Martin Pelikán, Ph.D.

Děkan

V Praze dne 29. 03. 2018

Čestné prohlášení

Prohlašuji, že svou diplomovou práci "Tvorba webových aplikací v Nette Framework" jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autor uvedené diplomové práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 29.3.2018

Poděkování

Rád bych touto cestou poděkoval Ing. Václav Lohr, Ph.D. za cenné připomínky a odborné rady, kterými přispěl k vypracování této diplomové práce.

Tvorba webových aplikací v Nette Framework

Abstrakt

Cílem této práce je analýza PHP frameworků, front-end frameworků, relačních databází, návrh a implementace webové aplikace - inzertního portálu. Diplomová práce přibližuje problematiku vývoje webových aplikací a zaměřuje se na charakteristiky a specifika třech PHP frameworků (Nette, Laravel, Symfony), třech front-end frameworků (Bootstrap, Foundation, Semantic UI), které byly popsány a porovnány na základě příslušných kritérií. Praktická část práce obsahuje výběr vhodného frameworku pro realizaci webové aplikace a popisuje jejich návrh, vývoj a nasazení.

Klíčová slova: vývoj webových aplikací, PHP, Nette, Laravel, Symfony, Bootstrap, Foundation, Semantic UI, MySQL, PostgreSQL, SQLite

Web apps development using Nette framework

Abstract

The aim of this thesis was to analyze PHP frameworks, front-end frameworks, relational databases, design and implementation of web application - advertising portal. This diploma thesis describes the development of web applications and focuses on the characteristics and specifics of three PHP frameworks (Nette, Laravel, Symfony), three front-end frameworks (Bootstrap, Foundation, Semantic UI) that have been described and compared on the basis of relevant criteria. The practical part of the thesis contains a selection of the appropriate framework for the implementation of the web application and describes their design, development and deployment.

Keywords: web apps development, PHP, Nette, Laravel, Symfony, Bootstrap, Foundation, Semantic UI, MySQL, PostgreSQL, SQLite

Obsah

1	Úvod	4
2	Cíl práce a metodika	5
2.1	Cíle práce.....	5
2.2	Metodika.....	5
3	Teoretická východiska	6
3.1	Webové PHP frameworky	6
3.1.1	PHP.....	6
3.1.2	Návrhové vzory	6
3.1.3	MVC	6
3.1.4	Kritéria pro porovnání PHP frameworku	8
3.1.5	Laravel framework.....	9
3.1.6	Nette.....	12
3.1.7	Symfony.....	15
3.1.8	Porovnání PHP frameworků.....	18
3.1.9	Tabulka porovnání PHP frameworků.....	19
3.2	Front-end frameworky	20
3.2.1	HTML5	20
3.2.2	CSS.....	20
3.2.3	Responzivní design.....	21
3.2.4	DOM.....	22
3.2.5	JavaScript.....	22
3.2.6	Ajax	23
3.2.7	Kritéria pro porovnání front-end frameworků	23
3.2.8	Bootstrap.....	24
3.2.9	Foundation	26
3.2.10	Semantic UI.....	28
3.2.11	Porovnání front-end frameworků.....	30
3.2.12	Tabulka porovnání front-end frameworků.....	31
3.3	Relační database	32
3.3.1	Normalizace	33
3.3.2	Kritéria pro porovnání relačních databází	34
3.3.3	Popularita vybraných databází.....	34
3.3.4	MySQL	36

3.3.5	PostgreSQL	38
3.3.6	SQLite.....	40
3.3.7	Porovnání relačních databází.....	42
3.3.8	Tabulka porovnání relačních databází.....	43
3.4	UX.....	44
3.4.1	Wireframe	44
3.4.2	Uživatelské testování	44
3.4.3	Persony.....	45
3.4.4	Scénáře a storyboard.....	45
3.5	Use case.....	46
3.5.1	User stories.....	46
3.5.2	Use case scénář.....	46
4	Vlastní práce.....	47
4.1	User stories inzertního portálu.....	48
4.2	Use case scénáře inzertního portálu.....	49
4.2.1	Zákazník.....	49
4.2.2	Prodejce	52
4.2.3	Use case diagram.....	53
4.3	Návrh wireframů inzertního portálu	54
4.4	Návrh MySQL databáze	57
4.4.1	MySQL Workbench.....	57
4.4.2	Popis databáze	57
4.4.3	Návrh MySQL databáze.....	58
4.5	Implementace inzertního portálu za pomoci Nette Framework	59
4.5.1	Model.....	59
4.5.2	Presenter.....	60
4.5.3	View.....	61
4.6	Implementace front-end framework Bootstrap.....	62
5	Závěr	66
6	Použitá literatura:	67
7	Seznam obrázků	69
8	Seznam tabulek.....	70

1 Úvod

Od zavedení Web 2.0, webové aplikace předbíhají internet a stávají se více populárnější. Rozvoj takových aplikací se stal složitějším, a přibývaly požadavky, které se velmi často opakovali. Během vývoje se programátoři často setkávali s problémy, které již dříve byly vyřešeny. Aby tyto problémy byly zobecněny, začaly se objevovat různé šablony a nápady určené k řešení tohoto opakujícího se problému. Tyto návrhové vzory byly vytvořeny za účelem zachování daného standardu a uspořádání ve zdrojovém kódu při vytváření komplexních informačních systémů. [1]

Návrhové vzory jsou úzce spojeny s frameworky. Cílem frameworku je zmírnit režijní náklady spojené s běžnými činnostmi prováděnými při vývoji webových stránek. Například, mnoho frameworků poskytuje knihovny pro přístup k databázím, tvorbu šablon a řízení relací. [1]

Návrhový vzor definuje vztahy mezi třídami a objekty bez konkrétní implementace. Framework je vývojové rozhraní, soubor softwarových nástrojů pro vytváření dynamických webových stránek, aplikací a služeb, které jsou specifickou realizací abstraktního návrhového vzoru. Framework však vývojářům neposkytne řešení daného problému. Jedná se o nástroj zjednodušující dosažení cíle. Záleží na konkrétním frameworku, zda bude určovat strukturu řešení a do jaké míry bude programátor schopen ovlivnit tuto strukturu. Díky tomu se programátor plně soustředí na navržení logiky aplikace a nemusí se zdržovat s řešením rutinních problémů. [1]

Webové aplikace obecně, jsou tvořeny pomocí základních technologií pro účely zobrazení obsahu v prohlížeči, jako je HTML, CSS nebo JavaScript. Nicméně i takové základní technologie, které byli původně navrženy pouze pro vytvoření webových stránek a přizpůsobení jejich vzhledu, mají tendenci ke znovu-použitelnosti. Vytváření stejných struktur, pro různé webové stránky, vedlo k zavedení front-end frameworků. Ty slouží pro rychlé vytvoření základních rozvržení stránek, které by bylo možné snadno přizpůsobit. Takové technologie jsou obvykle spojeny s webovým frameworkem a mají ještě více usnadnit vývoj aplikací. [4]

2 Cíl práce a metodika

2.1 Cíle práce

Cílém této práce bude vytvořit informační systém na webu, který bude zprostředkovávat prodej náhradních dílů pro automobily. Hlavním podmětem pro vývoj této webové stránky je nedostatek přehledných webových stránek, které by měly vhodné třídění inzerátů do kategorií, tak aby bylo snadné vyfiltrovat to, co návštěvník vyhledává.

K tvorbě serverové části, portálu se bude využívat programovací jazyk PHP, Framework Nette a pro uchování dat se využije databáze MySQL. Pro vývoj UI se použije HTML 5, CSS, Java Script, knihoven JQuery a front-end framework Bootstrap. Za pomoci uvedených technologií bude vytvořeno přehledné responzivní UI, které bude uživatelsky přívětivé jak na PC, tak tabletech či chytrých telefonech.

2.2 Metodika

V rešeršní části této práce budou analyzovány veškeré technologie související s vývojem webového portálu tak, aby bylo možné je vhodně implementovat v praktické části. V praktické části pak bude provedena analýza funkčních požadavků z hlediska rolí jednotlivých uživatelů portálu. Na základě této analýzy se veškeré funkční požadavky implementují do UX návrhu wireframů, podle kterých se realizuje finální UI. Pro uchování dat byla zvolena relační databáze MySQL. Následujícím krokem bude návrh této databáze s ohledem na normalizaci databázového modelu. Předposlední fází je implementace MVC architektury a naprogramování back-endu za pomoci Nette PHP frameworku. V poslední fázi se otestují veškeré funkční části portálu a odladí se poslední chyby. Na závěr práce bude provedeno zhodnocení, dosažení stanovených cílů a zamýšlení se nad případným rozšířením portálu do budoucna.

3 Teoretická východiska

3.1 Webové PHP frameworky

3.1.1 PHP

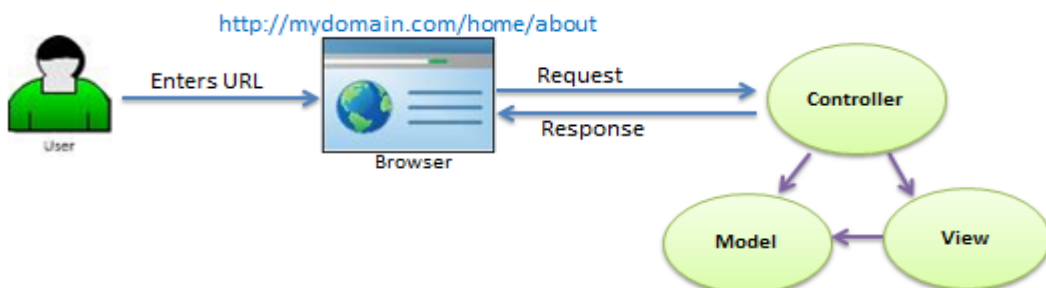
PHP (Hypertext Preprocessor) je původně skriptovací jazyk, který se stal nejpoužívanějším skriptovacím jazykem pro tvorbu webových aplikací. Stal se populárním především díky snadnému použití a velkému množství funkcí. PHP skvěle pracuje s HTML, je s ním snadno integrovatelné, a proto je vhodné k programování webových aplikací i pro méně zkušené vývojáře. Také většina open-source CMS, jako je Wordpress nebo Joomla, jsou napsány v jazyce PHP. Další nespornou výhodou PHP je snadná dostupnost webových hostingů s Apache serverem, z toho důvodu je velice vhodné i pro start-up s nízkým počátečním kapitálem. [3]

3.1.2 Návrhové vzory

Každý Framework, používaný pro tvorbu rozsáhlých webových aplikací, stojí na určitém návrhovém vzoru. Tyto vzory představují řešení pro běžné problémy a implementují nápady pro jejich řešení. Nehledě na části zdrojového kódu nebo další knihovny, ale jako šablona popisující přístup k řešení v různých situacích. Existují různé typy a úrovně návrhových vzorů, nicméně pro PHP je nejtýpější návrhový vzorem MVC. [6]

3.1.3 MVC

Model-View-Controller je model softwarové architektury pro implementaci uživatelských rozhraní. Je tedy oblíbenou volbou pro tvorbu webových aplikací. Obecně odděluje aplikační logiku do tří samostatných částí, což podporuje modularitu a usnadňuje spolupráci a opětovné použití. Díky tomu je aplikace výrazně flexibilnější. [6]



Obrázek 1: Princip Model - View – Controller [2]

3.1.3.1 Model

Model definuje data, která by aplikace měla obsahovat a logiku prováděnou nad těmito daty. Pokud se stav těchto dat změní, pak model obvykle upozorní View (takže se zobrazení může změnit podle potřeby) a někdy i Controller (pokud je pro ovládání aktualizovaného zobrazení potřebná jiná logika). [5]

Poskytuje-li model funkce na čtení a zapisování údajů, neznamená, že model se rovná databázi. “Ve většině MVC architektur ukládá model údaje v relační databázi, jako je například MySQL, PostgreSQL nebo Oracle.” [1] Model však není omezen jen na databáze, může také používat různá úložiště údajů jako jsou textový soubor, RSS nebo jiné webové služby.

3.1.3.2 View

View definuje postup, jak zobrazovat data z aplikace uživateli, jak komunikovat mezi uživatelem a aplikací. Jinými slovy se jedná o prezenční vrstvu aplikace, realizovanou pomocí HTML, CSS, JavaScriptu a šablon. View se z pravidla skládá z více šablon, které se do sebe různě zanořují, aby se minimalizoval duplicitní kód. [5]

3.1.3.3 Controller

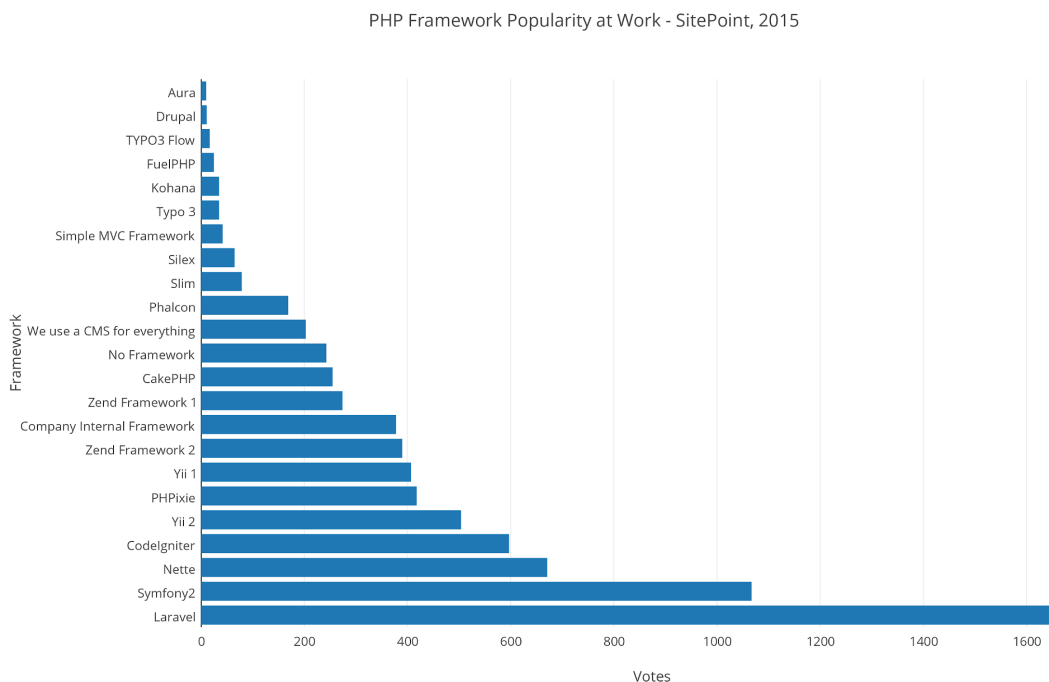
Controller je vlastně řídicím prvkem celého MVC modelu, přijímá a zpracovává veškeré požadavky od uživatelů (GET či POST). Na základě těchto požadavků, předává data modelu, aby nad nimi provedl logické operace. Na základě výsledků zpracování v Modelu, pak controller vybírá vhodný View, který vrací uživateli. [5]

3.1.4 Kritéria pro porovnání PHP frameworku

Pro objektivní porovnání a usnadnění výběru vhodného frameworku k tvorbě malých až středně velkých webových aplikací byla stanovena následující kritéria:

- Křivka učení
- Bezpečnost
- Licence
- Dokumentace a podpora komunity
- Funkce a rozšiřitelnost

Tato kritéria budou hodnocena napříč třemi nejoblíbenějšími frameworky, stanovenými průzkumem na webu www.sitepoint.com. Podle uvedeného průzkumu patří mezi 3 nejoblíbenější frameworky v pořadí od nejoblíbenějšího: Laravel, Symfony2 a Nette.



Obrázek 2: Graf oblíbenosti PHP frameworků [6]

3.1.5 Laravel framework

Laravel je moderní webový framework, založený na Symfony, používaný pro budování stabilních a rozsáhlých webových aplikací v PHP. Vyvinul jej Taylor Otwell v roce 2012 a je tedy považován spíše za mladý framework. Laravel se stal jedním z nejpobulárnějších PHP frameworků díky jeho kořenům v Symfony, Composeru, Blade atd. I když byl používán a podporován jen několik let, získal obdiv mnoha vývojářů PHP, kteří byli zvyklí pracovat s frameworky jako Symfony nebo Cake PHP. Laravel spojuje výhody těchto frameworků s certifikovanými postupy inspirovanými technologiemi jako ASP.NET, Sinatra nebo Ruby on Rails a transformuje je do syntaxe PHP. Taková syntéza vytváří silný nástroj pro rychlý vývoj stabilních a snadno rozšířitelných moderních aplikací. Laravel stojí čistě na architektuře MVC a vyznačuje se především optimalizací pro reálný svět (resp. use case). [7]

Laravelova komunita poskytuje jasné tutoriály, které pomáhají uživatelům dobře poznat framework. Vhodně strukturovaná dokumentace pokrývá instalaci, úvod do balíčkového systému a vysvětlení základů, jako je routování, middle-ware, controllery a View. Po seznámení se se základy používání, mohou uživatelé procházet dokumentací, která popisuje služby, architekturu a databáze, které jsou dobře interpretovány spolu s příklady zdrojového kódu. Podpora pro nové uživatele je velmi rozsáhlá. Laravel také poskytuje webovou stránku laracasts.com, která obsahuje hodiny komentovaných video tutoriálů jak zdarma, tak i předplacených. [7]

Framework využívá nástroje Composer, oblíbeného správce závislostí v PHP, který spravuje rozšíření frameworku. Z toho důvodu musí být aplikace nainstalována na vývojovém serveru, jinak by nemohl být Laravel rozumně provozován. Jak bylo uvedeno výše, Laravel je postaven na základech oblíbených PHP frameworků v kombinaci s certifikovanými postupy inspirovanými z technologií různých jazyků. Jedním z takových postupů je správa závislostí prostřednictvím balíčkového systému. Composer umožňuje uživatelům pracovat v konzoli, způsobem velmi podobným balíčkům v Ruby. Lze spravovat závislost jednoduše tím, že je daný balíček v aplikaci vyžadován. Tyto balíčky mohou obsahovat nástroje pro různé formy, autentizaci, PHP frameworky atd. Opětovné použití kódu je jedním z největších pokroků ve vývoji webových aplikací a Laravel jej umožňuje používat certifikovaným způsob, stejně jako automaticky generovaný kód pro Controller nebo View. [7]

Jedním z hlavních rysů Laravelu je jeho výjimečný mechanismus objektově relačního mapování, který může být také použit jako samostatné ORM, přestože byl původně vyvinut pro Laravel. ORM poskytuje implementaci ActiveRecord pro práci s databází, vytvářející model pro každou tabulku, který s ní umožňuje interagovat. Laravel také obsahuje builder databázového schématu, který je tvořen třídou, s jejíž pomocí umožňuje manipulovat s databázovými tabulkami. Správa tabulek je tedy možná jednoduše prostřednictvím PHP kódu a nevyžaduje žádné nástroje pro správu databáze. Tato schémata jsou typicky spárována s migracemi, které vývojářům umožňují modifikovat schéma databáze, a zůstat ve stavu schématu. [7]

Laravel také umožňuje maximálně přizpůsobit nastavení vývojového prostředí vlastním balíčkem virtuálního stroje Vagrant. Balík se nazývá Homestead a zahrnuje Ubuntu, PHP, MySQL, PostgreSQL a další užitečné nástroje, jako je vlastní manažer úloh Laravelu. Homestead se používá k vytvoření maximálně komfortního vývojového prostředí, které lze připojit přes SSH1. Uživatelé pak nemusí instalovat PHP ani libovolnou databázi, ale po správné instalaci Virtual Box2 a Vagrant mohou nainstalovat Homestead jednoduše pomocí Composeru. Tento přístup může zjednodušit přenositelnost, avšak běh Laravelu na vlastním PHP serveru může přinést problémy s kompatibilitou, které mohou být obtížně řešitelné v systémech Windows a Mac. [7]

Syntaxe v Laravelu je spíše expresivní, na první pohled se zdá být čistá a dobře čitelná. Je také přehlednější a lépe pochopitelná. Rozdíl může být v kombinaci Controller-View, který musí být vytvořen ručně, stejně jako pro ostatní poziční modely. V kombinaci s řízením databáze, routováním a konfigurací může být pro začínající vývojáře docela matoucí. Samotná syntaxe je však dobře zpracována. [7]

3.1.5.1 Hodnocení kritérií:

Křivka učení

Velmi dobré video tutoriály spolu s velmi příjemně strukturovanou dokumentací poskytují perfektní start s Laravelem. Nicméně, jako úvod do logiky webových frameworků a systémů balíčků může být Laravel nejprve trochu matoucí. Po pochopení základních aspektů a metod vývoje, je učení velmi rychlé.

Bezpečnost

I přes to, že framework Laravel je vybaven bezpečnostními nástroji chránícími před útoky XSS a CSRF, je zapotřebí je volat ručně. To nese větší nároky na programátora, který na to musí stále myslet, a pokud zapomene zavolat bezpečnostní funkce, může tím vzniknout bezpečnostní díra.

Licence

Laravel je distribuován pod licencí MIT s jeho zdrojovými kódy hostovaným na GitHubu.

Dokumentace a podpora komunity

Dokumentace v Laravelu je dobře strukturovaná, organizovaná a společně s velmi aktivní komunitou poskytuje perfektní kombinaci uživatelské podpory.

Funkce a rozšiřitelnost

Na velmi vysoké úrovni díky managementu závislostí, zajišťované s pomocí Composeru.

3.1.6 Nette

Nette je open-source framework pro snadný vývoj webových aplikací v PHP. Jejím původním autorem je David Grudl. V současné době ho, ale udržuje organizace Nette Foundation, což je skupina individuálních programátorů, kteří od roku 2009 vyvíjejí a poskytují podporu pro tento framework. Nette je populární především mezi začátečníky, díky jednoduchému procesu učení a užití. Hlavní část podpory a informací je k dispozici v českém jazyce, čímž se Nette stal jedním z nejpobulárnějších frameworků českých vývojářů. Byl napsán tak, aby byl snadno použitelný a dal vývojářům efektivní a srozumitelnou syntaxi, která by jim umožnila rychleji se učit. Realizuje vývoj založený na událostech a je založen hlavně na komponentovém využití. Návrhový vzor Nette byl použit Model-View-Presenter je velmi podobný modelu MVC. Model představuje vrstvu pracující s daty, která je úplně oddělena od zbývající části aplikace a komunikuje pouze s Presenterem, který předávává změny v datech do View. Framework Nette je znám také svým zaměřením na eliminaci bezpečnostních rizik a pokročilé zabezpečení aplikací. [8]

Dokumentace Nette poskytuje pěkný a jasný úvod k frameworku pomocí strukturovaného tutoriálu pro vývoj jednoduchého blogu. Každý krok procesu seznamování s Nette je popsán spolu s příkladem zdrojového kódu. Instalační balíček obsahuje nástroj pro kontrolu minimálních konfiguračních požadavků, který pomáhá uživateli ověřit správnou konfiguraci prostředí pro instalaci. Na rozdíl od jiných moderních frameworků je vývojový proces v Nette navržen tak, aby minimalizoval konfiguraci a zaměřil se na důležité části aplikační logiky.

Nette poskytuje vlastní databázový nástroj zvaný Adminer. Umožňuje vytvářet databáze a pracovat s tabulkami stejně jako každý jiný administrátorský nástroj pro databáze, ale také umožňuje automaticky vytvářet cizí klíče mezi tabulkami, za pomoci názvu tabulky, použité jako datový typ argumentu v druhé tabulce, což vývojářům šetří čas. [8]

Jednou z hlavních výhod Nette jsou technologie eliminující bezpečnostní díry. Automaticky chrání aplikaci před útoky jako jsou cross-site scripting za pomoci technologie context-aware escaping. Nette také chrání formuláře před cross-site request forgery, které umožní útočníkovi procházet stránky, zatímco je uživatel přihlášen. Mnoho jiných forem útoků, jako jsou útoky na URL nebo neoprávněná manipulace se session jsou automaticky blokovány.

Nette nabízí dva režimy spuštění aplikace na serveru. Vývojový režim, který programátorovi poskytuje co nejvíce informací o rychlosti aplikace, routování a zobrazení chyb. Naproti tomu produkční režim ukládá podrobnosti chyb do log souborů, tak aby byly skryté pro uživatele, ale zpětně dohledatelné programátorem. [8]

Další užitečné nástroje Nette, které mohou být užívány také jako samostatné komponenty, jsou šablonovací systém Latte a ladicí nástroj tzv. "laděnka". Latte je šablonový engine, který přetváří těžko čitelnou syntaxi PHP a HTML na syntaxi dobře čitelnou. Pomocí vlastních maker umožňuje psát cykly, podmínky nebo různé vlastní příkazy způsobem podobným HTML, přesto však poskytuje zmíněnou bezpečnost a automatickou ochranu před zmiňovanými útoky. Nette debugger a Tracy, umožňují rychle detekovat a zaznamenat chyby v kódu v uživatelsky přívětivém rozhraní. Nette velice dobře umožňuje používat AJAX volání s minimem konfigurace. Programátor si definuje tzv. "Snippets", které představují části webu, které mají být obnoveny po provedení definované akce. Tímto způsobem umožňuje vytvářet dynamickou strukturu obsahu přímo připojenou k databázi, aby její obsah mohl být dynamicky měněn na straně klienta bez obnovení celé stránky. [8]

Struktura souborů Nette kopíruje architekturu MVC. Kořenový adresář obsahuje složku aplikace, která obsahuje: Presenter, View, Model a vlastní komponenty. Zbytek struktury souborů se skládá z front-endu resp. "www" složky s JavaScripty, CSS a různými doplňky. Dále je tvořeno a složkou "vendor" obsahující soubory frameworku a soubory pluginů. Uživatelé však mohou spravovat systémy aplikačních složek podle svých potřeb, pokud dodrží správné používání Namespace. Tato zjednodušená struktura souborů spolu s jednoduchým konfiguračním souborem a routováním vytváří dobře strukturovaný základ pro tvorbu webových aplikací pro každého vývojáře. [8]

I když Nette nemá takovou podporu jako Laravel, je nyní distribuována jako set nezávislých, znovupoužitelných komponent, které lze použít v původním balíčku. Uživatelé mohou definovat rozšíření, které chtějí zahrnout do Nette, nebo jednoduše používat samostatné komponenty, jako je například Tracy. Stejně jako Laravel, i Nette může používat Composer pro správu závislostí a rozšíření. [8]

3.1.6.1 Hodnocení kritérií:

Křivka učení

Nette je velmi snadné k učení díky dobře organizované struktuře souborů, snadné konfiguraci, úvodním tutoriálům a dobře strukturovaným průvodcům na webových stránkách frameworku i na kanálu Youtube.

Bezpečnost

Předností frameworku Nette je, že všechny výstupy se ošetřují automaticky proti XSS, takže riziko, že programátor na něco zapomene je minimální.

Licence

Nette Framework je šířený jako svobodný software pro kohokoliv k užití. Umožňuje volbu mezi New BSD a GNU General Public License.

Dokumentace a podpora komunity

Nette má aktivní komunitu, která poskytuje podporu na fórech, které jsou většinou v češtině. Jeho webové stránky obsahují pěkně strukturované průvodce pro různé části frameworku. Programátorská příručka je dobře organizovaná a vizuálně dobře zpracovaná.

Funkce a rozšiřitelnost

Nette má na svých stránkách rozsáhlou databázi doplňků, pluginů a komponent. Může být dokonce kombinováno se Zend frameworkem a spravováno přes Composer stejně jako Laravel.

3.1.7 Symfony

Symfony se původně jmenovalo Sensio Framework, protože bylo vyvíjeno pro účely společnosti Sensio Labs. Původním vývojářem Symfony byl Fabien Potencier, který začal framework vyvíjet v roce 2003 s cílem vytvořit open-source PHP framework s MVC architekturou. Šablonovací struktura je velice podobná té z Ruby on Rails. První verze byla publikována v roce 2005 a poměrně brzy získala velkou oblibu mezi vývojáři. Symfony je také vydáváno pod svobodnou licenci a to MIT, která podobně jako BSD umožňuje i komerční využití pod podmínkou uvedení kopie licence v kódu. Jako jeden z prvních frameworků Symfony umožňovalo použít Dependency Injection container. [9]

V průběhu let získal vývoj v Symfony velkou popularitu. To mělo obrovský dopad na PHP frameworky a nástroje, jako jsou Doctrine, composer nebo PSR, které se staly standardem mnoho dalších PHP nástrojů. Má velmi dobře rozšiřitelnou architekturu a flexibilní uživatelské rozhraní. V době svého vzniku se jednalo o velmi revoluční Framework díky integraci již zmíněných nástrojů, které byly vhodné pro vývoj webových aplikací na velmi vysoké úrovni. V současné době se pořád jedná o velmi robustní framework, který nabízí dlouhodobou podporu pro verze označované LTS (Long Term Support version). Poslední LTS verze 3.4 byla vydána v září 2017 a Symfony garantuje její podporu do listopadu 2020. I přes zmíněné plusy se již nejedná o framework disponující revolučními řešeními, jako spíše o framework s velkou rozšířeností a dlouhodobou stabilitou. [9]

Stejně jako ostatní frameworky i Symfony drží dobře organizovaný kód, který předchází duplicitám v kódu a vede k rychlému vývoji webových aplikací. Za pomoci systému tzv. balíčků Symfony zajišťuje znovupoužitelnost kódu, který není nijak závislý na zbytku zdrojového kódu. V rámci balíčků jsou ve frameworku integrovány různé open-source nástroje jako například Doctrine, Assetic, Propel a mnoho dalších. I architektura MVC lze snadno implementovat s využitím balíčků. [9]

V rámci frameworku Symfony je prezentační vrstva řešena pomocí šablony TWIG. Tento speciální meta jazyk je stručnější než čisté PHP, výrazy a syntaxe je více čitelná a intuitivní pro lidi, rozšířená o zkratky pro typické vzory užití. TWIG byl vyvinut s PHP5, proto je plně flexibilní a rozšiřitelný o uživatelsky definované značky a filtry. Také je velmi rychlý, protože šablony jsou kompilovány až na čistý kód PHP při každém požadavku na vykreslení

View. TWIG byl původně vytvořen autory Symfony2, obě knihovny jsou tedy vzájemně plně kompatibilní. [9]

Symfony disponuje propracovaným systémem routování, které zajišťuje zpracování HTTP požadavků správným controllerem, přičemž je zachována srozumitelná URL pro člověka a zároveň optimalizovaná pro SEO. [9]

Vývojové prostředí v Symfony disponuje různými režimy, nicméně nejčastěji jsou využívány dva. Režimy pro vývoj, kdy se ve spodní části obrazovky zobrazuje ladící lišta, podobně jako v Nette, která poskytuje programátorovi řadu užitečných informací, např. parametry dotazu na server, výskyt vyjímky, hodnoty aktuálně používaných proměnných, informace o vykreslených formulářích atd. Pokud je vše připraveno, pak se používá režim produkční, který skrývá veškeré ladící nástroje a tím je i rychlejší než vývojový režim. [9]

Pro umožnění snadné práce s doménovou logikou je v rámci základní instalace frameworku integrován balíček ORM Doctrine, který mapuje tabulky databáze na objekty na základě tzv. metatříd. Vlastnosti entity Doctrine umožňuje definovat ve více formátech. [9]

Konfigurace Symfony je řešena pomocí YAML, což je akronym pro Ain't Markup Language. Jedná se vlastně o jazyk čitelný pro počítač i člověka, který je strojově spracovatelný, vhodný pro serializaci dat a schopný interakce se skriptovacími jazyky. YAML je nejvíce podobný jazyku XML, jen s výrazně jednodušší syntaxí, díky čemuž je jeho zpracování mnohem rychlejší. Právě díky zmíněným výhodám Symfony stále preferuje YAML před XML, nicméně i přes to Symfony podporuje použití XML. [9]

Dalším charakteristickým rysem Symfony jsou rozsáhlé možnosti konfigurace a absolutní oddělování jakéhokoliv nastavení od samotného kódu do samostatného souboru YAML. Může se stát i to, že vývojář tráví spoustu času pouze konfigurací a ani se k programování nedostane, příkladem konfigurace může být třeba logování, cachování, prostředí databáze, zabezpečení, routování, konfigurace View a mnoho dalších nastavení. Konfigurační soubory lze doplnit i o samotné PHP skripty, pokud je to nutné pro ucelenou konfiguraci. [9]

Mezi jednu z klíčových vlastností frameworku řadí vývojáři možnost správy aplikace za pomoci příkazové řádky. Zavolání několika jednoduchých příkazů si vývojář může velmi rychle vytvořit celou strukturu budoucí aplikace a automaticky vygenerovat mnoho zdrojového kódu. [9]

3.1.7.1 Hodnocení kritérií:

Křivka učení

Symfony poskytuje dobrou základnu rozsáhlé dokumentace a dlouholeté podpory od komunity, nicméně svým rozsahem a komplexností bude křivka učení delší než u framework Nette a Laravel.

Bezpečnost

Za pomoci YAML souborů, framework chrání aplikaci před XSS útoky, nicméně další způsoby ochrany závisí již na vývojáři a neposkytuje tedy automatickou ochranu všech vstupů, či výstupů jako Nette.

Licence

Symfony Framework je šířený jako svobodný software pod svobodnou licenci, a to konkrétně MIT, která podobně jako BSD umožňuje i komerční využití pod podmínkou uvedení kopie licence v kódu.

Dokumentace a podpora komunity

Symfony má již za sebou dlouhou historii a je tady pevně zakořeněno na mezinárodním poli PHP framework, takže snadno můžeme najít řadu tutoriálu, YouTube návodů či možnosti placeného, certifikovaného kurzu. Samotná dokumentace je velice podrobně vedená a je možné ji stáhnout jako samostatnou knihu.

Funkce a rozšiřitelnost

Stejně jako ostatní frameworky, Symfony nabízí možnost instalace balíčků za pomoci composeru z příkazové řádky, ovšem oproti Laravel nabízí necelou jednu třetí dostupných rozšíření.

3.1.8 Porovnání PHP frameworků

3.1.8.1 Laravel

Laravel je velmi výkonný nástroj pro tvorbu webových aplikací v PHP. Poskytuje přístupy k rozsáhlým rozšiřujícím balíčkům, snadné nastavení databáze pocházející z populárních frameworků, jako je Ruby on Rails, které implementuje v PHP, čímž snižuje potřebu učit se složitější programovací jazyky a ponechává vývojářům funkce, které z PHP znají. Laravel je založen na architektuře MVC a vyznačuje se především optimalizací pro reálný svět a skutečností, že vývojáře nezatěžuje rutinními kroky, ale nechává je se soustředit na vývoj logiky. Framework je šířený jako svobodný software a může být použit k vývoji rozsáhlých systémů, stejně jako blogů a CMS.

3.1.8.2 Nette

Nette je především nejlepší volbou pro začínající vývojáře webových aplikací. Většinou díky čistému přístupu MVC, jednoduché struktuře souborů a dobře známým typům tříd. Je navržen tak, aby byl co nejjednodušší a nevyžaduje žádné komplikované konfigurace. Kromě toho, že Nette je založen na PHP, používá vlastní šablonovací systém Latte, který snižuje množství HTML, které je třeba zapsat. Jedná se o velmi bezpečný framework s dobře chráněnými formuláři a mnoha vlastními bezpečnostními pluginy. Nette je vhodné pro vývoj menších webů, pro e-commerce, wiki, blogy nebo aplikace CRUD, ale není vhodná pro komplexní rozsáhlé aplikace.

3.1.8.3 Symfony

Symfony je mnoha ohledch přesným opakem Nette, a je tedy vhodné převážně pro zkušenější vývojáře, kteří ocení velké množství možných nastavení za pomoci YAML a možností instalace rozšiřujících balíčků, na kterých pak budou stavět rozsáhlé webové aplikace. Stejně jako výše zmíněné frameworky, tak i Symfony vychází z MVC architektury. Jednou z hlavních předností oproti Nette a Laravel je pravidelnost nově vydávaných verzí, garance opravy chyb a dlouhodobě garantovaná podpora verzí LTS, kde následující 3 roky garantují opravy chyb a po 4 roky opravy bezpečnostních chyb. Podpora komunity je také na vysoké úrovni, stejně tak jako dokumentace, kterou je možné stáhnout ve formátu knihy.

3.1.9 Tabulka porovnání PHP frameworků

Pro závěrečné hodnocení zvolených frameworků jsem vytvořil 2 tabulky, kde ke každému kritériu všem Frameworků přiřadím kvalitativní znaky (fakta) jako například: rok vzniku a licence, jinými slovy jsou to vlastnosti, které nelze jednoznačně kvantifikovat. Ve druhé tabulce přiřadím hodnocení na škále 1-10, přičemž 1 bod znamená nejhorší a 10 bodů znamená nejlepší. Jednotlivé hodnocení přiřazuji na základě analýzy všech tří frameworků.

Vlastnosti	Laravel	Nette	Symfony
Rok vzniku	2012	2013	2005
Licence	MIT	New BSD,GNU GPL	MIT
Architektura	MVC	MVP	MVC
Výsledků na YouTube ¹⁾	~ 39 000	~ 1 000	~ 10 000
Výsledků na Google ¹⁾	~ 3 700 000	~ 300 000	~ 3 600 000

Tabulka 1: Porovnání kvalitativních hodnot PHP frameworků

1) výsledků po zadání dvojice klíčových slov „název frameworku“ + Framework např. Nette Framework

Vlastnosti	Laravel	Nette	Symfony
Křivka učení	7	9	5
Podpora verzí	8	5	10
Bezpečnost	8	10	7
Dokumentace	9	5	8
Rozšiřitelnost	10	6	7
Průměr	8,4	7	7,4

Tabulka 2: Porovnání kvantitativních hodnot PHP frameworků

3.2 Front-end frameworky

Ve většině moderních webových aplikací je celkem náročné používat pouze základní internetové technologie jako, například: HTML5, CSS3 a JavaScript, které vytvářejí plně dynamické GUI. Proto programátoři dnes používají komplexní knihovny, které veškeré tyto technologie spojují v praktickém frameworku, který urychluje tvorbu front-endu, zajišťuje kompatibilitu napříč prohlížeči a redukuje duplicitní kód. [11]

3.2.1 HTML5

Hlavním účelem každého webového prohlížeče je zobrazit stránku na straně klienta pomocí HTML, který má přísný kódovací standard, aby se zabránilo chybám v zobrazení v různých prohlížečích. Se zavedením HTML 5 dostali vývojáři větší svobodu při dodržování standardů W3C, což umožnilo snadnější vytvoření front-endu. [12]

HTML5 měl poměrně vrtkavou historii vývoje a není tedy jen přeformulováním předchozích verzí jazyka, ale obsahuje všechny platné prvky z HTML4 i XHTML 1.0. Navíc je navržen s ohledem na některé principy, aby zajistil, že funguje na téměř každé platformě, je kompatibilní se staršími prohlížeči a elegantně zpracovává chyby. Za zmínku stojí především nové tagy: article, section, header, footer pro strukturování webu do logických částí nebo technologie pro přehrávání audia/video, kreslení pomocí canvas, drag-and-drop funkcionalitu a mnoho dalších technologií. [12]

3.2.2 CSS

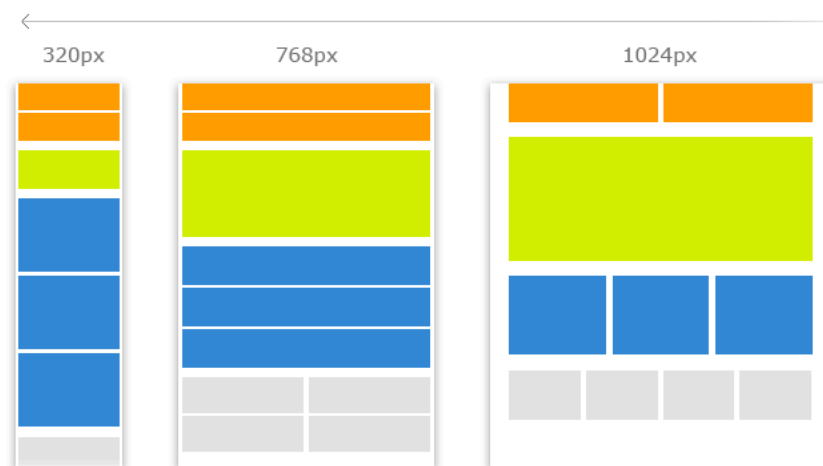
Kaskádové styly často přinášejí stejný problém přísných standardů, které mohou mít vliv na vizuální část webových stránek, pokud je vývojáři nebudou dodržovat. Hlavním problémem s CSS je však trvalé, opakované použití kódu a absence znovu-použitelnosti kódu, což často vede k dlouhým, opakovaným a nejasným řádkům kódu. [12]

Řešením těchto problémů bylo zavedení jazyka SASS, který rozšiřuje funkčnost CSS3 o několik mechanismů, které jsou k dispozici ve více tradičních programovacích jazycích, zejména v objektivně orientovaných jazycích, ale které nejsou k dispozici samotnému CSS3. SASS představil deklaraci proměnných, správné vkládání stylů, argumenty, dědičnost selektory atd., což vývojáři poskytuje výkonný nástroj pro přizpůsobení vzhledu webových stránek způsobem, který je podobný objektově orientovanému programování. [13]

V roce 2009 byl představen dynamický stylovací jazyk nazvaný LESS. Rozšířil SASS o několik dalších funkcí a stal se nejpoužívanějším nástrojem pro dynamickou tvorbu kaskádových stylů. [13]

3.2.3 Responzivní design

Responzivní design webových stránek je přístup k tvorbě webu, který renderuje rozložení webové stránky vhodně podle velikosti zařízení. Designéři responzivního webu využívají adaptivní obrázky, fluidní mřížky a tzv. media queries, aby přizpůsobily uspořádání prvků podle velikosti obrazovky zařízení. Jsou schopni využívat detekce prvků, a to především prostřednictvím webového klienta, k vytvoření vhodného uspořádání prvků podle konkrétního zařízení. V praxi se ovšem využívá spíše hotových komplexních řešení, jako jsou různé knihovny nebo frameworky, které umožňují rychlý vývoj a minimalizují duplicitní kód. [13]



Obrázek 3: Princip responzivního designu webu [14]

3.2.4 DOM

Objektový model dokumentu (DOM - The Document Object Model) je programovací rozhraní pro dokumenty HTML a XML. Představuje stránku tak, aby program mohl změnit strukturu dokumentu, styl a obsah. DOM představuje dokument jako uzly a objekty. Tímto způsobem mohou se stránkou efektivně pracovat programovací jazyky. [16]

Webová stránka je dokument. Tento dokument lze buď zobrazit v okně prohlížeče, nebo jako zdrojový kód HTML. V obou případech je to však tentýž dokument. Objektový model dokumentu (DOM) představuje stejný dokument tak, aby s ním mohlo být manipulováno. DOM je objektově orientovaná reprezentace webové stránky, kterou lze upravit skriptovacím jazykem, jako je například JavaScript. [16]

Normy W3C DOM a WHATWG DOM jsou implementovány ve většině moderních prohlížečů. Mnoho prohlížečů rozšiřuje standard, proto je třeba věnovat zvýšenou pozornost tomu, když je DOM používán na webu, kde mohou být dokumenty přístupné různými prohlížeči s různými DOMy. [16]

3.2.5 JavaScript

JavaScript byl zpočátku vytvořen pro "oživení webových stránek". Programy v tomto jazyce se nazývají skripty. Mohou být napsány přímo v HTML, a mohou být automaticky spouštěny při načítání stránky. Skripty jsou psány a prováděny jako prostý text. Nepotřebují speciální přípravu nebo kompilaci k běhu. V tomto ohledu je JavaScript velmi odlišný od jiného jazyka nazvaného Java. [16]

Knihovnou nad JavaScriptem je jQuery, které bylo původně vydáno v lednu 2006 na BarCamp NYC Johnem Resigem, který byl ovlivněn dřívější knihovnou CSSQuery Deana Edwardse. jQuery je meziplatforma JavaScriptu navržená pro zjednodušení skriptování HTML na straně klienta. Jedná se o bezplatný software s otevřeným zdrojovým kódem s otevřenou licencí MIT. Syntaxe jQuery je navržena tak, aby usnadnila navigaci dokumentu, výběr prvků DOM, vytváření animací, zpracování událostí a vývoj aplikací Ajax. [16]

3.2.6 Ajax

AJAX je asynchronní jazyk kombinující JavaScript + XML, ačkoli není technologií sám o sobě, je termín vytvořený v roce 2005 Jesse Jamesem Garrettem. Ten popisuje "nový" přístup k používání několika současných technologií, včetně HTML nebo XHTML, kaskádových stylů, JavaScript, Objektový model dokumentu, XML, XSLT a především objekt XMLHttpRequest. Když zkombinujeme všechny tyto technologie v Ajax model, získáme webové aplikace, které dokáží provádět rychlé, přírůstkové aktualizace uživatelského rozhraní bez opětovného načtení celé stránky prohlížeče. Díky tomu je aplikace rychlejší a lépe reaguje na akce uživatele. [17]

Ačkoli X ve zkratce Ajaxu označuje XML, JSON se dnes používá více než XML, protože má mnoho výhod, jako je menší objem dat pro přenos a je součástí JavaScriptu. Pro strukturování informací v modelu Ajax se používají jak JSON, tak XML. [17]

3.2.7 Kritéria pro porovnání front-end frameworků

Pro objektivní porovnání a usnadnění výběru vhodného frameworku k tvorbě malých až středně velkých webových aplikací byla stanovena následující kritéria:

- CSS preprocessor
- JavaScript
- Stabilita
- Grid
- Formuláře
- Podpora komunity
- Design

Tato kritéria budou hodnocena napříč třemi nejoblíbenějšími frameworky, stanovenými průzkumem na webu www.sitepoint.com. Podle uvedeného průzkumu patří mezi tři nejoblíbenější frameworky v pořadí od nejoblíbenějšího: Bootstrap, Foundation a SemanticUI.

3.2.8 Bootstrap

Bootstrap je bezplatná sada nástrojů pro vytváření webových stránek a webových aplikací. Obsahuje návrhové šablony založené na HTML a CSS pro typografii, formuláře, tlačítka, navigační a další rozhraní, stejně tak jako volitelná rozšíření jazyka JavaScript. Od zahájení projektu GitHub v roce 2010 se stal nejvýznamnějším vývojovým projektem společnosti GitHub a využívají ho instituce jako NASA nebo MSNBC. [19]

Hlavní myšlenkou nástroje Bootstrap je osvobodit webové vývojáře od kódování samotné struktury webu a poskytnout jim soubor funkcí, které jim umožní vytvářet základní webové stránky pomocí identifikátorů daného obsahu a vytvářet tak moderní responzivní web během několika minut. Vzhledem k tomu, že systém Bootstrap je otevřený a aktivní na GitHubu, sleduje moderní trendy ve webdesignu, které jsou následně implementovány do jeho struktury. [19]

Nejvíce náročný postup při vývoji webové stránky je dokončení zobrazení pro různá zařízení, rozlišení a webové prohlížeče. Vzhledem k tomu, že Bootstrap zavádí sekce a DIVy, které jsou pro takové podmínky předem navrženy, je vývojář vyloučen z řešení tohoto problému jednoduchým sledováním strukturované dokumentace, která již byla přeložena do několika jazyků. [19]

Bootstrap je modulární a sestává v podstatě z rozsáhlé série stylů LESS, které implementují různé komponenty sad nástrojů. Opakovaně použitelné komponenty umožňují vývojářům vyhnout se stylování každé samostatné sekce, a tím zjednodušit celý přístup při navrhování různých sekcí. [19]

Díky technologii Bootstrap a různým technologiím pro zjednodušení kódu se může front-end vývojář soustředit na dotažení rozvržení stránky a vzhledu webové stránky podle daného grafického návrhu. [19]

3.2.8.1 Hodnocení kritérií

CSS preprocessor

Bootstrap 4 je založen na SASS preprocesoru. Má velmi pěknou sadu mixins, samostatný list nastavení, opakovaně použitelné komponenty apod.

JavaScript

Bootstrap je napsány na standardu ES6, což znamená, že je potřeba Babel (nebo jakýkoli jiný transpilátor), pokud chceme individuálně zahrnout každý z JS nástrojů do vývojové fáze. Bootstrap o kousek vede nad ostatními frameworky, pokud jde o JavaScript, protože jeho komponenty jsou jednodušší, méně komplexní a dobře testované. Bootstrap umožňuje vybrat, jaké komponenty JS chceme, pokud se rozhodneme nepoužívat Babel.

Stabilita

I když Bootstrap 4 nebyl dosud veřejně vydán, jeho tým bere vážně stabilitu a zpětnou kompatibilitu. Používají automatizované testovací nástroje společnosti Saucelabs, aby se ujistil, že vše funguje správně ještě dřív, než provedou commit do hlavní větve. Díky tomu se jedná o velmi stabilní soubor nástrojů, který i přesto, že je v alfa verzi, může být do značné míry používán jako produkční.

Grid

Bootstrap má podporu pro responzivní mřížku a flexibilní breakpoints. Pokud jde o zpětnou kompatibilitu Bootstrap používá stejnou konvenci třídy jako jejich předchozí verze. Tím je zajištěna co nejméně náročná migrace.

Formuláře

Bootstrap 4 má jedny z nejhezčích formulářů. Existují speciální třídy pro přepínače a zaškrtačací políčka, které jim umožní vypadat exkluzivně ve srovnání s těmi ve standardním prohlížeči. Inline formuláře, validační ikony, štítky s ikonami dělají z Bootstrapu velmi silný nástroj.

Podpora komunity

Bootstrap je rozhodně vede, pokud jde o množství tvůrců pluginů a podporu komunity. Bootstrap je používán téměř všude a téměř pro všechno má dostupné pluginy. Stal se výchozím CSS systémem pro web.

3.2.9 Foundation

Původně bylo vydáno jako otevřený front-end framework v září 2011, Foundation začal jako interní příručka pro stylování postavená v projektové agentuře ZURB, jejíž klienty jsou Facebook, eBay a NYSE. Foundation 6 je nejnovější stabilní verze, která byla vydána v listopadu roku 2015. [20]

Foundation je mezi všemi responzivními front-end frameworky jediný framework, který je profesionálně podporovaný organizací (ZURB), která také používá framework každodenně pro své klienty, od malých startupů až po velké podniky. Foundation bylo zkoušeno a testováno již léta. Prošlo pěti cykly majoritních releasů a mnoha dílčími releasy, aby bylo zajištěno, že funguje všude a bude připravené k rozšiřování. [20]

Framework zahrnuje nejběžnější návrhové vzory potřebné pro rychlý prototyp responzivního webu. Díky využití Sass mixins jsou veškeré komponenty snadno designovatelné a jednoduše rozšiřitelné. [20]

Jádrem Foundation je HTML, CSS a Javascript. Nicméně také používá ověřené technologie jako jsou knihovny jQuery, HTML5 Boilerplate a Normalizr. Všechny komponenty a pluginy jsou navrženy, tak aby fungovaly v podporovaných prohlížečích. [20]

Vzhledem k tomu, že Foundation využívá pouze front-endové technologie, nemá problémy s kompatibilitou na libovolných back-endových či serverových technologiích. O kompatibilitě vypovídá fakt, že Foundation je například využíván na platformách Wordpress, Drupal či .Net. [20]

Od verze 4.0 se framework zaměřuje na přístup mobile-first, což je vlastně dvou-úrovňová technologie a filozofie. Foundation je postaveno tak, že se nejprve navrhne webová stránka nebo aplikace pro malé zařízení a pak se postupně rozšiřuje, aby byly zahrnuty obrázky, komponenty, menu, funkce pro větší displej a další plnohodnotné zařízení. I když není nutné jít touto cestou, ve Foundation je to poměrně jednoduché a představuje skutečně velký přínos při tvorbě webu. Tento přístup je v poslední době považován za nejefektivnější. [20]

Foundation je šířený jako open-source software a je k dispozici na GitHubu. [20]

3.2.9.1 Hodnocení kritérií

CSS preprocessor

Foundation je založen na SASS preprocessoru a stejně jako Bootstrap má velmi pěknou sadu mixins, samostatný list nastavení, opakovaně použitelné komponenty apod.

JavaScript

Foundation bylo přepracována na standard ES6 zhruba před rokem, což je dostatek času na to, aby odladily veškeré problémy a chyby, díky reportování chyb komunitou.

Stabilita

Foundation bylo oproti Bootstrapu vydáno jako "stabilní" před více než rokem. Při prvním spuštění byly některé součásti nedokončeny, jako například Abide (jejich knihovna pro validaci formuláře), která byla zcela odebrána. Od verze 6 do verze 6.3 tým změnil jen poměrně málo věcí, čímž se staly některé nastavení zastaralé. Od verze 6.2 zavedly také jako Bootstrap standard ES6. Offcanvas byl vylepšen a nastavení je nyní velmi flexibilní od verze 6.3. Vzhledem k tomu, že Foundation 6 bylo po delší dobu vydáno k volnému užívání, aby bylo otestováno na více produkčních prostředích, je nyní stabilita na vynikající úrovni. Komponenty se zdají být dobře otestovány a tým vydává aktualizace téměř každé 3-4 měsíce.

Grid

Foundation má lepší systém mřížky, pokud jde o čistotu kódu. Skutečnost, že Foundation nemá žádné kontejnery a spoléhá zásadně na rows, je velkým přínosem pro čistší DOM.

Formuláře

Foundation 6 má jednodušší formulářové uspořádání, která se hlavně opírá o mřížku. Inline formuláře ve Foundation neexistují. Také před-stylovaná vstupní pole nejsou součástí Foundation, takže je nutné se spoléhat na externí CSS, pro úpravu vybraného inputu.

Podpora komunity

Foundation nemá takový náskok v množství hotových pluginů jako Bootstrap, nicméně komunita se stále rozrůstá a je otázkou času, zda dožene náskok ostatních frameworků.

3.2.10 Semantic UI

Semantic UI je framework, který pomáhá vytvářet ohromující designy a navrhovat responzivní rozložení za pomoci velmi přátelského HTML. Jádro frameworku je postaveno na technologiích LESS a jQuery. [21]

Podle tvůrců Semantic UI je cílem frameworku usnadnění práce návrhářů a vývojářů "vytvořením jazyka pro sdílení uživatelského rozhraní". Toho dosahují tím, že využívají sémantický, popisný jazyk pro své třídy a konvence názvů. Namísto použití zkratk, stejně jako jiné frameworky, využívá reálná slova blíže k obyčejné angličtině. [21]

Oproti ostatním frameworkům je Semantic UI výjimečný ve 2 směrech, za prvé je to strukturování frameworku a za druhé to je využívání pěti popisných kategorií k pro definování znovupoužitelných UI komponent. [21]

Téměř každá komponenta má typy, stavy a variace. Některé typy tlačítek například obsahují: standardní tlačítko, tlačítko s ikonou, animované tlačítko, krom toho může tlačítko být aktivní, zakázané nebo načítané. Nakonec se tlačítko může lišit ve velikosti a barvě a může být formátováno jako základní, pro sociální sítě, fluidní, přepínací a další. Tato schémata dávají velkou flexibilitu ve vzhledu komponenty. [21]

Druhá unikátní věc v Semantic UI spočívá v tom, že poskytuje některé exkluzivní funkce a komponenty, které se v jiných frameworkcích nevyskytují. Například feed a komentáře v komponentách view nebo v postranním panelu. Také při interakci s komponentami dostáváme výstup pro ladění v reálném čase. Stačí otevřít webovou konzoli a vidíme komunikaci komponent co přesně jaké prvky aktuálně dělají. [21]

Další silná stránka spočívá v tom, že používá minimální a neutrální styl a nechává přizpůsobení otevřené úpravám. Obsahuje důležité a užitečné věci a současně vynechává další funkce, které pravděpodobně nikdy nepoužijeme. Navíc komponenty frameworku jsou přenosné a samostatné, takže je možné je přenést a používat pouze ty, které potřebujete. [21]

Framework využívá pro své prvky jednotky em a rem, což přináší plnou responzitivu komponent. Je potřeba pouze změnit základní písmo a všechny ostatní prvky se odpovídajícím způsobem upraví. Učící křivka s frameworkem je velice krátká díky velmi dobře zpracované dokumentaci, která poskytuje mnoho příkladů pro různé komponenty. Kromě toho má průvodce stylování s technikami a pokyny pro psaní kódu. [21]

3.2.10.1 Hodnocení kritérií

CSS preprocessor

Semantic UI je narozdíl od Bootstrap a Foundation založeno na LESS preprocessoru.

JavaScript

JavaScript je nezbytnou schopností pro vytváření webového rozhraní s využitím Semantic UI. (To může být obtížné pro méně zkušené vývojáře, všichni vývojáři na středních a vyšších úrovních by měli JS dobře znát). Téměř všechny funkce sémantického uživatelského rozhraní nebudou fungovat bez skriptů JS: modální okno, rozbalovací nabídka, posuvníky apod.

Stabilita

Design komponent je opravdu úžasný, ale Semantic UI však ztrácí za Bootstrapem v kompatibilitě s prohlížeči. Bootstrap je kompatibilní s IE 7+, sémantický UI je kompatibilní až s IE 9+. Nicméně pokud cílové publikum využívá pouze moderní prohlížeče, pak není důvod Semantic UI nevyužít.

Grid

Framework je samozřejmě jako ostatní v základu responzivní, nicméně mřížku oproti třeba Bootstrapu používá o 16 sloupcích.

Formuláře

Formuláře v Semantic UI nabízejí širokou škálu prvků a jejich přizpůsobení za pomoci přehledného a čitelného HTML kódu využívajícího sémantické označení tříd.

Podpora komunity

Semantic UI je mezi front-end frameworky poměrně nováček, takže nelze očekávat tak silnou podporu komunity a hotových řešení nebo doplňků. Nicméně dokumentace a samostatný web frameworku jsou velice dobře a přehledně zpracovány, na webu lze nalézt řadou příkladů včetně zdrojových kódů.

3.2.11 Porovnání front-end frameworků

3.2.11.1 Bootstrap

Silnou stránkou Bootstrapu je jeho obrovská popularita. Technicky není nutně lepší než ostatní frameworky, ale nabízí mnohem více zdrojů (články a tutoriály, pluginy a rozšíření třetích stran, šablony atd.) než ostatní front-end frameworky dohromady.

Stručně řečeno, Bootstrap je všude. A to je hlavní důvod, proč si jej lidé nadále vybírají. Jistou nevýhodou může být ztráta jisté unikátnosti a kreativity výsledného designu webu, a to protože většina responzivních webů je postavena na stejných základech.

3.2.11.2 Foundation

Foundation je druhým velkým hráčem mezi front-end frameworky. Se solidní společností, jako je ZURB, která podporuje tento framework, má opravdu pevné základy. Foundation se používá na mnoha velkých webových stránkách jako jsou Facebook, Mozilla, Ebay, Yahoo! a National Geographic.

Foundation je skutečně profesionální front-end framework s firemní podporou, školením a poradenstvím. Poskytuje také mnoho zdrojů, které jsou nápomocné učit se a používat framework rychleji a jednodušeji.

3.2.11.3 Semantic UI

Semantic UI vydává mnoho úsilí, aby se tvorba webových stránek stala mnohem více sémantická. Využívá principy přirozeného jazyka, což činí kód mnohem srozumitelnější a čitelnějším.

Ze všech uvedených frameworků se jedná o nejinovativnější a plnohodnotnější front-end framework. Celková struktura frameworku a konvence pojmenování, pokud jde o jasnou logiku a sémantiku jejích tříd, také překonává ostatní frameworky.

3.2.12 Tabulka porovnání front-end frameworků

Pro závěrečné hodnocení zvolených frameworků jsem vytvořil tabulku, kde ke každému kritériu všech Frameworků přikládám kvalitativní znaky (fakta) jako například: rok vzniku a licence atd.

Vlastnosti	Bootstrap	Foundation	Semantic UI
První vydání	2011	2011	2013
Popularita na GitHub	121,374 hvězd	26,956 hvězd	39,364 hvězd
Základní koncept	RWD, mobile first	RWD, mobile first, sémantika	Sémantika, responzivita
Velikost frameworku	578 KB	197.5 KB	806 KB
Preprocessor	Sass	Sass	Less
Set ikon	Ne	Foundation icon	Font Awesome
Licence	MIT	MIT	MIT
Podpora prohlížečů	Firefox, Chrome, Safari, IE810-11-Microsoft Edge.	Chrome, Firefox, Safari, IE9+; iOS, Android, Windows Phone 7+	Firefox, Chrome, Safari, IE10+ (IE9), Android 4, Blackberry 10
Rozšíření/doplňky	Mnoho rozšíření, ale jen třetích stran	Ano	Ne
Přizpůsobení balíčku	Základní GUI nastavení	Základní GUI nastavení	Pouze manuální nastavení bez GUI

Tabulka 3: Porovnání front-end frameworků

3.3 Relační databáze

Relační databáze je sada datových položek uspořádaných jako sada formálně popsaných tabulek, ze kterých mohou být data přístupná nebo znovu sestavena různými způsoby, aniž by bylo nutné reorganizovat databázové tabulky. Relační databáze byla vynalezena E. F. Coddem v IBM v roce 1970. [22]

Standardní uživatelské a aplikační programové rozhraní relační databáze je strukturovaný dotazovací jazyk (SQL). Příkazy SQL se používají jak pro interaktivní dotazy pro informace z relační databáze, tak pro shromažďování dat pro sestavy. [22]

Kromě relativně snadné tvorby a přístupu k relační databázi má významnou výhodu, že lze snadno rozšířit. Po vytvoření původní databáze lze přidat novou kategorii dat, aniž by bylo nutné upravovat všechny existující aplikace. [22]

Relační databáze je sada tabulek obsahující data vložená do předdefinovaných kategorií. Každá tabulka (která se někdy nazývá relací) obsahuje jednu nebo více kategorií dat ve sloupcích. Každý řádek obsahuje jedinečnou instanci dat pro kategorie definované ve sloupci. Například typická databáze pro obchodní potřeby by obsahovala tabulku, která popisuje zákazníka se sloupci pro jméno, adresu, telefonní číslo apod. Další tabulka by popisovala objednávku: produkt, zákazník, datum, prodejní cenu atd. Uživatel databáze by mohl získat pohled na databázi, která vyhovovala potřebám uživatele. Správce poboček by například mohl chtít zobrazit všechny zákazníky, kteří si po určitém datu koupili produkty. Správce finančních služeb ve stejné společnosti by mohl ze stejných tabulek získat zprávu o účtech, které je třeba zaplatit. [22]

Při vytváření relační databáze může být definována doména možných hodnot ve sloupci dat a další omezení, která se mohou vztahovat na danou hodnotu dat. Například doména případných zákazníků by mohla dovolit až deset možných jmen zákazníků, ale v jedné tabulce by měla být omezena, aby bylo možné specifikovat pouze tři z těchto názvů zákazníků.

Definice relační databáze vede k tabulce metadat nebo formálních popisů tabulek, sloupců, domén a omezení. [22]

3.3.1 Normalizace

Normalizace databáze je proces efektivního uspořádání dat v databázi. Existují dva hlavní důvody normalizačního procesu:

- Odstranění redundantních dat, například ukládání stejných dat do více tabulek.
- Zajištění, zda datové závislosti mají smysl.

Oba tyto důvody jsou hodnými cíli, protože snižují množství obsazeného prostoru, který databáze spotřebovává, a zajišťuje logické uložení dat. Normalizace se skládá ze série pokynů, které pomáhají při vytváření dobré struktury databáze. [23]

Hlavním cílem normalizace modelu je tedy zajištění reprezentace každé relace v databázi, optimalizace algoritmů vyhledávání, zjednodušení aktualizace a rušení vět. Pro dosažení normální formy se postupným reverzibilní procesem nahradí dané množiny relací, relacemi, které mají jednodušší strukturu. Slouží k dekompozičnímu návrhu tabulek s minimální redundancí dat. Hlavní přínosy datové normalizace jsou konzistentní neduplicitní data, zjednodušené aktualizace a vyhledávání dat. Veškeré dekomponované relace je možné opět funkčně spojit. [23]

Existují i další normální formy jako čtvrtá a pátá normální forma, nicméně jako zcela dostatečující je považována třetí normální forma. [23]

3.3.1.1 Jednotlivé normální formy

1. NF relace nesmí obsahovat násobná data (např. opakující se jméno – vertikálně). [23]
2. NF všechna neklíčová data relace musí funkčně záviset na celém primárním klíči. [23]
3. NF všechna neklíčová data musí záviset pouze na klíčových hodnotách a nikoliv mezi sebou. [23]

3.3.2 Kritéria pro porovnání relačních databází

Výběr relačních databází byl systematicky zúžen tak, aby databáze byla podporována každým z výše popisovaných PHP frameworků (Laravel, Nette a Symfony) bez nutnosti doinstalování rozšíření. Jako výsledný průnik podporovaných databází napříč frameworky vychází následující seznam:

- MySQL
- PostgreSQL
- SQLite

Pro objektivní porovnání a usnadnění výběru vhodné relační databáze k tvorbě malých až středně velkých webových aplikací byla stanovena následující kritéria:

- Bezpečnost
- Licence
- Rychlost
- Podpora
- Standardizace

Tato kritéria budou hodnocena napříč třemi vybranými relačními databázemi.

3.3.3 Popularita vybraných databází

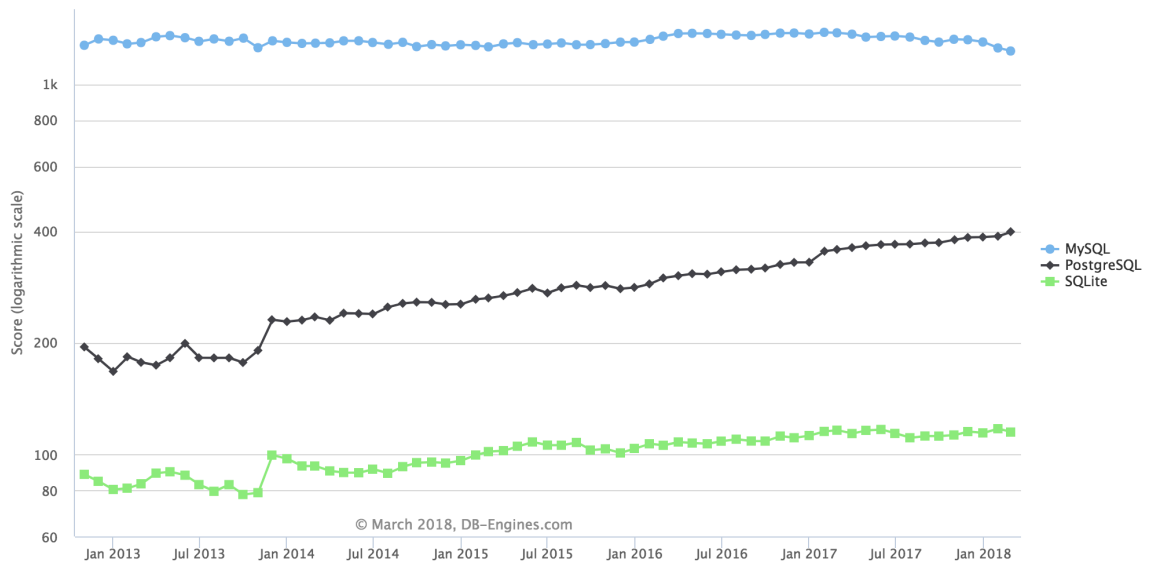
Popularita vybraných databází byla měřena podle serveru <https://db-engines.com>, který pro měření popularity využívá následující metodiku.

DB-Engines Ranking měří popularitu databází pomocí následujících parametrů:

- Počet zmínek systému na webových stránkách, měřeno jako počet výsledků dotazů vyhledávačů. V současné době využívá pro toto měření Google, Bing a Yandex. Aby byly počítány pouze relevantní výsledky, je hledáno <název systému> společně s termínem databáze, např. "Oracle" a "databáze". [24]
- Obecný zájem o systém. Pro toto měření se používá frekvenci vyhledávání v Google Trends. [24]
- Frekvence technických diskuzí o databázi. Používá se počet souvisejících otázek a počet zainteresovaných uživatelů na známých oblastech souvisejících s informačními technologiemi, které se týkají Stack Overflow a DBA Stack Exchange.

- Počet pracovních nabídek, ve kterých je systém uveden. Využívá se množství nabídek na hlavních vyhledávačích Indeed a Simply Hired. [24]
- Počet profilů v profesionálních sítích, ve kterých je systém uveden. Používají se nejoblíbenější profesionální sítě LinkedIn a Upwork. [24]
- Relevance v sociálních sítích. Počítá se počet Twitter tweets, ve kterých je systém uveden. [24]

Hodnotu popularity databáze se vypočítá standardizací a zprůměrováním jednotlivých parametrů. Tyto matematické transformace jsou vytvořeny tak, aby byly zachovány rozestupy jednotlivých databází. To znamená, že když systém A má dvakrát větší hodnotu v hodnocení DB-Engines jako systém B, pak je dvakrát tak populární, když je zprůměrován podle jednotlivých hodnotících kritérií. [24]



Podle uvedeného grafu je zřejmá největší popularita MySQL databáze, což je vzhledem popsaným kritériím hodnocení jen jeden z ukazatelů a pro výběr vhodné databáze. Pro konkrétní projekt může být klíčové zcela jiné kritérium, která budou popsána v další kapitole. [24]

3.3.4 MySQL

MySQL existuje od roku 1995 a je nyní vlastněna společností Oracle. Mimo open-source je k dispozici i několik placených vydání, které nabízejí další funkce, jako je geografická replikace klastru a automatické škálování. Vzhledem k tomu, že MySQL je průmyslovým standardem, je kompatibilní s téměř každým operačním systémem a je napsán jak v jazyce C, tak v jazyce C++. Toto řešení je skvělou volbou pro mezinárodní uživatele, protože server může poskytovat chybové zprávy klientům ve více jazycích, které zahrnují podporu několika různých sad znaků. [22]

Mimo to je nejoblíbenějším ze všech rozsáhlých databázových serverů. Je to bohatý, open-source produkt, který tvoří úložiště pro spoustu webových stránek a aplikací online. Začít s MySQL je poměrně snadné a vývojářům poskytuje přístup k masivnímu množství informací o databázi na internetu. [22]

Je třeba konstatovat, že vzhledem k popularitě produktu existuje spousta aplikací, nástrojů a integrovaných knihoven třetích stran, které velmi pomáhají s mnoha aspekty práce s tímto SŘBD. [22]

MySQL lze nainstalovat velmi snadno. Nástroje třetích stran, včetně vizuálních (tj. Grafických uživatelských rozhraní), usnadňují začátek práce s databází. MySQL podporuje mnoho funkcionalit SQL jazyka, které se očekávají od SŘBD - buď přímo nebo nepřímo. [22]

MySQL dokáže pracovat s velkým množstvím dat a dále může být v případě potřeby dobře škálovatelné. [22]

MySQL nemá v úmyslu pokrýt veškeré požadavky a přináší jistá funkční omezení, které mohou vyžadovat některé nejmodernější aplikace. Způsob, jakým určitá funkcionalita manipuluje s MySQL (např. Reference, transakce, audit apod.), jej činí trochu méně spolehlivou ve srovnání s jinými SŘBD. [22]

Přestože MySQL je stále technickým produktem s otevřeným zdrojovým kódem, existují stížnosti týkající se procesu vývoje od jeho akvizice. Je třeba poznamenat, že existují některé plně integrované databáze založené na MySQL, které mají přidanou hodnotu nad standardní instalací MySQL (například MariaDB). [22]

3.3.4.1 Hodnocení kritérií

Bezpečnost

Bezpečnost v MySQL je na relativně vysoké úrovni, je zde implementováno mnoho bezpečnostních prvků, které jsou spíše pokročilé.

Licence

MySQL je šíře pod open-source licencí GPL verze 2.

Rychlost

Odmítnutí některých standardů umožňuje MySQL pracovat velmi efektivně a vytvářet ne-standardní zkratky, čímž se dosáhne zvýšení rychlosti.

Podpora

Vzhledem k rozšířenosti MySQL je velice snadné najít již řešené úlohy na fórech, v tutoriálech nebo na StackOverflow.

Standardizace

Navzdory tomu, že se nepodařilo implementovat celý standard SQL, nabízí MySQL uživatelům spoustu funkcí. Jako samostatný databázový server, aplikace komunikují s MySQL skrz daemon proces pro přístup k samotné databázi, což například SQLite nedovede.

3.3.5 PostgreSQL

Po více než 15 letech aktivního rozvoje je PostgreSQL dalším velice oblíbený open-source SŘBD, který běží na všech hlavních operačních systémech, včetně Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64) a Windows. PostgreSQL je plně kompatibilní s ACID (Atomicity, Consistency, Isolation, Durability). [25]

PostgreSQL je moderní, open-source (objektově orientovaný) systém pro řízení báze dat, jehož hlavním cílem je být kompatibilní a rozšiřitelný. PostgreSQL se snaží splňovat normy ANSI / ISO SQL společně s revizemi. [25]

Vzhledem k dobré základní technologii je Postgres schopné velmi efektivně zvládnout mnoho úkolů. Podpora souběžnosti se dosahuje bez čtení zámek a díky implementaci kontroly řízení souběžného více-verzového přístupu (MVCC), který také zajišťuje kompatibilitu s ACID. [25]

PostgreSQL je dobře programovatelná databáze, a proto i snadno rozšiřitelná pomocí vlastních postupů, které se nazývají "uložené procedury". Tyto funkce mohou být vytvořeny pro zjednodušení provádění opakovaných, komplexních a často požadovaných operací databáze. [25]

Ačkoli tento SŘBD nemá popularitu jako MySQL, existuje mnoho úžasných nástrojů a knihoven třetích stran, které jsou navrženy tak, aby práce s PostgreSQL byla jednoduchá, a to i přes rozsáhlé jádro této databáze. V dnešní době je možné získat PostgreSQL jako balíček pomocí mnoha výchozích správců balíčků, integrovaných v operačních systémech. [25]

PostgreSQL není jen systém pro správu relačních databází, ale také pro objektivní přístup. Nicméně pro jednoduché operace čtení může být PostgreSQL zbytečně složitý systém a může se zdát méně výkonný než konkurence, jako je MySQL nebo SQLite. [25]

Pro potřeby tvorby databáze, která je velmi složitá je PostgreSQL ideální, poskytuje funkčnost pro bezproblémovou údržbu takové složitě databáze. [25]

3.3.5.1 Hodnocení kritérií

Bezpečnost

PostgreSQL Globální rozvojová skupina (PGDG) bere bezpečnost vážně, což přináší uživatelům důvěru ve webové stránky a aplikace postavené na PostgreSQL. Možnosti konfigurace pokrývají případy bezpečnostního selhání, bezpečný a robustní databázový server a dobrou integraci s jiným softwarem pro bezpečnostní infrastrukturu.

Licence

PostgreSQL je vydáváno pod liberální licenci open-source, která je podobná licenci BSD nebo MIT.

Rychlost

Pokud se od databáze očekává pouze rychlé čtení dat, pak PostgreSQL není vhodnou volbou a lépe pro tyto účely poslouží MySQL nebo SQLite.

Podpora

PostgreSQL je podporováno věrnou a zkušenou komunitou, která je přístupná bezplatně prostřednictvím znalostních bází a stránek Q&A 24 hodin denně a 7 dní v týdnu.

Standardizace

PostgreSQL se snaží splňovat normy ANSI / ISO SQL společně s revizemi.

3.3.6 SQLite

Samozvaná jako nejrozšířenější databáze na světě, SQLite byla vydána v roce 2000 a byla používána významnými společnostmi, jako jsou Apple, Facebook, Microsoft a Google. Každé vydání je pečlivě testováno, aby byla zajištěna spolehlivost. I když jsou ve vydání chyby, vývojáři SQLite jsou zcela upřímní ohledně možných nedostatků, takže poskytují seznamy chyb a chronologii změn kódu pro každé vydání. [10]

SQLite je úžasná knihovna, která je vložena do aplikace. Jedná se o samostatnou databázi založenou na souborech a nabízí úžasnou sadu nástrojů pro zpracování všech druhů dat s mnohem méně omezeními a snadnějším používáním v porovnání s hostovanými relačními databázemi založenými na procesech. [10]

Celá databáze se skládá z jediného souboru na disku, což z ní dělá nejlépe přenosnou databázi mezi výše zmíněnými. [10]

Během fáze vývoje většiny aplikací je pro mnoho lidí velmi pravděpodobné, že bude potřebovat řešení, které může škálovat podle potřeb. SQLite s bohatou základnou funkcí může nabídnout mnohem víc než to. Vývoj bude velice rychlý díky jednoduchosti práce s jediným souborem a propojenou knihovnou založenou na jazyku C. [10]

Pokročilé databáze nabízejí podporu více uživatelů, tj. řízených připojení s nastavenými oprávněními přístupu do databáze a tabulek. Vzhledem k účelu a charakteru SQLite (bez podpory vyšší úrovně souběžnosti více klientů) tato funkcionální v SQLite neexistuje. [10]

SQLite je vhodná pro všechny aplikace, které potřebují snadnou přenositelnost, které nevyžadují souběžnost, např. lokální aplikace pro jednoho uživatele, mobilní aplikace nebo hry. V mnoha případech mohou aplikace, které potřebují přímo číst nebo zapisovat soubory na disk, těžit z přechodu na SQLite pro další funkce a jednoduchost, které pocházejí z použití strukturovaného dotazovacího jazyka (SQL). [10]

Jedno z omezení SQLite je operace zápisu. Tento SŘBD umožňuje, aby se v kterémkoli okamžiku uskutečnil pouze jeden zápis, což umožňuje pouze omezenou propustnost. [10]

3.3.6.1 Hodnocení kritérií

Bezpečnost

Pokud je pro aplikaci požadována vysoká míra zabezpečení, SQLite rozhodně nebude nejlepší volbou a bude vhodnější zvolit MySQL nebo PostgreSQL.

Licence

SQLite je vydáváno pod open-source licencí konkrétně open-domain, což znamená, že si kdokoliv může vytvářet libovolný počet kopií a užívat jí bez omezení. Nicméně SQLite není otevřený zdrojový kód, tudíž není možné přispívat svými doplňky pro SQLite.

Rychlost

Přihlédneme-li k povaze SQLite nelze očekávat, aby bylo dosaženo většího výkonu. Knihovna je jednoduše laděná a jednoduchá k použití. Není tedy technicky možné ji učinit výkonnějším, než aktuálně je, nicméně už tak je pro jednoduché čtení výrazně rychlejší než MySQL a PostgreSQL.

Podpora

Bezplatná podpora pro SQLite je k dispozici na veřejném e-mailovém listu. E-mail je monitorován rozsáhlou komunitou odborníků, včetně hlavního vývojového týmu SQLite, který dokáže vyřešit téměř všechny problémy s SQLite, které mohou nastat. Mimo to také nabízí placenou privátní podporu.

Standardizace

Ačkoli to může vypadat jako "jednoduchá" implementace databáze, SQLite používá převážně standardní SQL jazyk. Bohužel má některé vlastnosti vynechané (RIGHT OUTER JOIN nebo FOR EACH STATEMENT), nicméně některé další jsou naopak přidány.

3.3.7 Porovnání relačních databází

3.3.7.1 MySQL

MySQL je systém pro řízení báze dat (SŘBD), který tvoří nejpoblíbenější databázové řešení pro webové aplikace. Díky velmi dobré integraci této databáze s Linuxem, Apache a PHP jsou na ní postaveny největší internetové portály jako např. Facebook, Google Twitter nebo YouTube, které ji používají k zobrazování dat. MySQL je napsán v jazyce C a C++, ale funguje na mnoha systémových platformách, jako například: AIX, FreeBSD, Linux, OS X, Microsoft Windows, Oracle Solaris a SunOS. MySQL projekt je více zaměřený na výkon než na kompatibilitu se standardem SQL. V průběhu let se však začal projekt věnovat v souladu s normou ISO / IEC 9075-1: 2003, která podporuje více primárních funkcí SQL jako jsou transakce, uložené procedury, spouštěče, pohledy, kurzory, rozdělení tabulek, rozvrh úkolů. Technologie MySQL nyní používá nejnovější standard SQL: 2003.

3.3.7.2 PostgreSQL

PostgreSQL je druhým nejoblíbenějším SŘBD, konkrétně objektově-relační databáze (ORDBMS). Na rozdíl od MySQL je PostgreSQL mnohem více v souladu s normami SQL, konkrétně SQL: 2011. Je k dispozici pro většinu z operačních systémů, ale především pro distribuce Linuxu. Nejdůležitější funkcí nástroje PostgreSQL je řízení více transakcí (MVCC) pro správu transakcí. V PostgreSQL lze také napsat předdefinovaná pravidla, která mohou přepisovat příchozí dotaz. To také podporuje mnohem více typů indexů, datových typů a velkých datových dotazů než MySQL. Technologie je plná doplňků, škálovatelnosti a přizpůsobení tak, aby vyhovovaly potřebám vývojářů a plnila požadavky aplikace.

3.3.7.3 SQLite

SQLite je integrovaný relační databázový engine. Jeho vývojáři jej nazývají samostatným databázovým, bezserverovým, bezkonfiguračním a transakčním SQL enginem. Je to velmi populární a dnes se používá stovky milionů kopií po celém světě. SQLite se používá v operačních systémech Solaris 10 a Mac OS a iPhone a Skype. Knihovna Qt4 má vestavěnou podporu pro SQLite, stejně jako jazyky Python a PHP. Mnoho populárních aplikací používá SQLite interně, jako jsou Firefox, Google Chrome nebo Amarok.

3.3.8 Tabulka porovnání relačních databází

Pro závěrečné hodnocení zvolených relačních databází jsem vytvořil tabulku, kde ke každému kritériu všech databází přikládám kvalitativní znaky (fakta) jako například: rok vzniku a licence atd.

Vlastnosti	MySQL	PostgreSQL	SQLite
První vydání	1995	1989	2000
Jazyk implementace	C a C++	C	C
Popularita db-engines.com	1228 [18]	399 [18]	114 [18]
Server-side skriptování	Ano (proprietální)	Ano (proprietální)	Ne
Cizí klíče	ano	ano	ano
Metody replikace	Master-master Master-slave	Master-slave	žádné
Licence	GPL verze 2	BSD	Public domain
Podporované servery	FreeBSD Linux OS X Solaris Windows	FreeBSD HP-UX Linux NetBSD OpenBSD OS X Solaris Unix Windows	Bez serverová implementace
Řízení přístupu uživatelů	Ano	Ano (podle SQL standartu)	Ne

Tabulka 4: Porovnání relačních databází

3.4 UX

Návrháři UX se primárně zabývají tím, jaký mají pocit z webu. Určitý návrhový problém nemá jednu správnou odpověď. Návrháři UX prozkoumají mnoho různých přístupů k řešení konkrétního problému s uživatelem. Širokou odpovědností návrháře UX je zajistit, aby produkt logicky plynul z jednoho kroku do druhého. Jedním ze způsobů, jak to může návrhář UX udělat, je provedení osobních testů uživatele, který je sledován. Za pomoci identifikace verbálních a neverbálních znaků během uživatelských testů vylepšují návrh, aby vytvořili co nejlepší uživatelskou zkušenost. Příkladem projektu je vytvoření nového uživatelského prostředí pro příjemný proces učení se s novým SW. [26]

Definuje modely interakcí, toky úloh uživatelů a specifikuje uživatelské rozhraní. Komunikuje scénáře, zážitky od začátku do konce, modely interakcí a návrhy obrazovky pro zúčastněné strany. Spolupracuje s vizuálními designéry, aby začlenili vizuální identitu do funkcí. Vytváří a udržuje návrh wireframů, modelů a specifikací podle potřeby. [26]

3.4.1 Wireframe

Wireframe je pouze hrubý průvodce pro uspořádání webových stránek nebo aplikací, je to návrh, který je nejvíce spojený s UX Designéry. [26]

Jakmile designéři vytvoří sérii statických obrazovek, pak je velice snadné převést je za pomoci dnešních moderních nástrojů (jako jsou Balsamiq, Mockups a Axure RP) do interaktivního prototypu bez psaní jakéhokoli kódu. [26]

Zatímco mnoho UX designérů zdůrazňuje, že jsou více než jen tvůrci drátěných modelů, ale také je pravda, že mnoho UX designérů začíná hlavně s wireframy. Vytváření základního rozvržení stránek je něco, co zvládne udělat téměř kdokoliv, protože nástroje pro jejich tvorbu jsou velice snadné k naučení. [26]

3.4.2 Uživatelské testování

Základním kamenem uživatelského testování je uživatel sedící před webovými stránkami nebo aplikací, který nahlas přemýšlí nad předem naplánovanými úkoly. [26]

Kolik účastníků se do testu zapojí, jak úzce se účastníci testů shodují se skutečnými uživateli a kolik iterací testování je spuštěno, jsou všechna rozhodnutí určená rozpočtem a časovými omezeními. [26]

Testování uživatelů je tak jednoduché, že by to měl každý alespoň jednou vyzkoušet. Bytí ve stejné místnosti, zatímco někdo se snaží použít navržený produkt, je silným spouštěčem pro vytváření empatie s uživateli. [26]

3.4.3 Persony

Persona je fiktivní identita, která odráží jednu ze skupin uživatelů, pro které je web či aplikace navrhována. [26]

Persony musí být podle výzkumu užitečné. Zde není příliš vhodné zapojovat kreativitu a vymýšlet detaily, tak aby byly persony zajímavé. Cílem je, aby persony odrážely vzory, které jsou identifikovány u cílových uživatelů (nebo potenciálních uživatelů). [26]

Není možné nijak zkrátit proces identifikace těchto modelů, pocházejících z uživatelského výzkumu, který je založen na: provádění rozhovorů, průzkumů, testování uživatelů, kontextuálního dotazování a dalších činnostech. [26]

3.4.4 Scénáře a storyboard

Scénář je příběh popisující "jeden den v životě" jedné z cílových person, včetně toho, jak webové stránky nebo aplikace zapadá do jejich života. Pro psaní scénářů je důraz kladen na pravidelné použití, ne na okrajové případy. [26]

V závislosti na publiku může být storyboard vhodnějším nástrojem pro zjištění, kdy, kde a proč by někdo mohl používat cílový produkt. [26]

Storyboard je inspirovaný filmařským průmyslem. Jedná se o vizuální sekvence událostí používaných k zachycení interakcí uživatele s produktem. Může to být extrémně hrubý náčrtek, čistý návrh pro krystalizaci vlastních nápadů, a nebo učený komiks pro účinnější zapojení publika. [26]

3.5 Use case

Use case je termín softwarového a systémového inženýrství, který popisuje, jak uživatel používá systém k dosažení konkrétního cíle. Use case je technika modelování softwaru, která definuje funkce, které mají být implementovány, a řešení případných chyb, které mohou nastat. [27]

Use case definuje interakce mezi externími aktéry a systémem za účelem dosažení konkrétních cílů. Existují tři základní prvky, které tvoří případ použití: [27]

- Actors: Herci jsou typem uživatelů, kteří interagují se systémem. [27]
- System: Případy použití zachycují funkční požadavky, které určují zamýšlené chování systému. [27]
- Goals: Případ užití je zpravidla iniciován uživatelem k plnění cílů, popisuje činnosti a varianty, které se jsou součástí procesu k dosažení cíle. [27]

Use case je modelován pomocí jednotného modelovacího jazyka UML a jsou reprezentovány ovály obsahujícími názvy případu použití. Actors jsou zastoupeni pomocí čar s názvem Actor, napsaným pod objektem. Chceme-li znázornit spojení Actor se systémem, realizujeme to spojovací čarou. Ohraničení kolem use case představuje hranici systému. [27]

3.5.1 User stories

User stories jsou jedním z hlavních vývojových artefaktů pro projektové týmy Scrum a Extreme Programming (XP). User stories je definice požadavku na vysoké úrovni, která obsahuje dostatek informací, aby vývojáři mohli přiměřeně odhadnout úsilí o jeho realizaci. [27]

3.5.2 Use case scénář

Je snadné zaměnit definici use case s definicí use case scénáře. Use case představuje akce, které jsou zapotřebí k dosažení nebo nedosažení cíle. Use case má několik "cest", které mohou být kdykoli obsazeny jakýmkoli uživatelem. Use case scénář je jedinou cestou v use case diagramu. [27]

4 Vlastní práce

Na základě výše uvedené analýzy technologií nezbytných pro tvorbu webových aplikací byly implementovány některé vybrané technologie do informačního systému na webu a tím došlo k ověření výsledků z analýzy uvedených technologií.

Jako vhodný projekt pro implementaci vybraných technologií byl stanoven inzertní portál cílený na úzce zaměřené produkty, konkrétně na prodej a nákup dílů pro sportovní vozy.

Toto zaměření informačního webového portálu jsem si vybral na základě zkušenosti s prodejem a koupí různých dílů určených pro sportovní vozy. Trendem poslední doby je prodej takto specifického zboží skrze uzavřené skupiny na Facebooku, na různých fórech nebo na portálu bazos.cz. Nicméně tento proces je poměrně komplikovaný, protože často takových Facebookových skupin existuje celá řada, není možné v nich vhodně filtrovat to co konkrétně uživatel hledá a nezbyvá nic jiného než použít vyhledávání podle textu, a nebo pravidelně sledovat všechny tyto skupiny, zda se zde daná součástka neobjeví. Ta samá situace nastává v případě hledání na různých fórech. Ani na portále bazos.cz není cesta k požadovanému dílu nijak jednoduchá, je sice možné zde procházet různé kategorie, nicméně vzhledem k povaze portálu, která je velice obecně navržena pro prodej čehokoliv od chovatelských potřeb, přes elektroniku až po díly na automobily, je tedy poměrně komplikované hledat konkrétní součástku. Uživatel tedy i v případě portálu bazos.cz je nucen procházet velké množství inzerátů a velmi často se stává, že ani nenajde požadovaný díl. Hlavní motivací pro tvorbu tohoto inzertního portálu je tedy zjednodušení celého tohoto procesu.

Osobně jsem si vybral téma, které je mi blízké a je cílené na úzkou skupinu uživatelů. Nicméně tato myšlenka v případě potvrzení hypotézy, že uživatelé uvítají inzertní portál úzce cílený na konkrétní produkty, bude možná aplikovat na jakýkoliv jiný portál, kde se budou prodávat náhradní díly například pro veterány a budou tak suplovat různé burzy nebo bleší trhy.

4.1 User stories inzertního portálu

Jako **zákazník**, chci vytvořit účet, abych mohl ukládat/mazat inzeráty do/z oblíbených, mohl se vrátit do historie navštívených inzerátů a mohl udělit hodnocení prodejci.

- Zadat celé jméno, email, tel. Číslo.
- Ověřit, zadat heslo a potvrdit.

Jako **zákazník**, chci mít možnost hledat inzeráty podle značky, modelu automobilu, ceny a stavu dílu.

Jako **zákazník**, chci mít možnost hledat inzeráty podle kategorie, upřesňující podkategorie, ceny a stavu dílu.

Jako **zákazník**, chci vidět má poslední hledání.

Jako **zákazník**, chci mít možnost řadit nalezené inzeráty podle data přidání inzerátu a ceny.

Jako **zákazník**, chci mít možnost filtrovat inzeráty podle kraje, kde se prodejce nachází.

Jako **zákazník**, chci vidět detaily inzerátu.

Jako **zákazník**, chci vidět hodnocení prodejce a další inzeráty prodejce.

Jako **zákazník**, chci mít možnost kontaktovat prodejce e-mailem, telefonicky.

Jako **zákazník**, chci mít možnost sdílet inzerát na sociálních sítích a e-mailem.

Jako **prodejce**, chci se zaregistrovat, abych mohl přidat inzerát.

- Zadat celé jméno, email, tel. Číslo.
- Ověřit, zadat heslo a potvrdit.

Jako **prodejce**, chci přidat inzerát.

- Titulek inzerátu, výrobce dílu, stav dílu, cena, popis, kraj, kategorie, podkategorie, značka automobilu, model automobilu, fotografie.
- Uložit inzerát.

Jako **prodejce**, chci být kontaktován emailem, telefonem od zákazníků.

Jako **prodejce**, chci mít možnost mazat a upravovat své inzeráty.

4.2 Use case scénáře inzertního portálu

Portál vyžaduje pro určité kroky identifikaci uživatele, na základě registrace a následně přihlášení do systému. V rámci use case scénáře se vyskytují 2 role uživatelů – zákazník a prodejce, aby bylo jasné, jakého cíle chce daná role dosáhnout, nicméně systém obecně nerozlišuje role uživatelských účtů a tedy registrace i přihlášení je pro obě role zcela totožné.

Registrace

1. Uživatel klikne v hlavním menu na tlačítko „Registrovat“.
2. Systém zobrazí formulář pro registraci.
3. Uživatel vyplní celé jméno, email, telefon a heslo a klikne na tlačítko „Registrovat“.
4. Systém ověří vyplněné údaje, zaregistruje uživatele na portálu a informuje uživatele o úspěšném dokončení registrace.

Přihlášení uživatele

1. Uživatel klikne v hlavním menu na tlačítko „Přihlásit“.
2. Systém zobrazí formulář pro přihlášení.
3. Uživatel vyplní své registrační údaje a klikne na tlačítko pod formulářem „Přihlásit“.
4. Systém ověří vyplněné údaje, identifikuje uživatele a zobrazí tlačítka v hlavním menu „Moje menu“ a „Odhlásit“.

Odhlášení

1. Přihlášený uživatel klikne v hlavním menu na tlačítko „Odhlásit“.
2. Systém uživatele odhlásí a informuje uživatele o úspěšném ohlášení.

4.2.1 Zákazník

Najít inzerát – kontaktovat prodejce

1. Zákazník zadá kritéria do systému, podle kterých chce hledat.
2. Systém nabídne klientovi inzeráty podle zadaných kritérií.
3. Zákazník si vybere vhodný inzerát a klikne na něj.
4. Systém zobrazí detaily inzerátu.
5. Zákazník klikne na tlačítko „Kontaktovat prodejce“.
6. Systém otevře e-mailový klient s předvyplněným kontaktem na prodejce.

Najít inzerát – uložit do oblíbených

1. Přihlášený zákazník zadá kritéria do systému, podle kterých chce hledat.
2. Systém nabídne klientovi inzeráty podle zadaných kritérií.
3. Zákazník si vybere vhodný inzerát a klikne na něj.
4. Systém zobrazí detaily inzerátu.
5. Zákazník klikne na tlačítko „Přidat do oblíbených“.
6. Systém uloží inzerát do seznamu oblíbených.

Najít inzerát – přidat hodnocení prodejce

1. Přihlášený zákazník zadá kritéria do systému, podle kterých chce hledat.
2. Systém nabídne klientovi inzeráty podle zadaných kritérií.
3. Zákazník si vybere vhodný inzerát a klikne na něj.
4. Systém zobrazí detaily inzerátu.
5. Zákazník klikne na tlačítko „Přidat hodnocení“.
6. Systém zobrazí formulář s hodnocením na stupnici 1-5 hvězd a pole pro komentář.
7. Uživatel vyplní formulář a klikne na tlačítko „Uložit“.
8. Systém uloží zadané hodnocení a vrátí uživatele na detail inzerátu.

Najít inzerát – zobrazit hodnocení prodejce

1. Zákazník zadá kritéria do systému, podle kterých chce hledat.
2. Systém nabídne klientovi inzeráty podle zadaných kritérií.
3. Zákazník si vybere vhodný inzerát a klikne na něj.
4. Systém zobrazí detaily inzerátu.
5. Zákazník klikne na odkaz se jménem prodejce.
6. Systém zobrazí seznam hodnocení prodejce.

Najít inzerát – zobrazit ostatní inzeráty prodejce

1. Zákazník zadá kritéria do systému, podle kterých chce hledat.
2. Systém nabídne klientovi inzeráty podle zadaných kritérií.
3. Zákazník si vybere vhodný inzerát a klikne na něj.
4. Systém zobrazí detaily inzerátu.
5. Zákazník klikne na tlačítko „Zobrazit inzeráty prodejce“.
6. Systém zobrazí seznam dalších inzerátů prodejce.

7. Zákazník klikne na vybraný inzerát.
8. Systém zobrazí klientovi detaily inzerátu.

Najít inzerát – zobrazit podobný inzerát

1. Zákazník zadá kritéria do systému, podle kterých chce hledat.
2. Systém nabídne klientovi inzeráty podle zadaných kritérií.
3. Zákazník si vybere vhodný inzerát a klikne na něj.
4. Systém zobrazí detaily inzerátu.
5. Zákazník klikne na inzerát ve spodní části obrazovky nazvané „Podobné inzeráty“.
6. Systém zobrazí detaily inzerátu.

Najít inzerát – řazení inzerátů

1. Zákazník zadá kritéria do systému, podle kterých chce hledat.
2. Systém nabídne zákazníkovi inzeráty podle zadaných kritérií.
3. Zákazník si zadá kritérium, podle kterého chce inzeráty řadit.
4. Systém zobrazí inzeráty seřazené podle zvoleného kritéria.
5. Zákazník si vybere vhodný inzerát a klikne na něj.
6. Systém zobrazí klientovi detaily inzerátu.

Zobrazit oblíbený inzerát

1. Přihlášený zákazník klikne v hlavní liště na tlačítko „Moje menu“.
2. Systém zobrazí nabídku menu.
3. Zákazník klikne na tlačítko „Oblíbené inzeráty“.
4. Systém zobrazí seznam oblíbených inzerátů.
5. Zákazník si vybere hledaný inzerát a klikne na něj.
6. Systém zobrazí detaily inzerátu.

Zobrazit navštívené inzerát z historie

1. Přihlášený zákazník klikne v hlavní liště na tlačítko „Moje menu“.
2. Systém zobrazí nabídku menu.
3. Zákazník klikne na tlačítko „Historie navštívených inzerátů“.
4. Systém zobrazí seznam navštívených inzerátů.

4.2.2 Prodejce

Vložit inzerát

1. Přihlášený prodejce klikne na tlačítko nad horním menu „Přidat inzerát“.
2. Systém zobrazí formulář pro zadání detailů inzerátů.
3. Prodejce vyplní detaily inzerátu a stiskne tlačítko „uložit“.
4. Systém inzerát uloží a přesměruje uživatele na detail vloženého inzerátu.

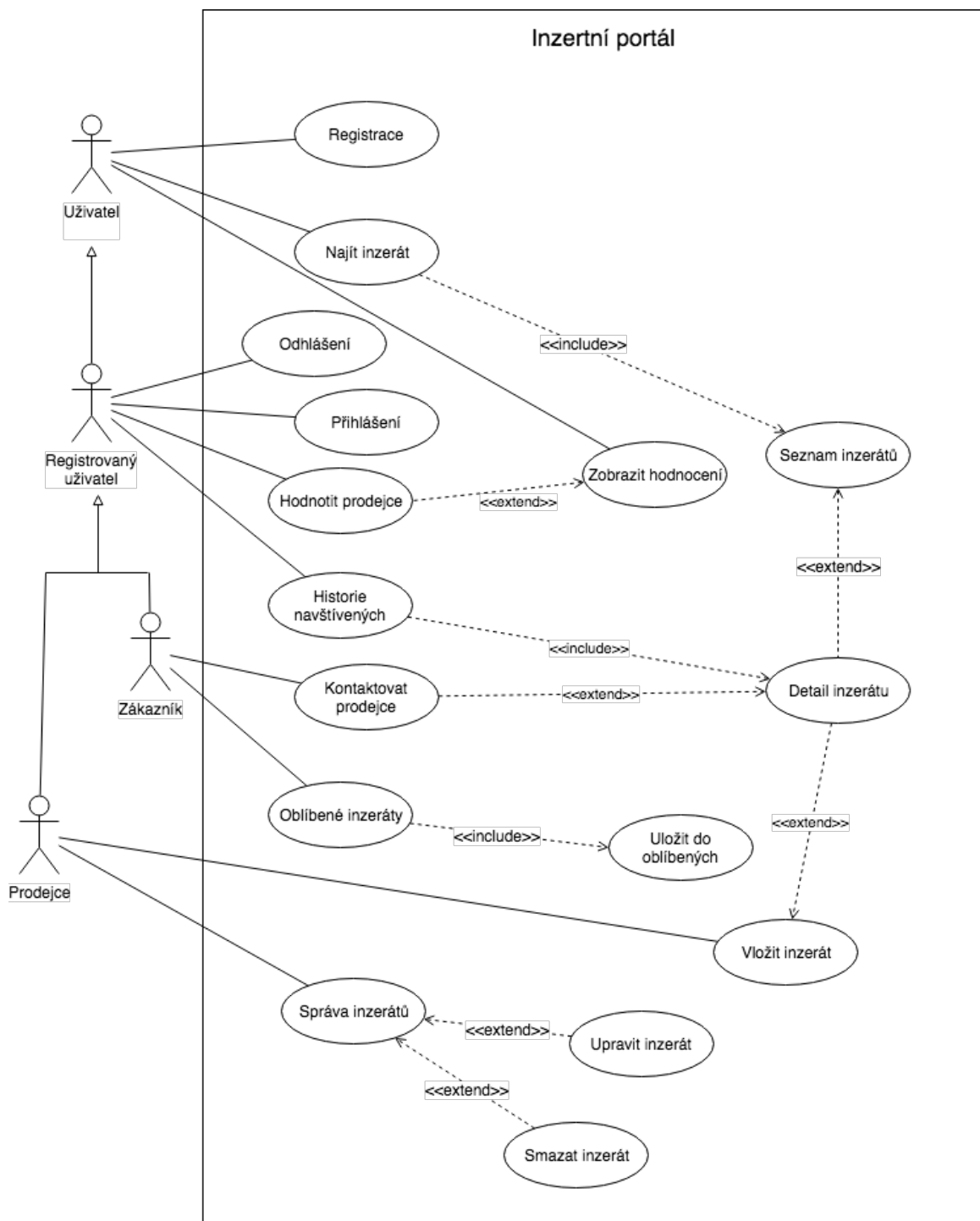
Upravit inzerát

1. Přihlášený prodejce klikne v hlavní liště na tlačítko „Moje menu“.
2. Systém zobrazí nabídku menu.
3. Zákazník klikne na tlačítko „Moje inzeráty“.
4. Systém zobrazí seznam uživatelem vytvořených inzerátů.
5. Prodejce si vybere hledaný inzerát a klikne na tlačítko „Upravit inzerát“.
6. Systém zobrazí formulář pro úpravy inzerátu.
7. Prodejce upraví detaily inzerátu ve formuláři a uloží úpravy pomocí tlačítka „Uložit“.
8. Systém uloží úpravy inzerátu a přesměruje prodejce na detail inzerátu.

Smazat inzerát

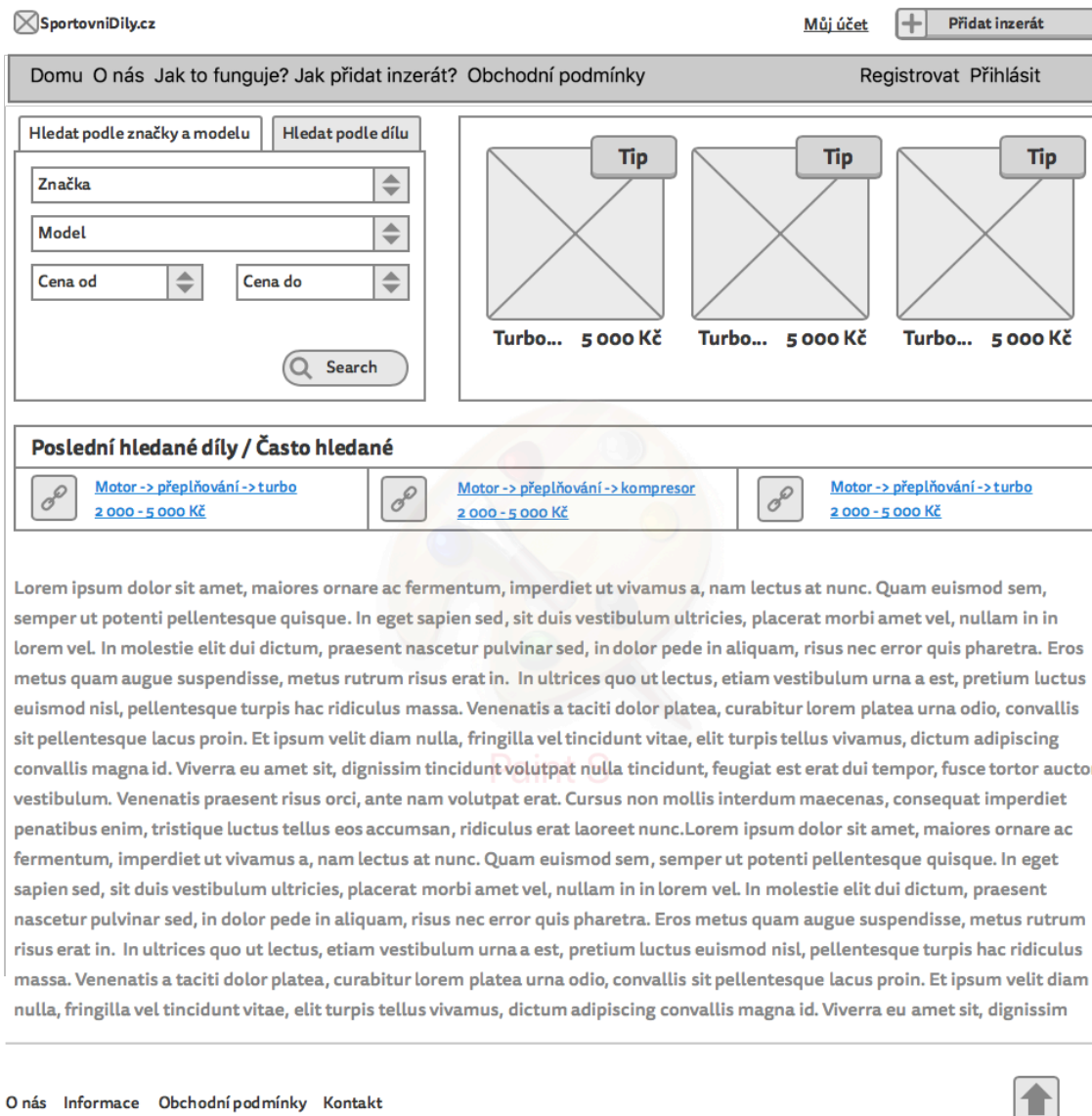
1. Přihlášený prodejce klikne v hlavní liště na tlačítko „Moje menu“.
2. Systém zobrazí nabídku menu.
3. Zákazník klikne na tlačítko „Moje inzeráty“.
4. Systém zobrazí seznam uživatelem vytvořených inzerátů.
5. Prodejce si vybere hledaný inzerát a klikne na tlačítko „Smazat inzerát“.
6. Systém se zeptá uživatele, zda opravdu chce inzerát smazat.
7. Prodejce klikne na tlačítko „Ok“.
8. Systém inzerát smaže.

4.2.3 Use case diagram

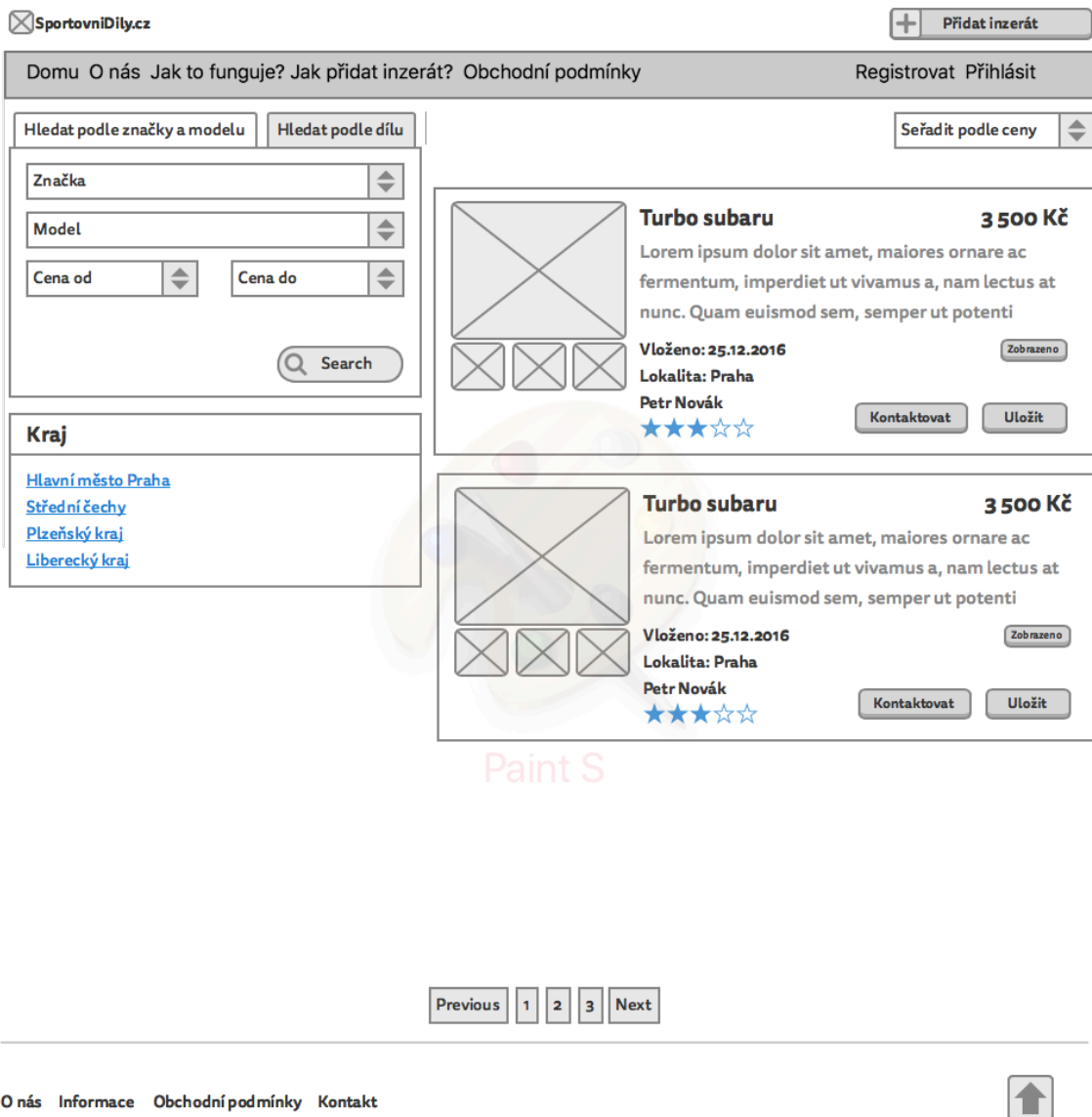


Obrázek 4: Use case diagram (vlastní tvorba)

4.3 Návrh wireframů inzertního portálu



Obrázek 5: Wireframe homepage (vlastní tvorba)



Obrázek 6: Wireframe výsledky vyhledávání (vlastní tvorba)

Turbo subaru TDo4**Cena: 3 500 Kč**

Soukromý prodejce: Petr Novák

★★★★☆ (Hodnocení 23x)

Vloženo: 25.12.2016

Lokalita: Praha

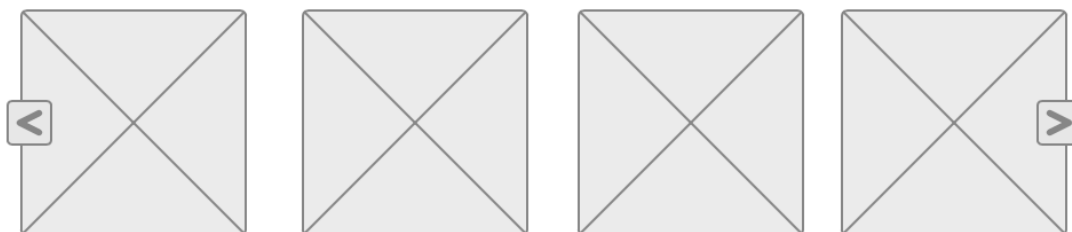
Tel.: +420 777 111 222

Kontaktovat prodejce

Uložit

232x232
Advertisement

Lorem ipsum dolor sit amet, maiores ornare ac fermentum, imperdiet ut vivamus a, nam lectus at nunc. Quam euismod sem, semper ut potenti pellentesque quisque. In eget sapien sed, sit duis vestibulum ultricies, placerat morbi amet vel, nullam in in lorem vel. In molestie elit dui dictum, praesent nascetur pulvinar sed, in dolor pede in aliquam, risus nec error quis pharetra. Eros metus quam augue suspendisse, metus rutrum risus erat in. In

Podobné inzeráty / Další díly prodejce

Obrázek 7: Wireframe detail inzerátu (vlastní tvorba)

4.4 Návrh MySQL databáze

Pro uchování a správu dat inzertního portálu byl na základě předchozí analýzy uvedené v kapitole 5.7 vybrán jako nejvhodnější SŘBD MySQL a to pro své přednosti v rychlosti, snadné konfiguraci, dostupnosti podpory ve formě různých tutoriálů online a obecně širokému rozšíření prakticky na každém webovém hostingu, pak tedy nebude problém s nasazením portálu na jakýkoliv běžný webhosting.

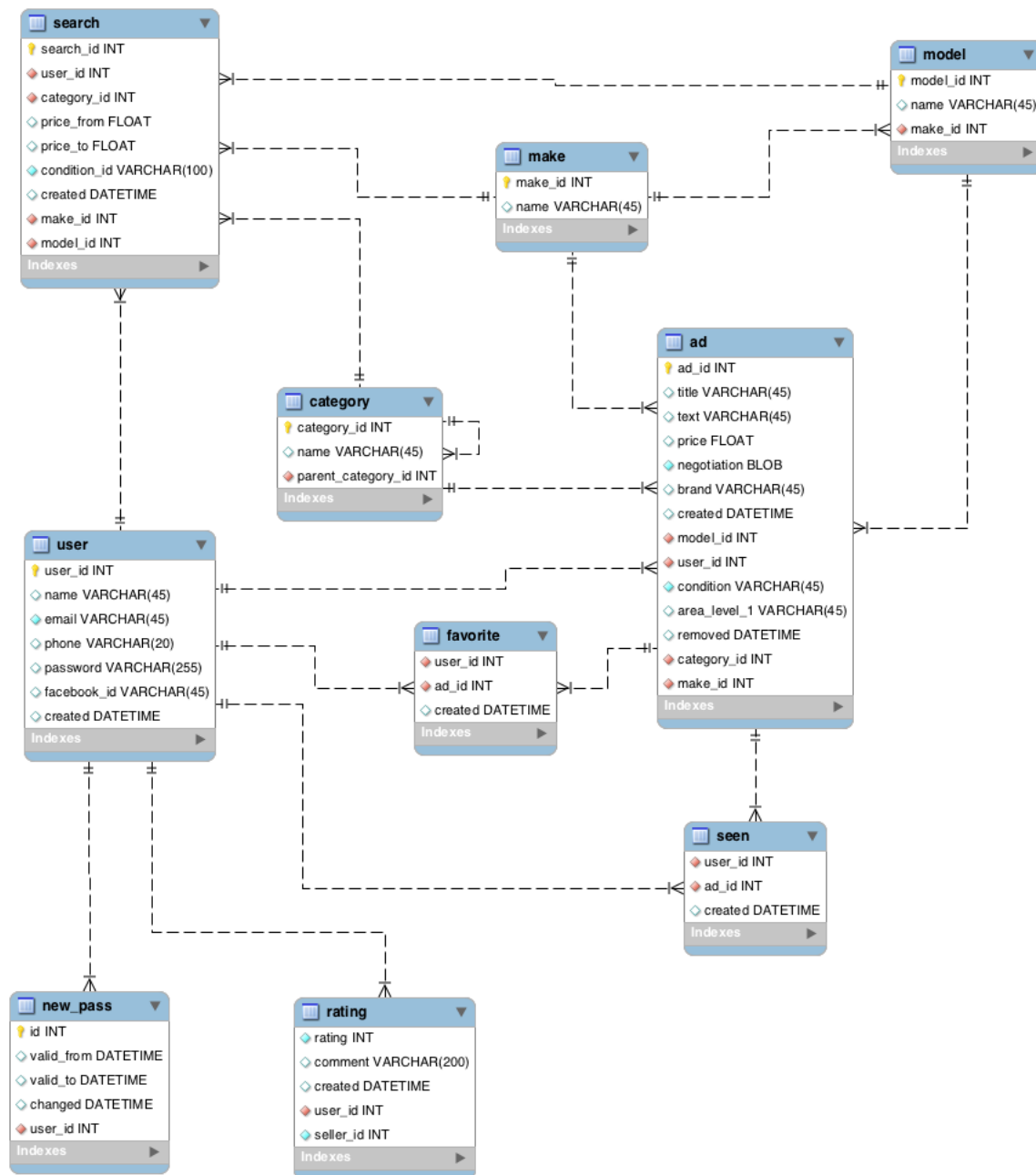
4.4.1 MySQL Workbench

Pro návrh databáze byl použit software MySQL Workbench, který zjednodušuje návrh a údržbu databáze, automatizuje časově náročné a k chybám náchylné úkoly. Umožňuje designovat model databáze, což je nejúčinnější metoda pro vytváření dobře fungující databáze, a zároveň poskytuje flexibilitu reagovat na vyvíjející se požadavky. Model a Schema validation pomůcky umožňují dodržování praktických pravidel pro modelování báze dat.

4.4.2 Popis databáze

- **Ad** – tabulka, která tvoří základ celého portálu, slouží k uchování veškerých dat souvisejících s inzerátem jako je například: titulek inzerátu, popis dílu, cena dílu atd.
- **Category** – uchovává data o kategoriích, tvoří relaci společně s tabulkou ad a slouží k třídění inzerátů do stromu kategorií.
- **Favorite** – tvoří relaci mezi tabulkama ad a user, slouží k zařazení inzerátu mezi oblíbené inzeráty přihlášeného uživatele.
- **Make** – tabulka, která uchovává data značek automobilů, podle kterých se pak třídí inzeráty do skupiny, pro kterou značku automobilu jsou díly určeny.
- **Model** – tabulka uchovává data o model automobilů, které jsou podmnožinou značek automobilů a společně s tabulkama ad a make slouží k roztřídění inzerátu do skupin, pro který model automobilu, jsou díly určeny.
- **New_pass** – tabulka slouží ke generování odkazů pro obnovu zapomenutého hesla.
- **Ratings** – tabulka tvoří relaci s tabulkou user a slouží k hodnocení prodejce.
- **Search** – tabulka slouží k uložení posledního vyhledávání uživatele, aby se k němu mohl později snadno vrátit.
- **Seen** – tabulka slouží k zaznamenání historie navštívených inzerátů.
- **User** – tabulka slouží k uložení dat o uživateli a možnosti autentizace uživatele.

4.4.3 Návrh MySQL databáze

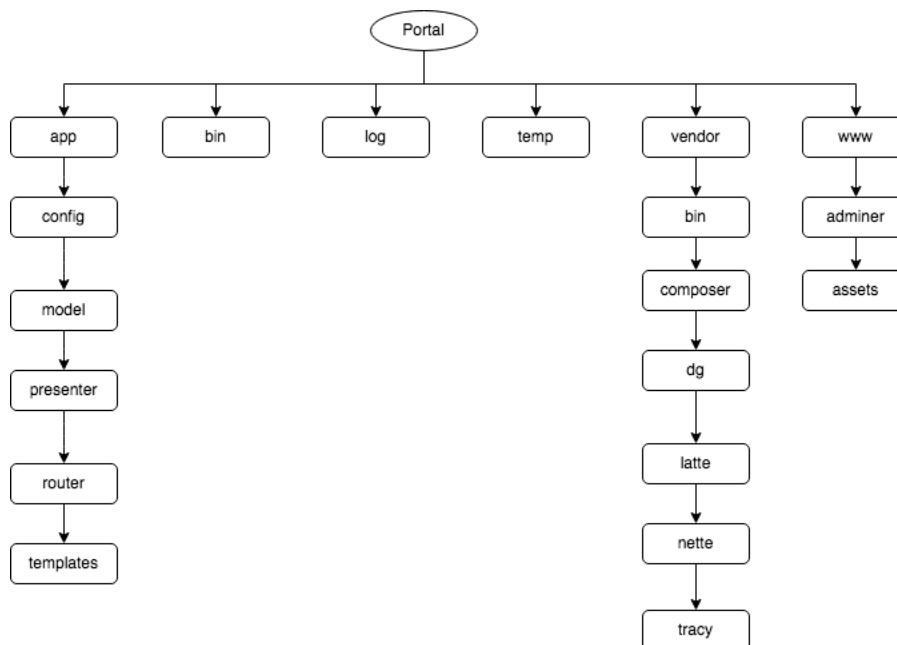


Obrázek 8: Návrh MySQL databáze

4.5 Implementace inzertního portálu za pomoci Nette Framework

Na základě analýzy PHP framework byl vybrán framework Nette jako vhodný pro tvorbu inzertního portálu. Hlavními důvody pro výběr Nette byly jeho přednosti týkající se vysoké úrovně zabezpečení aplikace, přehledné adresářové struktury a velmi snadnému konfiguraci.

Jak již bylo zmíněno v kapitole o Nette framework, jedná se o objektově orientovaný framework s podporou MVC resp. MVP architekturu. V souladu s tímto vzorem je vyvíjen i celý portál. Adresářová struktura byla kompletně převzata z Nette, tedy aplikace byla vyvinuta v adresáři app, kde je dodržena architektura MVP, s tím, že view je zde reprezentováno adresářem templates. Dále byl rozšířen adresář www, který byl rozšířen o css šablony a javascriptové pluginy pro tvorbu uživatelsky přívětivého GUI.



Obrázek 9: Adresářová struktura aplikace (vlastní tvorba)

4.5.1 Model

Model v MVP architektuře představuje vrstvu nesoucí samotnou logiku aplikace, jak již bylo zmíně v kapitole o MVC/MVP. Stěžejní logikou inzertního portálu jsou různé operace nad daty, jako je vkládání inzerátu, úpravy inzerátu, přidání fotografie, kategorizace inzerátů, filtrování inzerátů, správa a řízení uživatelských účtů atd. V souhrnu se tedy jedná o práci s databází a s file systémem pro ukládání fotografií na server.

V modelu byla vytvořena třída Base, která je potomkem Nette\Object a představuje základní třídu pro práci s databází. Všechny třídy, představující modely, které pracují s databází, dědí metody právě od třídy Base. V třídě Base je vytvořen konstruktor pro připojení k databázi a metody pro nejčastější operace s databází, jako jsou například: insert, update, delete, count, find atd. Výsledkem je tedy čitelný kód a velmi rychlý vývoj dalších tříd, pracujících s databází.

4.5.2 Presenter

Presenter jak již bylo řečeno v kapitole o Nette zastupuje v architektuře MVC funkci controlleru, což je je komponenta, se kterou komunikuje uživatel. Předá jí parametry a ona mu vrátí HTML stránku. Presenter typicky parametry předá modelům, od kterých získá data a ty pak předá view (templates), které data začlení do konkrétního HTML kódu. Tento HTML kód pošle presenter uživateli do prohlížeče.

Stejně jako u modelu je zde využita dědičnost pro zamezení duplicitního kódu a zlepšení čitelnosti kódu, veškré presentery jsou tedy potomci abstraktního presenteru BasePresenter, který například obstarává registraci a přihlašování uživatelů napříč celým webem.

Nette nabízí velmi efektivní a bezpečný způsob práce s formuláři, což je vzhledem k povaze inzertního portálu prakticky nejvíce využívaná funkcionalita. Třída Nette\Application\UIForm umožňuje definici vstupů a zároveň ošetření vstupních hodnot na jednom místě, takže není nutné řešit validaci na front-endu a na straně server zvlášť. Příkladem může být třeba registrační formulář viz. komponenta níže.

```
protected function createComponentRegistration()
{
    $form = new Form;
    $form->addText( name: 'name')
        ->setRequired( value: "Zadejte prosím vaše jméno.")
        ->addRule( validator: Form::MIN_LENGTH, message: 'Zadejte prosím alespoň dva znaky.', arg: 2);
    $form->addText( name: 'email')
        ->setRequired( value: "Zadejte prosím váš e-mail.")
        ->addRule( validator: Form::EMAIL, message: 'Zadejte prosím správný formát e-mailu.');
```

```
$form->addText( name: 'phone')
    ->setRequired( value: "Zadejte prosím váš telefon.")
    ->addRule( validator: Form::PATTERN, message: 'Telefonní číslo není ve správném tvaru', arg: '^[\+]?[()0-9. -]{9,}$');
```

```
$form->addPassword( name: 'password')
    ->setRequired( value: "Zadejte prosím heslo")
    ->addRule( validator: Form::MIN_LENGTH, message: 'Heslo musí mít alespoň %d znaků', arg: 8);
    $form->addSubmit( name: 'submit');
```

```
// call method signInFormSucceeded() on success
$form->onSuccess[] = array($this, 'registrationSucceeded');
return $form;
}
```

Obrázek 10: Komponenta pro registraci uživatele (vlastní tvorba)

4.5.3 View

View je v Nette reprezentováno za pomoci templates resp. šablonovacím systémem Latte, jak již bylo zmíněno v kapitole o Nette. Předností Latte je především implicitní bezpečnost, snadné vytváření cyklů a větvení, přehlednost kódu, a užitečné makra, které umožňují různé operace obecně s řetězci, například ořez textu, formátování číslovek, formátování data a mnoho dalších funkcí.

Veškeré šablony pro jednotlivé stránky jsou postaveny na výchozí šabloně (označované většinou jako @layout.latte), která tvoří základ určité části webu nebo i celého webu, v případě inzertního portálu tvoří hlavičku a patičku. Do této šablony pak mezi patičku a hlavičku includeje tělo každé stránky zvlášť. Mimo to je možné možné jednotlivé části kódu zabalit do tzv. bloku, který lze znovu používat v různých šablonách, tak aby nevznikal duplicitní kód.

Níže na obrázku je vidět jak vhodně se dá využít Latte pro výpis seznamu inzerátů. Kód je na první pohled velice dobře čitelný i přes používání větvení, cyklů, ořezu textů, formátování data nebo číslovek. Latte také umí efektivně generovat odkazy na stránky v rámci aplikace, takže není nutné řešit složitě SEO friendly url, ty se vytvářejí automaticky na základě nastavení routování.

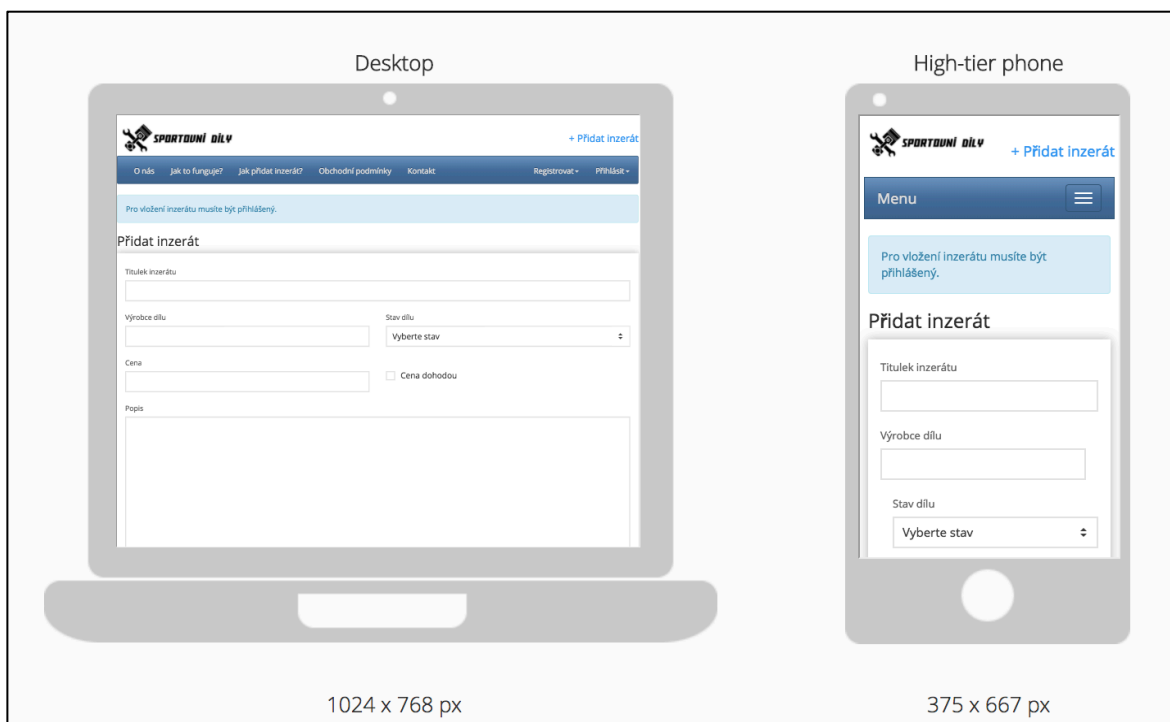
```
{if $results}
  {foreach $results as $id=>$row}
    <div class="row record">
      <div class="col-sm-3 col-xs-3">
        <a href="{link Inzerat:default, $id}" class="thumbnail " >
          
        </a>
        <h5>Cena: {$row['price']|number:0:', ' } Kč</h5>
        <h6>Vloženo: {$row['created']|date:'%d.%m.%Y'}</h6>
        <h6>Lokalita: {$row['area_level_1']}</h6>
        <h6>{$row['username']}</h6>
      </div>
      <div class="col-sm-9">
        <a href="{link Inzerat:default, $id}"><h3>{$row['title']}</h3></a>
        <p>{$row['text']|truncate:350}</p>
        <p class="hidden-xs">
          <a href="mailto:{$row['email']}?subject=Dotaz ze sportovnidily.cz na {$row['title']}&body=Dotaz z portálu sportovnidily.cz" >
            {if $row['favorite'] > 0}
              <a n:href="Listing:favorite"><i class="fa fa-star"></i> Uloženo v oblíbených</a>
            {else}
              <a href="{ $id}" class="addFavorite"><i class="fa fa-star"></i> Přidat do oblíbených</a>
            {/if}
            {if $row['seen']}
              Zobrazeno: {$row['seen']|date:'%d.%m.%Y'}
            {/if}
          </p>
        </div>
      </div>
    <hr />
  {/foreach}
{else}
  <p class="text-center">Je nám líto, ale nenašli jsme žádný inzerát vyhovující vašim kritériím.</p>
  <p class="text-center" style="..."><i class="fa fa-user-times"></i></p>
{/if}
```

Obrázek 11: Latte cyklus pro výpis inzerátů (vlastní tvorba)

4.6 Implementace front-end framework Bootstrap

Front-end inzertního portálu byl postaven na responzivním frameworku Bootstrap. Výběr vhodného frameworku je založen na základě předchozí analýzy uvedené v kapitole Porovnání front-end frameworku. Největšími klady pro volbu Bootstrap je jeho rozšíření napříč internetem a s tím souvisejí množství různých hotových doplňků, které stačí jen použít, například validace formulářů na straně front-endu je velice elegantně vyřešena implementací javascriptového pluginu, který přejímá validační pravidla přímo z presenteru v Nette, tedy ze strany serveru.

Inzertní portál je tedy kompletně responzivní a přizpůsobuje se přesně podle velikosti displeje různých zařízení, tak aby byl co nejlépe použitelný. Příkladem může být formulář pro vkládání inzerátu viz. obrázek níže.



Obrázek 12: Srovnání webové šablony na mobilu a notebooku (vlastní tvorba)

Titulní stránka portálu je stylována s ohledem na návrh wireframů v kapitole 8.3. Cílem titulní strany je zaujmout zákazníka a směřovat ho k vyhledávání inzerátů za pomoci formuláře umístěného v levé části obrazovky přímo pod hlavním menu. Napravo od formuláře pro vyhledávání jsou umístěny náhledy třech nejnovějších inzerátů, které mají za úkol plnit hlavní stranu, stále čerstvým obsahem. Pod náhledy inzerátů jsou umístěny tři nejhledanější kombinace různých parametrů, na základě, kterých uživatelé vyhledávají inzeráty. Pokud je uživatel přihlášen, pak se mu naopak zobrazují poslední 3 kombinace parametrů na základě kterých naposledy vyhledával.

The screenshot shows the homepage of 'SPORTOVNÍ DÍLY'. At the top left is the logo with a wrench and screwdriver. The navigation bar includes links like 'O nás', 'Jak to funguje?', 'Jak přidat inzerát?', 'Obchodní podmínky', 'Kontakt', 'Registrovat', and 'Přihlásit'. A '+ Přidat inzerát' button is in the top right. The search filter on the left has sections for 'Značka', 'Model', 'Stav dílu', and 'Cena od/do'. The main content area shows 'Právě přidáno...' with three items: 'Volant RRS...' (1000 Kč, Brno), 'Turbo Subaru...' (5000 Kč, Praha), and 'ATS' (4 Kč, Praha). A login form is overlaid on the right with fields for 'Email', 'Heslo', and a 'Přihlásit' button. The 'Oblíbené hledaní' section at the bottom lists three popular searches with category and price range details, each with a 'Zobrazit výsledky' link.

Obrázek 13: Screen úvodní stránky portálu (vlastní tvorba)

Po zvolení kombinace parametrů pro vyhledávání inzerátu je uživatel přesměrován na stránku se seznamem inzerátů, kde jsou uvedeny další možnosti filtrování a řazení výsledku vyhledávání. Zákazník může přímo z výsledků kontaktovat prodejce nebo pokud je přihlášen, tak uložit inzerát do oblíbených.

The screenshot shows the 'SPORTOVNÍ DÍLY' website interface. At the top, there is a navigation bar with links: 'O nás', 'Jak to funguje?', 'Jak přidat inzerát?', 'Obchodní podmínky', 'Kontakt', 'Moje menu', and 'Odhlásit se'. A '+ Přidat inzerát' button is located in the top right corner. Below the navigation bar, there are two search filters: 'Hledat dle značky' and 'Hledat dle dílu'. The 'Hledat dle značky' filter includes dropdown menus for 'Značka', 'Model', 'Stav dílu', and 'Cena', along with a 'Vyhledat' button. The 'Hledat dle dílu' filter includes dropdown menus for 'Typ dílu', 'Upřesnění dílu', 'Stav dílu', and 'Cena', also with a 'Vyhledat' button. The search results section shows 'Nalezeno inzerátů: 7' and a 'Seřadit podle:' dropdown set to 'Nejnově'. Two results are visible: 'Volant RRS VELOCE SEMIŠ' (a steering wheel) and 'Turbo Subaru WRX STI 2014' (a turbocharger). Each result includes a price, date listed, location, and seller name, along with 'Kontaktovat' and 'Přidat do oblíbených' buttons. A third result, 'ATS', is partially visible at the bottom.

Obrázek 14: Screen stránky s výsledky vyhledávání (vlastní tvorba)

Na detailu inzerátu je možná zobrazovat veškeré fotografie inzerátu v modálním okně, které je realizované za pomoci Fancyboxu, dále hodnotí uživatele, také pomocí modálního okna, ve kterém vyplní své hodnocení včetně komentáře a uloží na server za pomoci Ajax technologie. Dále jsou zde k dispozici ikony pro sdílení inzerátu na sociálních sítích, možnost kontaktovat prodejce nebo uložit inzerát do oblíbených (pouze pro přihlášené uživatele).



O nás

Jak to funguje?

Jak přidat inzerát?

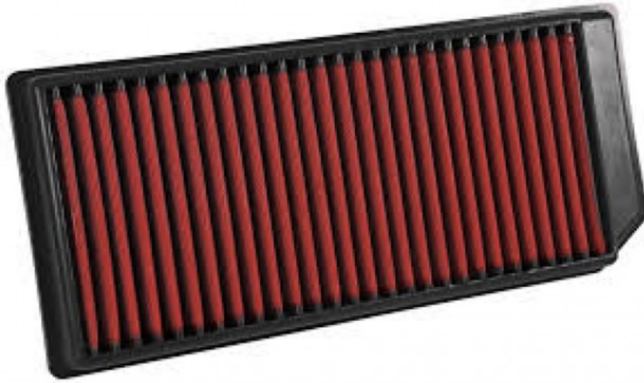
Obchodní podmínky

Kontakt

Registrovat ▾

Přihlásit ▾

Vzduchový filtr



Cena: 200 Kč

Soukromý prodejce: Tomáš Dosoudil

Zobrazit inzeráty prodejce



Vloženo: 28.02.2018 11:33

Lokalita: Hlavní město Praha

Telefon: 603521455

Kontaktovat prodejce

Uložit do oblíbených



Parametry dílu

Značka: KN

Vhodné pro: Abarth

Typ dílu: Motory a motorové díly Vzduchové filtry

Stav dílu: Použitý

Popis

Vysoký průtok vzduchu a vynikající filtrace Zvyšují výkon a zrychlení vašeho vozu. Lze je vyčistit a znovu použít V závislosti na podmínkách provozu vydrží filtr až 80,000 km (50,000 mil) než je potřeba jej vyčistit Funguje s originální elektronikou výrobce vozidla Šetrný k životnímu prostředí Konstruován tak, aby vydržel po dobu životnosti vašeho auta nebo motocyklu Originální vysoce propustný vzduchový filtr K&N se zabývá vývojem, výrobou a neustálým zlepšováním vysoce propustných vzduchových filtrů již od roku 1969. Jsme společnost s celosvětovou působností, která vyvinula a vyrobila více než 2400 různých vysoce propustných vzduchových filtrů, určených pro osobní automobily, motocykly a čtyřkolky. V našem portofiliu se nejen nachází více položek než u kterékoliv jiné firmy vyrábějící obdobné produkty, ale jsme tvůrci samotné kategorie vysoce propustných filtrů a související technologie. Ochrana motoru a filtrace K&N vzduchové filtry jsou navrženy tak, aby dosahovaly vysoké, prakticky neomezené, propustnosti vzduchu a zároveň si zachovávaly filtrační schopnosti na úrovni, která zaručuje dlouhou životnost motoru. Naše motorové výměnné vzduchové filtry obvykle zvyšují výkon o 1-4 koňské síly. Tyto vysoce propustné bavlněné filtry je možné prát a vícenásobně použít a jsou navrženy tak, aby vydržely po celou dobu životnosti motoru. K&N vzduchové filtry se skládají ze čtyř až šesti vrstev speciální bavlny, která je zakomponována mezi dvě hliníkové sítky. Tento materiál je následně skládán a napuštěn olejem, aby se zvýšily jeho filtrační schopnosti. Výsledkem je vzduchový filtr, který propouští výrazně více vzduchu do motoru, je pratelný, lze jej vícenásobně použít a chrání motor po celou dobu životnosti vašeho vozidla.


Podobné inzeráty



2 000 Kč
Pedály



400 Kč
podložky 5x120



4 Kč
ATS



250 Kč
BMW budíky dekor

Obrázek 15: Screen detailu inzerátu (vlastní tvorba)

5 Závěr

Tato práce se zabývá tvorbou webových aplikací v PHP od zrodu myšlenky, přes analýzu dostupných technologií až po samotnou realizaci webové aplikace s využitím vybraných technologií. Teoretická část nejprve shrnula současné principy a metodiky tvorby webových aplikací, dále přímo navazuje na porovnání nejoblíbenější PHP frameworků, vybraných podle průzkumu webových stránek sitepoint.com, kterými jsou Laravel, Symfony a Nette. V další kapitole teoretické části, jsou analyzovány technologie pro tvorbu front-endové části webové aplikace a porovnání třech nejrozšířenější front-end frameworků, kterými jsou Bootstrap, Foundation a Semantic UI. Dále jsou v teoretické části práce porovnávány SŘBD podle stanovených kritérií, které jsou kompatibilní s uvedenými PHP frameworky. Konkrétně byly porovnávány tři databáze (MySQL, PostgreSQL, SQLite), které splňují podmínku kompatibility se všemi třemi PHP frameworky a jsou šířeny pod svobodnou licenci.

Praktická část se zabývá tvorbou inzertního portálu od základních návrhů funkcionalit ve formě use case scénářů, user stories a use case diagramů. Na základě use case byly navrženy wireframy, tak aby aplikace nabídla co nejlepší zážitek z používání. Wireframy byly v další fázi vývoje nahrazeny funkčními šablonami postavenými na front-end frameworku Bootstrap, který zajišťuje uživatelsky přívětivé grafické rozhraní aplikace nejen na desktopu, ale i na mobilních zařízeních. Ze zmíněných wireframů a use case byla navržena relační MySQL databáze za pomoci nástroje MySQL workbench. Jádrem celé aplikace je Nette Framework, který využívá v poslední době jednu z nejrozšířenějších architektur pro tvorbu webových aplikací, čímž je MVC resp. MVP, což je vlastně ekvivalent, jen s rozdílem v pojmenování controlleru jako presenter.

V praktické části práce byla vytvořena funkční webová aplikace, postavená na technologiích, které byly analyzovány v teoretické části. Aplikace byla i testována za pomoci kolegů, kteří mi reportovali problémy a já jsem na základě toho mohl odladit aplikaci až do fáze vhodné k produkčnímu provozu. Portál v tomto stavu není zcela konečnou verzí a je zde stále spousta prostoru pro rozšíření a zlepšení různých funkcionalit jako je třeba přihlášení skrze Facebook API nebo různé možnosti zvyrazňování inzerátů.

6 Použitá literatura:

- (1) BÖHMER, M. Návrhové vzory v PHP. 2012. ISBN 978-80-251-3338-5.
- (2) MVC Architecture. TutorialsTeacher.com [online]. [cit. 2017-12-11]. Dostupné z: <http://www.tutorialsteacher.com/mvc/mvc-architecture>
- (3) Usage of server-side programming languages for websites. W3 Techs [online]. [cit. 2017-11-11]. Dostupné z: https://w3techs.com/technologies/overview/programming_language/all
- (4) WEST, M. HTML5 Foundation. 2013. ISBN 978-1-118-43269-3.
- (5) MVC architecture: The theory behind Model View Controller. MDN Web docs [online]. 18.12.2017 [cit. 2017-12-18]. Dostupné z: https://developer.mozilla.org/en-US/Apps/Fundamentals/Modern_web_app_architecture/MVC_architecture
- (6) SKVORC, Bruno. The Best PHP Framework for 2015: SitePoint Survey Results. SitePoint [online]. 30.4.2015 [cit. 2017-11-11]. Dostupné z: <https://www.sitepoint.com/best-php-framework-2015-sitepoint-survey-results/>
- (7) OTWELL, Taylor. Laravel documentation. Laravel [online]. [cit. 2017-11-21]. Dostupné z: <https://laravel.com/docs/5.6>
- (8) GRUDL, David. Nette framework. Nette [online]. [cit. 2017-11-30]. Dostupné z: <https://nette.org/cs/>
- (9) Symfony. Symfony [online]. [cit. 2017-12-17]. Dostupné z: <https://symfony.com/what-is-symfony>
- (10) Appropriate Uses For SQLite. SQLite [online]. [cit. 2018-01-03]. Dostupné z: <https://www.sqlite.org/whentouse.html>
- (11) The 5 Most Popular Front-end Frameworks Compared. SitePoint [online]. [cit. 2018-01-13]. Dostupné z: <https://www.sitepoint.com/most-popular-frontend-frameworks-compared/>
- (12) CASTRO, E. -- HYSLOP, B. HTML5 a CSS3 : názorný průvodce tvorbou WWW stránek. Brno: Computer Press, 2012. ISBN 978-80-251-3733-8.
- (13) SHARKIE, Craig a Andrew FISHER. Responzivní webdesign: okamžitě. Brno: Computer Press, 2015. ISBN 978-80-251-4384-1.
- (14) Responsive web design. In: QuintaGroup [online]. [cit. 2018-01-23]. Dostupné z: <http://quintagroup.com/services/web-design/responsive-website-design.png>
- (15) Introduction to the DOM. MDN Web docs [online]. 21.1.2018 [cit. 2018-02-01]. Dostupné z: https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model/Introduction

- (16) VÁVRŮ, J. jQuery Mobile. Brno: Computer Press, 2013. ISBN 978-80-251-3811-3.
- (17) LACKO, L. Ajax hotová řešení. Brno: Computer press, a.s. 2008. ISBN 978-80-251-2108-5.
- (18) System Properties Comparison MySQL vs. PostgreSQL vs. SQLite. DB engines [online]. [cit. 2018-02-14]. Dostupné z: <https://db-engines.com/en/system/MySQL%3BPostgreSQL%3BSQLite>
- (19) Bootstrap documentation [online]. [cit. 2018-02-24]. Dostupné z: <http://getbootstrap.com/>
- (20) Zurb Foundation doc: Documentation [online]. Zurb [cit. 2018-02-25]. Dostupné z: <https://foundation.zurb.com/sites/docs/>
- (21) Semantic UI: User Interface is the language of the web [online]. [cit. 2018-02-27]. Dostupné z: <https://semantic-ui.com/>
- (22) DELISLE, M. phpMyAdmin : efektivní správa MySQL. Brno: Zoner Press, 2004. ISBN 80-86815-09-9.
- (23) VOSTROVSKÝ, V. Vytváření databází v Oracle. Praha: Česká zemědělská univerzita v Praze, 2009. ISBN 978-80-213-1191-6.
- (24) DB-Engines: Knowledge Base of Relational and NoSQL Database Management Systems [online]. [cit. 2018-03-04]. Dostupné z: <https://db-engines.com/en/>
- (25) PostgreSQL [online]. [cit. 2018-03-04]. Dostupné z: <https://www.postgresql.org/>
- (26) KRUG, Steve. Don't Make Me Think!. Second Edition. Berkeley, CA: New Riders, 2006. ISBN 9786612430572.
- (27) VRANA, Ivan. Projektování informačních systémů s UML. V Praze: Česká zemědělská univerzita, Provozně ekonomická fakulta, 2008. ISBN 978-80-213-1817-5.

7 Seznam obrázků

Obrázek 1: Princip Model - View – Controller [2].....	6
Obrázek 2: Graf oblíbenosti PHP frameworků [6]	8
Obrázek 3: Princip responzivního designu webu [14]	21
Obrázek 4: Use case diagram (vlastní tvorba).....	53
Obrázek 5: Wireframe homepage (vlastní tvorba).....	54
Obrázek 6: Wireframe výsledky vyhledávání (vlastní tvorba).....	55
Obrázek 7: Wireframe detail inzerátu (vlastní tvorba).....	56
Obrázek 8: Návrh MySQL databáze	58
Obrázek 9: Adresářová struktura aplikace (vlastní tvorba).....	59
Obrázek 10: Komponenta pro registraci uživatele (vlastní tvorba).....	60
Obrázek 11: Latte cyklus pro výpis inzerátů (vlastní tvorba)	61
Obrázek 12: Srovnání webové šablony na mobilu a notebooku (vlastní tvorba).....	62
Obrázek 13: Screen úvodní stránky portálu (vlastní tvorba).....	63
Obrázek 14: Screen stránky s výsledky vyhledávání (vlastní tvorba)	64
Obrázek 15: Screen detailu inzerátu (vlastní tvorba).....	65

8 Seznam tabulek

Tabulka 1: Porovnání kvalitativních hodnot PHP frameworků.....	19
Tabulka 2: Porovnání kvantitativních hodnot PHP frameworků.....	19
Tabulka 3: Porovnání front-end frameworků	31
Tabulka 4: Porovnání relačních databází	43