

Katedra informatiky
Přírodovědecká fakulta
Univerzita Palackého v Olomouci

BAKALÁŘSKÁ PRÁCE

Internetový obchod



2015

Vedoucí práce: RNDr. Arnošt Ve-
čerka

Ondřej Žádník

Studijní obor: Aplikovaná informatika,
prezenční forma

Bibliografické údaje

Autor: Ondřej Žádník
Název práce: Internetový obchod
Typ práce: bakalářská práce
Pracoviště: Katedra informatiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci
Rok obhajoby: 2015
Studijní obor: Aplikovaná informatika, prezenční forma
Vedoucí práce: RNDr. Arnošt Večerka
Počet stran: 43
Přílohy: 1 CD/DVD
Jazyk práce: český

Bibliographic info

Author: Ondřej Žádník
Title: Ecommerce system
Thesis type: bachelor thesis
Department: Department of Computer Science, Faculty of Science, Palacký University Olomouc
Year of defense: 2015
Study field: Applied Computer Science, full-time form
Supervisor: RNDr. Arnošt Večerka
Page count: 43
Supplements: 1 CD/DVD
Thesis language: Czech

Anotace

Tato práce se zabývá tématem internetového obchodu. Návrh obsahuje rozdělení na administrační a zákaznickou část. Datové úložiště je realizováno pomocí databázového systému PostgreSQL. Aplikace samotná je navržena v moderním frameworku Ruby on Rails využívající jazyka Ruby. Vzhled je navržen responzivně ve frameworku bootstrap 3.

Synopsis

This thesis deals with model of ecommerce system. System is splitted into two big sections. Administration section and customer section. Data storage is implemented using PostgreSQL database system. Application is designed in Ruby on Rails using Ruby language. Appearance is designed in responsive framework Bootstrap 3.

Klíčová slova: internetový obchod; elektronický obchod

Keywords: ecommerce; eshop

Rád bych poděkoval RNDr. Arnoštu Večerkovi za cenné rady při vývoji a vedení.
Rád bych také poděkoval rodině za trpělivost.

Místopřísežně prohlašuji, že jsem celou práci včetně příloh vypracoval/a samostatně a za použití pouze zdrojů citovaných v textu práce a uvedených v seznamu literatury.

datum odevzdání práce

podpis autora

Obsah

1	Zadání bakalářské práce	9
2	Úvod	10
2.1	Moderní webové technologie	10
2.1.1	HTML 5	10
2.1.1.1	HTML5 značky	11
2.1.1.2	HTML5 funkce	11
2.1.2	CSS 3	12
2.1.2.1	Border radius	12
2.1.3	JavaScript	12
2.1.4	AJAX	13
2.2	Programovací jazyky používané při práci s webem	13
2.2.1	PHP	13
2.2.1.1	PHP Frameworky	13
2.2.2	Python	14
2.2.2.1	Django framework	14
2.2.3	Ruby	14
2.2.3.1	Ruby on Rails	14
2.2.4	ASP.NET	14
2.3	Modely internetového podnikání	15
2.3.1	Internetové obchody	15
2.3.2	Elektronická aukce	15
2.4	Druhy obchodů	15
2.4.1	Open source řešení	15
2.4.1.1	Řešení zdarma	15
2.4.2	Řešení na míru	16
2.4.3	Měsíční pronájmy	16
2.5	Moje vize internetového obchodu	16
2.5.1	Produkční část	17
2.5.2	Administrační část	17
3	Použité technologie	17
3.1	Architektura MVC - MVP	17
3.1.1	Modely (models)	17
3.1.2	Pohledy (views)	18
3.1.3	Kontrollery (controllers)	18
3.2	Databáze	18
3.2.1	doplňěk HStore	18
3.3	Framework	18
3.3.1	Rails principy:	19
3.3.1.1	Součásti Rails	19
3.3.2	REST	20
3.3.3	MailChimp	21

3.3.4	Gemy	22
4	Uživatelská dokumentace	23
4.1	Zákaznická část	23
4.1.1	Header	23
4.1.2	Levé menu kategorií	24
4.1.3	Body	24
4.1.3.1	Registrace	24
4.1.3.2	Přihlášení	24
4.1.3.3	Kategorie	24
4.1.3.4	Produkt	24
4.1.3.5	Košík	25
4.1.3.6	Objednávka	25
4.1.4	Footer	25
4.1.4.1	O nás	26
4.1.4.2	Informace	26
4.1.4.3	Kontaktujte nás	26
4.2	Administrační část	26
4.2.1	Úvodní přihlašovací strana	26
4.2.2	Záhlaví dokumentu	26
4.2.3	Levé menu	26
4.2.3.1	Položky menu	26
4.2.4	Tělo dokumentu	27
4.2.4.1	Objednávky	27
4.2.4.2	Produkty	27
4.2.4.3	Kategorie	28
4.2.4.4	Výrobci	28
4.2.4.5	Uživatelé	28
4.2.4.6	Uživatelé	28
4.2.4.7	Způsob platby	28
4.2.4.8	Způsob dopravy	28
4.2.4.9	Daně	29
4.2.4.10	Atributy	29
4.2.4.11	Nastavení	29
5	Programátorská dokumentace	30
5.1	Databázová vrstva	30
5.1.1	Produkt	31
5.1.2	Atributy produktu	32
5.1.3	Kategorie	33
5.1.4	Zařazení produktů do kategorií	33
5.1.5	Košík	34
5.1.6	Jednotlivé položky v košíku	34
5.1.7	Objednávky	35
5.1.8	Doprava a platba	35

5.2	Aplikační vrstva	36
5.2.1	OrdersController	36
5.2.2	ProductsController	37
5.2.3	CategoriesController	37
5.2.4	ManufacturersController	37
5.2.5	UsersController	37
5.2.6	CustomersController	37
5.2.7	PaymentsController	37
5.2.8	ShippingsController	37
5.2.9	VatsController	37
5.2.10	PropertiesController	38
5.2.11	SettingsController	38
5.3	Pohledová vrstva	38
5.3.1	Šablony	38
5.3.1.1	Formulář (_form.html.haml)	38
5.3.1.2	Nová (new.html.haml)	38
5.3.1.3	Úprava (edit.html.haml)	38
5.3.1.4	Přehled (index.html.haml)	39
5.3.1.5	Zobrazení (show.html.haml)	39
5.3.1.6	Hlavní kostra (application.html.haml)	39
	Závěr	40
	Conclusions	41
	Reference	42
	A Obsah příloženého CD/DVD	43

Seznam tabulek

1	Tabulka produktů(products)	31
2	Tabulka atributů(properties)	32
3	Tabulka hodnot hodnot atributů(property_values)	32
4	Tabulka variant(variants)	32
5	Tabulka kategorií(categories)	33
6	Tabulka relací mezi produkty a kategoriemi(categorizations) . . .	33
7	Tabulka košíku(carts)	34
8	Tabulka položek v košíku(cart_items)	34
9	Tabulka objednávek (orders)	35
10	Tabulka způsobů dopravy (shippings)	35
11	Tabulka platebních metod (payments)	36
12	Tabulka relací mezi platbami a dopravou (payments_h)	36

1 Zadání bakalářské práce

Cílem práce je sestavit systém pro realizaci internetového obchodu. Systém by měl mít vlastnosti a funkce: Možnost rozdělení zboží do kategorií, uložení informací o zboží, údaje o stavu zboží na skladě. Poskytování informací o zboží prodávaném v obchodě pro porovnávače zboží (internetové stránky, na kterých je srovnáváno zboží dostupné v různých internetových obchodech). V internetovém obchodu bude možná registrace zákazníků, kterým budou posílány informace o akcích, novinkách v nabídce zboží a další. Pro správce bude systém poskytovat statistiky o prodeji a objednávkách.

2 Úvod

V dnešní moderní době je pro mladé podnikatele nepředstavitelné snažit se prorazit na trh bez pomoci internetu. Stal se jeho součástí. Proto bych rád poskytl tuto bakalářskou práci jako prostředek pro mladé či začínající podnikatele, kteří nemají možnost pořídit si drahý systém na míru a nechtějí používat robustní a složitá řešení.

2.1 Moderní webové technologie

Dříve se používaly webové prohlížeče k prohlížení webového obsahu, čtení a procházení elektronické pošty. Pokrok ale udělal z webu opravdu silný nástroj. Dnes tu máme složité systémy pro správu obsahu, systémy pro internetové obchody, osobní weby a portfolia plná efektů. Není tomu ani tak dávno, co webové aplikace pronikly i do mobilních zařízení.

V dnešní době každý někam spěchá, proto každá úspora času je pro člověka přínosem. Trendem jsou tedy weby optimalizované pro naše chytré telefony či tablety. Lidé ocení, když si mohou zkontrolovat objednávky na svém přenosném zařízení.

Možnosti těchto aplikací vycházejí z použitých webových technologií. Tyto technologie se vyvíjejí a rozrůstají obrovským tempem. Dnes máme tedy k dispozici širokou škálu jazyků pro práci s webem. Kvůli jejich nekontrolovatelnému vývoji bylo potřeba začít dodržovat určité standardní postupy. Proto vzniklo mezinárodní konsorcium W3C, jež společně s odbornou veřejností začínají vyvíjet webové standardy. Každý web by měl tyto standardy dodržovat, zajistí to jeho přístupnost z širokého spektra prohlížečů.

Uvedu zde tedy základ, ze kterého by měl programátor vybírat, pokud se chce v dnešní době začít věnovat tvorbě webů či webových aplikací.

2.1.1 HTML 5

HTML5 je značkový jazyk, ale standardem W3C se prozatím nestal. Jeho finální specifikace byla vydána 28. října 2014. Specifikace je rozdělena do několika částí. Některé z nich jsou používány již nyní. Tyto části implementují tvůrci prohlížečů. I tak se zdá, že právě HTML 5 prorazí jako hlavní prostředek pro kvalitní a moderní web. Neexistuje žádné kritérium, co by mělo HTML 5 pokrývat ve webových prohlížečích.

Používání starších HTML 4.01, XHTML 1.0 je samozřejmě stále možné a mnohými kritiky HTML5 doporučované. Musíme se ale zamyslet nad tím, že na webu máme pouze jednu verzi HTML. Proč se tedy okrádat o funkce, které naše

moderní prohlížeče dávno umí. Již nyní můžete využít spoustu funkcí a nových značek. Uvedu zde některé základní z nich:

2.1.1.1 HTML5 značky

- **<header>** - používá se pro záhlaví stránky, většinou obsahuje menu a logo popř. nadpis
- **<main>** - používá se pro hlavní obsah
- **<footer>** - používá se pro zápatí stránky
- **<section>** - používá se pro části stránek popř. kapitol
- **<article>** - používá se pro menší části stránek jako příspěvky atd..
- **<aside>** - používá se často jako doplněk pro boční či vedlejší navigaci
- **<nav>** - používá se pro hlavní navigaci stránek
- **<figure>** - používá se pro obrázky či videa popř. jiné data
- **<figcaption>** - používá se jako popis pro figure
- **<video>** - používá se pro vkládání videa
- **<audio>** - používá se pro vkládání audia
- **<canvas>** - používá se pro vkládání vektorové grafiky, je možné do něj kreslit pomocí skriptovacích jazyků. Vhodné pro grafy.

2.1.1.2 HTML5 funkce

- **<mark>** - zvýraznění textu
- **<time>** - zobrazuje datum a čas
- **<progress>** - zobrazuje stav dokončení úlohy u stahování
- **<bdi>** - používá se pro text psaný v jiném směru
- **<wbr>** - používá se pro zalomení textu
- **<details>** - používá se pro informace zobrazené po najetí myši
- **<output>** - používá se pro výstup jako jsou výsledky aritmetických operací

2.1.2 CSS 3

CSS 3 neboli třetí verze kaskádových stylů je tu již od roku 2005. Na rozdíl od HTML5 je vyvíjeno konsorciem W3C. Stále není dokončeno, ale většina webových prohlížečů nejenže podporuje tento standard, ale také se snaží o co nejrychlejší implementaci nových funkcí. Představuje neodmyslitelnou součást moderního webu. Stará se nám o styl našich stránek popř. aplikací. Dle W3C je doporučena jeho zastaralejší druhá verze. Nové vlastnosti, které se nám dosud objevily jsou např. **zaoblené rohy, stíny, 2D a 3D transformace a spoustu dalších...**

Dřívější hit jednoduchých webových stránek s minimem velkých obrázků či datově náročných prvků nám pomalu mizí. S rostoucími rychlostmi internetu dnes navštěvujeme stránky plné efektů a fotek ve vysokém rozlišení. Příchod chytrých telefonů nám poskytl nové možnosti, jak vylepšovat stávající weby. V CSS3 se objevily věci jako Flexbox a další vlastnosti vhodné pro responzivitu. Responzivní web je stránka, která se nám vždy přizpůsobí velikosti displeje či jeho orientaci.

CSS3 používá pro každé jádro prohlížeče specifické prefixy. Do stylů je tak třeba pro jednu vlastnost zadat přinejmenším třikrát až čtyřikrát stejný styl, ale s jiným prefixem. Ukázka pro zaoblené rohy o velikosti 5 pixelů:

2.1.2.1 Border radius

- **border-radius: 5px** - Defaultní nastavení pro prohlížeče
- **-ms-border-radius: 5px** - Nastavení pro prohlížeče Internet Explorer
- **-o-border-radius: 5px** - Nastavení pro starší verze prohlížeče Opera
- **-webkit-border-radius: 5px** - Nastavení pro prohlížeče s jádrem Webkit(Google Chrome, Apple Safari,)
- **-moz-border-radius: 5px** - Nastavení pro prohlížeče s jádrem Gecko(Mozilla Firefox)

2.1.3 JavaScript

JavaScript je objektově orientovaný a multiplatformní skriptovací jazyk. Můžeme ho zapisovat jak do externích souborů, které poté přidáme do hlavičky dokumentu, nebo jej zapisujeme přímo do HTML kódu. Pracuje na straně klienta, to znamená, že je vykonáván v prohlížeči. Často se používá pro různé efekty či události. Nesmí být zaměňován s jazykem Java. Do dnešní doby vznikla velká spousta knihoven z něj odvozených. Mezi nejznámější patří JQuery či Prototype. Vznikla také obrovská spousta doplňků, jako jsou galerie, portfolio, grafy.

Podpora javascriptu je dnes ve všech dostupných prohlížečích. Ovšem je možno ji explicitně vypnout. To však může způsobit velké problémy vývojářům,

kteří implementují některé důležité funkce svých webů právě jen v javascriptu. Web by tedy měl být připraven, aby byl čitelný a interagující jak s javascriptem, tak i bez něj. Častokrát byl označován za zdroj potencionálních hrozeb kvůli jeho možnostem, jako je generování škodlivého kódu.

2.1.4 AJAX

Asynchronní javascript a XML, to je technologie jménem AJAX. Je to technologie poměrně mladá a dokáže měnit námi určenou část stránek bez nutnosti kompletního obnovení. Zapotřebí je použití novějších verzí webových prohlížečů. Velkou výhodou je tady načtení jen požadovaného obsahu. To nám pomůže snížit snížit zátěž na náš server.

2.2 Programovací jazyky používané při práci s webem

2.2.1 PHP

Dynamický skriptovací jazyk. Skripty jsou prováděny na straně serveru. K uživateli se tak dostane jen výsledek. Dynamicky typovaný jazyk. Datové typy nejsou vázané na proměnnou, ale na hodnotu proměnné. Je nezávislý na použité platformě. Jde tedy libovolně přenášet mezi operačními systémy. PHP si časem vypracovalo obrovskou popularitu a dnes ovládá zhruba 80% trhu. Své oblíbenosti dosáhl především díky své jednoduchosti a velké základně knihoven. Taktéž umí přistupovat k širokému spektru databázových systémů, jako jsou MySQL, PostgreSQL, MSSQL, SQLite a další. Podporuje široké spektrum internetových protokolů, jako jsou HTTP, HTTPS, FTP, IMAP a další. Velice často se používá v kombinaci s webovým serverem Apache. V PHP jsou například napsané i ty největší webové stránky, jako je Wikipedia či některé části Facebooku.

Poskytuje taky širokou škálu frameworků.

2.2.1.1 PHP Frameworky

- **Zend** - open source framework, který staví na modularitě a umožňuje vývojářům použít jen ty části, které potřebují.
- **CodeIgniter** - další open source framework, založený na MVC architektuře
- **Nette** - nejznámější Český php framework, založený na MVP architektuře
- **Symfony** - open source framework založený a MVC architektuře, inspirovaný frameworky jako ruby on rails či django
- **Laravel** - open source framework založený a MVC architektuře, jeden z nejrozšířenějších ve světě

2.2.2 Python

Další z řady dynamických objektově orientovaných skriptovacích jazyků. Je vyvíjen jako open source a nabízí instalační balíčky pro všechny operační systémy. Python je zajímavý tím, že se skládá z více paradigmat. Můžeme používat jak objektově orientované paradigma, tak i procedurální či funkcionální. Ve srovnání s jinými jazyky je jeho syntaxe krátká a výstižná. Proto je velice oblíbený z hlediska rychlosti psaní aplikací. Často je Python používán ve spojení s ostatními programovacími jazyky. Aplikace v něm napsané se vyznačují velkou rychlostí. Některé jeho části jsou implementovány v jazyce C.

2.2.2.1 Django framework Django je open source framework postavený na MVC architektuře. Nabízí zajímavou možnost vytvoření administrativní části, ke které poskytuje i grafické prostředí. Dokáže pracovat s následujícími databázemi PostgreSQL, MySQL, SQLite. Je zde také k dispozici samostatný server pro testování.

2.2.3 Ruby

Interpretovaný skriptovací jazyk, který vyniká svou velice jednoduchou syntaxí, ale zároveň velice výkonný. Ruby má zajímavou vlastnost a to, že vše v něm je objekt, tudíž se jedná o plně objektově orientovaný jazyk. Jako ostatní moderní jazyky je multiplatformní a dynamicky typovaný. V posledních letech jeho obliba výrazně stoupla právě kvůli jeho webovému frameworku Ruby on Rails.

2.2.3.1 Ruby on Rails Používá návrhový vzor MVC a také velice zajímavý princip, kdy programátor konfiguruje jen ty části, které se liší od běžného nastavení. To znamená, že vytvořený model se automaticky snaží hledat tabulku stejného názvu v databázi. Mapuje data z databáze a převádí je na objekty.

Používá se k tomu návrhový vzor ActiveRecord, který z řádků udělá instance objektu a ze sloupců jejich atributy. Velice silnou stránkou frameworku jsou databázové migrace, kdy můžete lehce převést celou strukturu všech tabulek mezi různými databázemi. Podporované jsou MySQL, PostgreSQL, SQLite. Možnost využít velkou základnu takzvaných gemů, což jsou speciálně vytvořené knihovny s doplňky.

2.2.4 ASP.NET

ASP.NET není programovacím jazykem jako ostatní, ale je součástí rozsáhlého frameworku .NET, který používá velkou škálu programovacích jazyků. Programátor má tedy možnost vytvářet své projekt v jakémkoliv jazyce podporujícím CLR. Aplikace jsou velice rychlé a po zkompilování se rozdělí do několika DLL souborů. K dispozici jsou knihovny tříd a rozsáhlé komponenty sloužící k ulehčení práce.

2.3 Modely internetového podnikání

V internetovém podnikání se setkáváme s různorodými modely použití. Situace si vyžádala vyčlenit odlišné druhy obchodování. Bylo třeba rozlišit použití obchodu. Zda se jedná o doplněk ke kamennému obchodu, který slouží spíše jen jako katalog zboží. Pokud se ale jednalo o hlavní zdroj podnikání, či dnes již tak populární internetové aukce, bylo potřeba daleko více propracovanějších systémů .

2.3.1 Internetové obchody

Základní stavební kámen každého, kdo se chce zapojit do podnikání na internetu. Nabízí nám způsob, jak prezentovat naše produkty. Bližší popis rozeberu v kapitole internetový obchod v praxi.

2.3.2 Elektronická aukce

Elektronické aukce se staly obrovským hitem. Aukce může zahrnovat jakékoliv produkty a její nasazení je velice efektivní tam, kde se setkáváme s poptávkou menší než nabídkou. Velkým plusem je jednoduchost a rychlost, s jakou mohou uživatelé dražit. Aukce je velice silná zbraň na psychiku kupujícího, kdy stále vidí zvyšující se nabídku a pokouší se vyhrát.

2.4 Druhy obchodů

2.4.1 Open source řešení

První varianta je zdarma a pod volně šiřitelnou licenci. Tyto řešení se vyznačují především univerzálním systémem. Ten nám povolí široké možnosti nastavení a ladění. Většinou jsou tyto systémy velice robustní a rozsáhlé. Snaží se systém přizpůsobit každému druhu zboží či služeb. Výhodou jsou téměř nulové náklady, pokud si jej dokážeme spustit sami. Máme tedy hotové řešení a funkční internetový obchod s minimem práce pro nás. Na výběr je celá řada již hotových šablon přímo pro konkrétní systém.

Bohužel toto řešení má i nevýhody. Vlastní grafické řešení u těchto systémů již není úplnou banalitou. Specifická funkcionalita vyžaduje doprogramování vlastních doplňků od programátora. Při obchodu, kde jdou položky do sta tisíců, se můžeme setkat s výkonnostními problémy.

2.4.1.1 Řešení zdarma

- **OpenCart** - jednoduchý internetový obchod pod licencí GNU LGPL. Nabízí svým uživatelům vložení neomezeného počtu produktů, kategorií a výrobců. Aplikace má přehledné administrační rozhraní a nabízí i českou lokalizaci.
- **PrestaShop** - jeden z nejpoužívanějších elektronických obchodů. Nabízí mnoho zásuvných modulů. Jeho nová verze obsahuje responzivní šablonu, která zajistí optimalizované zobrazení webu i na mobilních zařízeních.
- **Magento** - moderní řešení s množstvím inovačních funkcí. U nás nepříliš známý. Umožňuje správu více obchodů pomocí jedné instalace (tzv. multi-store).

2.4.2 Řešení na míru

Druhou variantou jsou řešení napsaná přímo na míru. Dostaneme systém přesně takový, jaký jsme si vymysleli popř. zadali. To se nám jistě promítne do pořizovací ceny, která u těchto řešení není zrovna nízká. Zde je nevýhoda, pokud časem zjistíme, že bychom rádi přidali novou funkcionalitu. Většinou musíme požádat tvůrce o dodatečné doprogramování, což představuje další finanční výlohy.

2.4.3 Měsíční pronájmy

Třetí variantou jsou řešení, která jsou již hotová a roky osvědčená. Neplatíme ovšem pořizovací cenu, nýbrž měsíční pronájem. Některé firmy poskytují i službu vytvoření grafiky na míru k pronájmu. Častokrát se setkáme s různými verzemi odstupňovanými dle možností či rozsahu poskytovaných služeb. Tato alternativa zažívá v poslední době velký růst a je velmi populární.

2.5 Moje vize internetového obchodu

Při vytváření jsem dbal na to, aby vše bylo pro správce co nejjednodušší. Proto je dbáno na příjemný design a přehlednost. Aplikace poskytuje přehledné rozdělení kategorií. Katalog produktů obsahuje u každého z nich detailní informace o cenách a specifikacích.

Mou vizí bylo vytvoření jednoduchého internetového obchodu i s administrační částí, která slouží ke správě téměř všech položek obchodu. Obchod obsahuje dvě části. Produkční část, kam se dostane zákazník, když navštíví příslušnou webovou stránku. Administrační část, kam se přistupuje přidáním sufixu /admin za naši adresu.

2.5.1 Produkční část

Produkční část nám poskytne rychlý přístup ke kategoriím či produktům. Umožní nám jednoduché přidání do košíku, odkud potom můžeme postupovat k potvrzení a samotné objednávce. Objednání zboží je možné i bez předchozí registrace. Produkční část mi zde posloužila jen pro prezentaci jednoduchého obchodu a nevyužívá celý potenciál, který nabízí část administrační. Záleží tedy na každém programátorovi, jak ji implementuje a rozvrhne.

2.5.2 Administrační část

Administrační část se stará o základní akce přidání, odebrání, editace u příslušných modelů. Dále zde máme přehled o zákaznících a objednávkách. U objednávek přehledně vidíme jak nové, tak ty již přijaté či odeslané. Můžeme zde rozeslat hromadný email o slevách či nových produktech a akcích. Můžeme zde exportovat zboží pro porovnávače zboží jako Zboží.cz. Administrační část je stěžejní část aplikace. Představuje celý propracovaný systém, který nám umožní spravovat obchod.

3 Použité technologie

Technologie využívané v mé aplikaci jsem se snažil volit s ohledem na znovupoužitelnost a přehlednost kódu. Snažil jsem se aplikaci přizpůsobit zařízením s různou velikostí obrazovky či rozlišení.

3.1 Architektura MVC - MVP

Model - View - Controller je softwarový model či architektura, která odděluje 3 základní části programu. Modelová vrstva představuje aplikační logiku či práci nad databází. View, jak již anglický název napovídá, se stará o zobrazování dat. Controller se stará o obsluhu aplikace a je jakýmsi prostředníkem mezi částí Model a View.

3.1.1 Modely (models)

Model reprezentuje informace (data) ve vaší aplikaci a pravidla pro práci s nimi. V případě Rails jsou modely primárně využívány pro interakci s příslušnou tabulkou v databázi a pro ukládání pravidel této interakce. Ve většině případů odpovídá jedna tabulka v databázi jednomu modelu ve vaší aplikaci. Modely obsahují většinu aplikační logiky.

3.1.2 Pohledy (views)

Pohledy, neboli anglicky views, reprezentují uživatelské rozhraní vaší aplikace. V Rails jsou views obvykle HTML soubory s vloženými částmi Ruby kódu, který provádí pouze úkony týkající se prezentace dat. Views mají na starosti poskytování dat webovému prohlížeči nebo jinému nástroji, který zasílá vaši aplikaci požadavky.

3.1.3 Kontrollery (controllers)

Kontrollery fungují jako „lepidlo“ mezi modely a views. V Rails slouží kontrollery k zpracování požadavků, které přichází z webového prohlížeče, získávání dat z modelů a k odesílání těchto dat do views, kde budou zobrazeny.

3.2 Databáze

Jako datová základna mi posloužila technologie PostgreSQL, kterou jsem zvolil hlavně kvůli její rychlosti a jednoduchosti. Je to multiplatformní databáze, která používá jazyk SQL. Lze jej jednoduše implementovat na operační systémy Linux či Windows. Pro zjednodušení a efektivní práci s jazyky jsem zvolil použití doplňku HStore.

3.2.1 doplněk HStore

Implementuje datový typ pro ukládání sad párů klíč/hodnota v rámci jednoho databázové řádku. Klíče a hodnoty jsou uloženy jako jednoduché řetězce.

3.3 Framework

Ruby On Rails je framework pro vývoj webových aplikací napsaný v jazyce Ruby. Je navržen tak, aby usnadnil programování webových aplikací na základě obecných předpokladů vývoje pro web. Umožňuje vám psát méně kódu a zároveň dosáhnout více než v mnoha jiných jazycích a platformách.

Umožňuje pohodlné a rychlé vytváření moderních webových aplikací. Poskytuje abstraktní vrstvu pro práci s databází. Přehlednou implementaci model-view-controller architektury. Zabudovanou podporu pro automatizované testování všech vrstev aplikace. Generátory kódu nebo konzoli pro interaktivní práci s aplikací.

Publikoval jej v roce 2004 dánský programátor David Heinemeier Hansson a dnes ho vylepšují tisíce přispěvatelů a používají ho společnosti jako je Apple, Oakley, The New York Times, Twitter, Electronic Arts, Github nebo Yellow Pages.

Rails jsou od počátku zdrojem inspirace pro frameworky na ostatních platformách.

3.3.1 Rails principy:

- **DRY** - „Don't repeat yourself“ aneb „neopakujte se“ — znamená, že psát stejný kód stále dokola je špatné a dobrý kód je „znovupoužitelný“
- **Konvence má přednost před konfigurací** - znamená, že Rails předpokládají, co asi chcete udělat a jak to chcete udělat, místo aby vás nutily specifikovat každou drobnost v nekonečném množství konfiguračních souborů
- **REST je nejlepší architektonický vzor pro webové aplikace** - znamená, že uspořádat webovou aplikaci jako soubor „zdrojů“ (resources) a standardních HTTP „sloves“ (verbs) je nejsnazší a nejrychlejší způsob, jak modelovat entity a jejich vztahy

3.3.1.1 Součásti Rails Rails nejsou monolitický framework, ale jsou poskládané z mnoha dílčích součástí:

- **Action Pack**
 - **Action Controller** - Komponenta která obstarává všechny kontroly v aplikaci. Stará se o příchozí požadavky, extrahuje z nich parametry a směřuje je na příslušné akce. Stará se o sessions úložiště, vykreslování šablon a přesměrování.
 - **Action Dispatch** - Zpracovává směřování příchozích požadavků a posílá je do aplikace nebo přidružených aplikací.
 - **Action View** - Stará se o pohledy v aplikaci. V základu poskytuje výstup v HTML a XML. Řídí vykreslování šablon, včetně šablon vnořených a parciálních, a obsahuje zabudovanou podporu pro Ajax.
- **Action Mailer** - Podpora pro e-maily. Umožňuje rozesílat základní emaily, e-maily s přílohami. Možnost použití flexibilních šablon. Zároveň se dokáže postarat o příjem emailu a jeho zpracování.
- **Action Model** - Definuje rozhraní mezi službami součástí Action Pack a vrstvami tzv. ORM (Object Relational Mapping, objektově relační mapování). Jednou z takových vrstev je i Active Record.
- **Action Record** - Základ pro modely v Rails aplikaci. Poskytuje mimo jiné nezávislost na SQL dialektu konkrétní databáze, základní operace pro získávání, vytváření, editaci a mazání záznamů (tzv. CRUD, Create Read Update Delete), pokročilé získávání dat z databáze (operace typu JOIN,

GROUP, aj.) a v neposlední řadě i možnost definovat vztahy mezi jednotlivými modely.

- **Action Resource** - Umožňuje propojit modely ve vaší aplikaci s webovou službou typu REST. Mapuje modely na vzdálené webové služby, umožňuje vám s nimi pracovat podobně jako by byly mapovány na databázi.
- **Action Support** - Active Support je rozsáhlý soubor podpůrných funkcí a rozšíření základních tříd Ruby, které kromě kódu samotných Rails může využít i vaše aplikace.
- **Railties** - Railties jsou jádrem kódu Ruby On Rails, který umožňuje vygenerovat novou Rails aplikaci a propojuje jednotlivé frameworky a součásti dohromady.

3.3.2 REST

Zkratka pro Representational State Transfer a je základem tzv. REST architektury. Ta se projevuje těmito způsoby:

vaše aplikace přenáší reprezentace stavu těchto entit mezi jednotlivými součástmi celého systému

- vaši aplikaci tvoří určité entity („zdroje“), které lze jednoznačně identifikovat, typicky prostřednictvím jejich URL
- vaše aplikace přenáší reprezentace stavu těchto entit mezi jednotlivými součástmi celého systému

3.3.3 MailChimp

Aplikace poskytuje pohodlné rozhraní pro rozesílání emailů mezi zákazníky. Ať už různé měsíční newslettery nebo jen informativní emaily o akcích. Do 2 tisíc zákazníků je zcela zdarma. Proto jsem se rozhodl moji aplikaci propojit s MailChimpem. Stačí si vytvořit účet u MailChimpu. Kde si založíme list zákazníků kterým budeme chtít posílat emaily. V jeho nastavení si najdeme unikátní id. Vygenerujeme si unikátní API klíč. Oba údaje potom zadáme v administraci do nastavení. Tím nám bude aplikace sama sbírat emaily a základní informace. Potom už jen stačí spustit mailovou kampaň.

- vaši aplikaci tvoří určité entity („zdroje“), které lze jednoznačně identifikovat, typicky prostřednictvím jejich URL
- vaše aplikace přenáší reprezentace stavu těchto entit mezi jednotlivými součástmi celého systému

3.3.4 Gemy

Rails aplikace používá ke svému běhu takzvané Gemy. Jsou to balíčky které jsou zahrnuty do aplikace při jejím sestavení. Každý gem je většinou zaměřen na nějakou specifickou část. Dokáží programátorovi ulehčit práci a zlepšit redundanci kódu.

- gem **coffee-rails** - přidává podporu CoffeeScriptu a dovoluje používat jeho syntax
- gem **jquery-rails** - přidává podporu JQuery knihoven
- gem **lightbox2-rails** - přidává Lightbox plugin
- gem **turbolinks** - nahrazuje přesměrování v aplikaci za ajaxové přesměrování
- gem **haml__rails** - přidává podporu pro haml syntax, která ulehčí kodérovi práci s html šablonami. Odpadá nutnost uzavírání párových tagů.
- gem **bcrypt__ruby** - přidává podporu pro šifrování hesel
- gem **haml__rails** - přidává plugin font-awesome používaný pro ikony
- gem **bootstrap__rails** - přidává podporu bootstrap frameworku a jeho tříd
- gem **pundit** - přidává třídy pro restrikcí přístupů k našim modelům
- gem **pg** - přidává PostgreSQL adaptér, v základním nastavení Rails používá SQLite
- gem **nested__form** - přidává podporu pro použití vnořených formulářů
- gem **bootstrap__multiselect__rails** - přidává speciální select box. Dovoluje vícenásobný výběr s vyhledáváním
- gem **carrierwave** - přidává knihovnu pro nahrávání obrázků
- gem **mini__magic** - přidává knihovnu pro práci s obrázky
- gem **money__rails** - přidává knihovnu pro práci s měnami a cenami
- gem **wicked** - přidává knihovnu pro práci s vícestránkovými formuláři

- gem **admin** - přidává naši administrační část
- gem **globalize** - přidává podporu pro práci s vícejazyčným prostředím
- gem **carrierwave** - přidává knihovnu pro práci s překlady typu hstore
- gem **amoeba** - přidává knihovnu pro práci s duplikacemi a jejich správou
- gem **wicked_pdf** - přidává knihovnu pro práci s pdf dokumenty
- gem **gibbon** - přidává nástroje pro práci s API mailchimu

4 Uživatelská dokumentace

Celá aplikace je rozdělena do velkých dvou částí. První část je zákaznická, kterou vidí každý, kdo přijde na náš internetový obchod. Druhou částí je část administrační, kam je zabezpečený přístup.

4.1 Zákaznická část

Skládá se ze 4 hlavních částí, které dohromady reprezentují jednoduchou šablonu internetové obchodu. Tato část může být svým rozsahem obrovská. Zde slouží jen pro demonstraci základních možností katalogu a objednání zboží.

- **Header** - Záhloví dokumentu
- **Left menu** - Levé menu kategorií
- **Body** - Tělo dokumentu
- **Footer** - Zápatí dokumentu

4.1.1 Header

Slouží jako základní rozcestník obchodu. Najdeme v něm informace o přihlášení, registraci nebo základní kontaktní email na provozovatele.. . Přepínače k volbě jazyka a informace o aktuální měně.

- **Levá část** - V levém horním rohu, najdeme jako první email na provozovatele. Další je ukazatel aktuální měny. Měna je momentálně implementována Česká koruna. Na pravé straně od ní nalezneme ukazatel aktuálního jazyka. Po najetí, se nám zobrazí seznam dostupných jazyků. Momentálně zde najdeme, přepínatelné jazyky Češtinu a Angličtinu.
- **Pravá část** - Zde rozhoduje, zda je uživatel přihlášený nebo není. Přihlášený uživatel, uvidí pod jakým emailem je přihlášený. Vedle něj je položka Můj účet, ta nás přesměruje na registrační informace o uživateli. Poslední je zde tlačítko pro odhlášení.
Pro uživatele, který není přihlášen, je zobrazena registrace a přihlášení.

- **Mini košík** - Položka mini-košíku, zobrazuje aktuální počet a cenu položek přidanych v košíku.

4.1.2 Levé menu kategorií

Slouží pro zobrazení všech aktivních kategorií. Kategorie jsou zanořeny do sebe. Je zde proto použito jednoduché odražení. Každá šablona, ovšem může kategorii vypisovat jinak.

4.1.3 Body

Nejdůležitější obsahová část stránek. Dynamicky se mění dle toho, kde se uživatel momentálně pohybuje. Zobrazuje obsah aktuální stránky, na které se pohybujeme.

4.1.3.1 Registrace Formulář pro registraci nového zákazníka.

Povinné informace:

- **Jméno** - Pole pro jméno uživatele.
- **Příjmení** - Pole pro příjmení uživatele.
- **Email** - Pole pro email uživatele. Musí mít správný tvar.
- **Heslo** - Pole pro zadání hesla, které musí mít minimálně 6 znaků
- **Potvrzení hesla** - Musí se shodovat se zadaným heslem.

4.1.3.2 Přihlášení Formulář pro registraci nového zákazníka.

Povinné informace:

- **Jméno** - Pole pro email uživatele.
- **Příjmení** - Pole pro heslo uživatele.

4.1.3.3 Kategorie Po kliknutí na jednotlivé kategorie se nám zobrazí katalog všech produktů, které jsou k dané kategorii přiřazeny.

4.1.3.4 Produkt Po otevření produktu vidíme detailní informace.

- **Povinné informace:**
- **Název produktu**

- **Krátký popis** - Omezený na 150 znaků, slouží ke krátkému představení bez větších detailů
- **Cena**
- **Počet kusů** - Pole pro zadání počtu kusů, které chceme přidat do košíku. Nelze zadat větší počet, než je aktuálně skladem.
- **Do košíku** - Potvrzovací tlačítko pro přidání do košíku.
- **Popis** - Detailní popis který, není nijak omezen počtem znaků.
- **Varianty** - Pokud je produkt hlavním produktem, poté nám zobrazí jeho varianty. Jinak zobrazí nadřazený produkt společně s ostatními variantami.

4.1.3.5 Košík Pokud je košík prázdný, zobrazí se nám základní popis prázdného košíku. Pokud budeme mít přidanou nějakou položku v košíku, zobrazí se nám přehledná tabulka s následujícími daty:

- **Náhled** - Miniatura náhledového obrázku
- **Název produktu** - Název produktu provázaná s odkazem na detail produktu. Pod názvem vidíme atributy přiřazené k produktu.
- **Počet** - Množství položek přidanych v košíku
- **Cena za kus**
- **Celková cena** - Cena za kus násobená počtem položek
- **X** - Značí symbol pro smazání produktu z košíku

4.1.3.6 Objednávka Objednávkový proces se skládá ze 3 kroků.

- **Krok 1** - Zadání informací o zákazníkovi. Pokud je uživatel přihlášen, jsou informace zadané při registraci vyplněny automaticky.
- **Krok 2** - Nutnost zvolit metodu dopravy a platbu k ní přiřazenou.
- **Krok 3** - Rekapitulace našeho zboží a možnost doplnit krátkou poznámku k objednávce

4.1.4 Footer

Zde máme jednoduchý 3 sloupcový návrh.

- O nás
- Informace
- Kontaktujte nás

4.1.4.1 O nás V této sekci najdeme jednoduchý popis našeho obchodu, jedná se o takové představení naší firmy. Text by neměl být příliš dlouhý. Měl by být spíše kratší a výstižný.

4.1.4.2 Informace Libovolný text do patičky zadávaný v administraci.

4.1.4.3 Kontaktujte nás Velice důležitá věc je kontaktní adresa, email, popřípadě telefon. Všechny tyto věci opět mohou být editovány v administraci.

4.2 Administrační část

Tato část se nám skládá jen ze 3 částí.

- **Login** - Úvodní přihlašovací strana
- **Header** - Záhloví dokumentu
- **Sidebar** - Levé menu
- **Body** - Tělo dokumentu

4.2.1 Úvodní přihlašovací strana

Stránka pro zabezpečení obchodu. Pro přístup k administrativní částí je vyžadován email a heslo administrátora popř. jiného uživatelského účtu.

4.2.2 Záhloví dokumentu

V levém rohu najdeme název aplikace. Na pravé straně najdeme informace o tom, kdo je přihlášen s možností změny údajů či odhlášení. Informace o aktuální jazykové mutaci s možností změny jazyka.

4.2.3 Levé menu

Základní menu, ve kterém je po přihlášení aktivní položka objednávky, aby měl administrátor okamžitý přehled.

4.2.3.1 Položky menu

- **Objednávky**
- **Produkty**
- **Kategorie**
- **Výrobci**
- **Uživatelé**

- **Zákazníci**
- **Způsob platby**
- **Způsob dopravy**
- **Atributy**
- **Nastavení**
- **Odhlásit**

4.2.4 Tělo dokumentu

4.2.4.1 Objednávky Po přesměrování vidíme jednoduchý přehled všech objednávek. V posledním sloupci máme jednotlivé akce, které můžeme s konkrétní objednávkou provádět. Je zde možnost objednávku upravit (ikona tužky) či smazat (ikona koše).

V editaci najdeme 3 hlavní záložky a tlačítko pro tisk faktur. První z nich se týká celkové ceny objednávky a metodě platby či dopravy. Ve druhé záložce najdeme přehled objednaných produktů a jejich počet či cenu. Třetí záložka zobrazuje informace o zákazníkovi. Jeho dodacích a fakturačních informací. Taktéž zde najdeme pole sloužící k informování zákazníka o změně stavu objednávky.

4.2.4.2 Produkty Hlavní strana nám nabídne přehled všech produktů. V pravém rohu tabulky se nacházejí tlačítka k přidání, hromadnému smazání a XML exportů.

K akcím nám přibyla ikona 2 listů. Tato ikona nám slouží k duplikaci produktů. Pokud nechceme podobný produkt vytvářet celý znovu, můžeme použít tuto funkci. Ihned po duplikaci jsme přesměrováni na editaci duplikovaného produktu. Duplikace produktu nám duplikuje veškeré informace včetně asociovaných dat.

Formulář je rozdělen do 3 záložek. První jsou informace o produktu. Uvedu zde jen ty povinné.

- **Název** - Název produktu v aktuální jazykové mutaci
- **Krátký popis** - Krátký popis v aktuální jazykové mutaci
- **Cena** - Cena produktu
- **Měna**
- **Počet kusů skladem** - Počet počtu kusů na skladě, pokud se rovná 0 nejde zakoupit

- **Daň - DPH**
- **Hlavní kategorie** - Hlavní kategorie do které je produkt směřován

Další záložkou jsou ostatní produktové obrázky. Zobrazují se pod hlavním produktovým obrázkem.

V poslední záložce můžeme k produktu přiřadit námi vytvořené atributy. Pokud máme více produktů stejného základu, můžeme každému přiřadit jiný atribut a přiřadit jej jako variantu. Tím získáme krásně přehledné varianty produktů.

4.2.4.3 Kategorie Kategorie jsou stěžejní částí celé aplikace, proto bylo třeba je vyřešit jednoduchou a uživatelsky příjemnou cestou. Zde nevidíme tabulku, nýbrž stromovou strukturu kategorií, kde dle zanoření vidíme jejich pořadí. Využívá technologie drag and drop. To značí, že kategorii můžeme chytit a tahem ji přesunout jinam do stromu. Zakázaný pohyb je prohození vrcholu s nejnižším listem v jednom kroku.

Tlačítko uložit nám slouží k uložení aktuální struktury stromu.

4.2.4.4 Výrobci Slouží pro potřebu sjednotit určitou značku produktů pod jednoho výrobce. 3 základní atributy jako název, popis a odkaz na webovou stránku výrobce. V mé ukázkové aplikaci nevyužité.

4.2.4.5 Uživatelé Uživatele neboli také účty s určitými pravomocemi k přístupu do naší administrace. V přehledu vidíme tabulku, tentokrát ale bez možností řazení a vyhledávání. Atribut role se nám stará o oprávnění jednotlivých akcí v systému.

- **Uživatel** - Základní pohledy bez možností editací a potvrzování
- **Manažer** - Má přístup k rozsáhlejšími funkcím než uživatel, může editovat, přidávat
- **Admin** - Neomezené oprávnění, přístup k veškerému nastavení

4.2.4.6 Uživatelé Zákazníci, kteří se zaregistrovali u nás v obchodě.

4.2.4.7 Způsob platby Možnost nastavení námi definovaných druhů plateb.

4.2.4.8 Způsob dopravy Možnost nastavení námi definovaných druhů dopravy, ke kterým je potřeba přiřadit námi vytvořené platební metody.

4.2.4.9 Daně Nastaveno v základu na DPH 21

4.2.4.10 Atributy Vytváření atributů sloužících pro vytváření variant produktů. Ke každému atributu je potřeba vytvořit jednotlivé hodnoty atributu. Ty se potom zobrazují u každého produktu a je možnost zde přiřadit atributy a jejich hodnoty.

4.2.4.11 Nastavení Nastavení obchodu je rozděleno do 3 záložek. První představuje základní informace ohledně názvu, vlastníka a kontaktech. V další záložce najdeme vše ohledně textových částí obchodu, jako jsou popisy a texty. Poslední část nastavuje základní počet produktů na stránce.

5 Programátorská dokumentace

Aplikace vyžadovala jak zákaznickou, tak administrační část. Bylo potřeba vytvořit dva systémy, nebo aplikaci rozdělit. Rozhodl jsem se jít první cestou a vytvořit dva oddělené systémy. Přesněji řečeno, je administrační část, vytvořená jako samostatná knihovna a je připojena k aplikaci. Každá z částí, má svou vlastní MVC strukturu.

5.1 Databázová vrstva

Obě částí aplikace pracují se stejnými daty. Používají tedy stejnou databázi. Nebudu zde uvádět všechny tabulky celé databáze. Uvedu zde Produkt a jeho vazby. Objednávku a její vazby do košíku a produktů.

5.1.1 Produkt

Produkt je v celé aplikaci stěžejním prvkem. Cena produktu je zadávána v nejmenší možné jednotce. Jméno, krátký popis a popis jsou zadány jako HStore. Ten jsem specifikoval na začátku práce, kde je popsána i jeho funkcionality. Při výběru kategorií, je potřeba zadat i hlavní kategorii. Z důvodu jasného určení, kam je produkt zařazen. Pole `in_stock`, slouží k nastavení počtu produktů, které aktuálně prodáváme.

Atribut	Typ
id	[Primární klíč] SERIAL
manufacturer_id	INTEGER
vat_id	INTEGER
intern_code	VARCHAR
order	INTEGER
order_quantity	TEXT
active	BOOLEAN
main_image_name	VARCHAR
main_image_file	VARCHAR
price_subunits	INTEGER
price_currency	VARCHAR
normal_price_subunits	INTEGER
normal_price_currency	VARCHAR
name_translations	HSTORE
short_description_translations	HSTORE
description_translations	HSTORE
main_category	INTEGER
in_stock	INTEGER

Tabulka 1: Tabulka produktů(products)

5.1.2 Atributy produktu

Atributy produktu, jsou k produktu připojeny přes tabulku variants. Ta nám stanovuje, který atribut, je přiřazen k jakému produktu a jaká je jeho hodnota. Pro lepší práci s atributy, je každý atribut přiřazen k produktu a uživateli stačí vybrat jen jeho hodnota.

Varianty tvoříme tak, že se vytvoří produkt a nastaví mu atributy i jejich hodnoty. Potom se přiřadí jeho nadřazený produkt a tím se stává jeho variantou. Nadřazený produkt sjednocuje produkty.

Atribut	Typ
id	[Primární klíč] SERIAL
name_translations	HSTORE
active	BOOLEAN
deleted	BOOLEAN

Tabulka 2: Tabulka atributů(properties)

Atribut	Typ
id	[Primární klíč] SERIAL
property_id	INTEGER
value_translations	HSTORE

Tabulka 3: Tabulka hodnot hodnot atributů(property_values)

Atribut	Typ
id	[Primární klíč] SERIAL
product_id	INTEGER
property_id	INTEGER
property_value_id	INTEGER

Tabulka 4: Tabulka variant(variants)

5.1.3 Kategorie

Kategorie, jsou do databáze ukládány jako strom. Ukládáme rodiče, levého potomka a pravého potomka. Pokud je záznam rodiče nulový, značí nám to, že kategorie je kořenem.

Atribut	Typ
id	[Primární klíč] SERIAL
parent_id	INTEGER
lft	INTEGER
rgt	INTEGER
slug	VARCHAR
name_translations	HSTORE
active	BOOLEAN
deleted	BOOLEAN

Tabulka 5: Tabulka kategorií(categories)

5.1.4 Zařazení produktů do kategorií

Pro možnost zařazení produktu do více kategorií, je použit číselník. Obsahuje id číselníku, id kategorie a id produktu.

Atribut	Typ
id	[Primární klíč] SERIAL
product_id	INTEGER
category_id	INTEGER

Tabulka 6: Tabulka relací mezi produkty a kategoriemi(categorizations)

5.1.5 Košík

Tabulka košíku je jednoduchá, obsahuje jen jeho id.

Atribut	Typ
id	[Primární klíč] SERIAL
order_id	INTEGER

Tabulka 7: Tabulka košíku(carts)

5.1.6 Jednotlivé položky v košíku

Aby bylo možné do košíku přidávat zboží, bylo nutné vytvořit ještě tabulku jednotlivých položek. Ta nám představuje relaci mezi produktem a košíkem. Nese přídatné informace, jako aktuální cenu, její měnu a množství položek.

Atribut	Typ
id	[Primární klíč] SERIAL
product_id	INTEGER
cart_id	INTEGER
quantity	INTEGER
price_subunits	INTEGER
price_currency	VARCHAR

Tabulka 8: Tabulka položek v košíku(cart_items)

5.1.7 Objednávky

Objednávka nese informace o zákazníkovi. Ten může či nemusí být registrován. Platební metodu a způsob dopravy. Obsahuje id košíku, které se naváže po odeslání objednávky. Status objednávky značí námi určený stav, se kterým můžeme pracovat v administraci. Step neboli krok značí do jakého kroku, se objednávka dostala. V administraci, sbíráme jen objednávky, které se dostali až do koncového kroku. V databázi, máme ale uloženy všechny objednávky, které mohou sloužit k pozdějším analýzám. V tabulce neuvádím veškeré informace o uživateli a cenách.

Atribut	Typ
id	[Primární klíč] SERIAL
shipping_id	INTEGER
payment_id	INTEGER
cart_id	INTEGER
status	INTEGER
step	VARCHAR

Tabulka 9: Tabulka objednávek (orders)

5.1.8 Doprava a platba

Způsob dopravy jsou úzce propojeny. Každá doprava nese základní údaje o službě. Dále má přiřazené možnosti plateb. Tohle řešení jsem zvolil z důvodu, aby nesešla nevhodná kombinace, kdy určitá platba, nebude při zvoleném způsobu dopravy možná.

Atribut	Typ
id	[Primární klíč] SERIAL
company_translations	HSTORE
title_translations	HSTORE
description_translations	HSTORE
price_subunits	INTEGER
price_currency	VARCHAR
deleted	BOOLEAN
active	BOOLEAN

Tabulka 10: Tabulka způsobů dopravy (shippings)

Atribut	Typ
id	[Primární klíč] SERIAL
name_translations	HSTORE
description_translations	HSTORE
price_subunits	INTEGER
price_currency	VARCHAR
deleted	BOOLEAN
active	BOOLEAN

Tabulka 11: Tabulka platebních metod (payments)

Atribut	Typ
payment_id	INTEGER
shipping_id	INTEGER

Tabulka 12: Tabulka relací mezi platbami a dopravou (payments_shippings)

5.2 Aplikační vrstva

Použit je návrhový vzor REST, který jsem popsal v úvodu. Používá u většiny modelů 4 základní operace. Vytvoření, upravení, smazání a zobrazení. U specifických akcí, jako je třeba export do XML nebo tisk faktur, jsou vytvořeny vlastní metody. Nebudu zde popisovat zákaznickou část. Popsána bude administrativní část. Všechny třídy v této části, jsou složeny z klíčového slova controller a názvu modelu ke kterému se váže. To vyvolá propojení s modelovou třídou, která se musí jmenovat stejně jako tabulka v databázi.

Od frameworku verze 4, se používají takzvané silné parametry (strong params). Ty určují jaké parametry, controller přijme z formuláře. Parametry by se ve většině případů měly rovnat názvům sloupců v databázi.

5.2.1 OrdersController

Objednávky obsahují 2 akce. První z nich je editace objednávky, ta slouží zároveň i jako náhled. Kvůli redundanci kódu jsem ponechal jen tuto akci. Při editaci je možnost odeslat informaci o změně stavu. Ta aktivuje mailer, který odesílá zákazníkovi email dle předepsané šablony. Možnost tisku faktury jako pdf soubor. Převod probíhá z html šablony do pdf. Druhá akce je smazání objednávky.

5.2.2 ProductsController

Obsahuje základní REST akce, bez zobrazení, které je opět odebráno. Přidána akce duplikace. Produkt je zduplikován, se všemi jeho vazbami a následně může být upraven. Další funkcí navíc je export do XML formátu. Funkce exportuje jen produkty, které mají počet větší než 0. XML soubor má formát validní pro Zboží.cz a obsahuje jen základní údaje. Dále může být rozšířen o další položky. Doplnkovou funkcí je zde hromadné smazání.

5.2.3 CategoriesController

Obsahuje základní REST akce. Přidána je zde funkce na přeskupení kategorií. Akce s ikonou oka, zde značí výpis produktů v dané kategorii. Pomocné třídy, starající se o organizaci kategorií, najdeme ve třídě `categories_helper`.

5.2.4 ManufacturersController

Obsahuje základní REST akce bez zobrazení. Slouží k manipulaci nad tabulkou výrobců, kteří mohou být přiřazeni k produktu.

5.2.5 UsersController

Obsahuje základní REST akce bez zobrazení. Obsluhuje přidávání uživatelů, kteří se mohou zapojit do správy administrace. Jsou rozděleni do 3 rolí. Admin má veškerá práva nad všemi akcemi v aplikaci. Manažer může obsluhovat vše okolo produktů a objednávek. Uživatel by se měl starat, výhradně o kompletaci objednávek.

5.2.6 CustomersController

Obsahuje základní REST akce bez zobrazení. Informace o zákaznících, kteří se zaregistrovali nebo byli přidáni do systému.

5.2.7 PaymentsController

Obsahuje základní REST akce bez zobrazení. Informace o platebních metodách.

5.2.8 ShippingsController

Obsahuje základní REST akce bez zobrazení. Informace o způsobech dopravy. Způsoby dopravy jsou provázány s platebními metodami.

5.2.9 VatsController

Obsahuje základní REST akce bez zobrazení. Informace o aktuální dani. Mohou být přidány vícekrát. Smysl jen v případě meziroční změně DPH.

5.2.10 PropertiesController

Obsahuje základní REST akce bez zobrazení. Informace. Při přidání či úpravě atributu, je možné přidat jeho hodnoty.

5.2.11 SettingsController

Obsahuje jedinou REST akci. Není třeba více nastavení, proto se vždy upravuje jen jedno. Základní nastavení je přidáno, při sestavení databáze automaticky.

5.3 Pohledová vrstva

Systém šablon ve frameworku automaticky hledá šablonu dle jména akce. Pro formulář, kde vytváříme nový objekt třídy Product, využijeme akci new. Proto bude šablona s názvem new, hledána ve složce products. Soubory mají příponu .html.erb. To značí že obsah je v html a máme možnost využití pomocných metod frameworku. Já jsem zde ale použil přídatnou knihovnu HAML. S tou pracuji v rámci celé aplikace.

5.3.1 Šablony

Nebudu popisovat zvlášť každou šablonu každého controlleru. Popíšu zde tedy základní šablony, se kterými se můžeme v aplikaci setkat.

5.3.1.1 Formulář (`_form.html.haml`) Stará se o vykreslení formuláře, pro konkrétní objekt. Formulář tvoří zvláštní soubor. Ten využijeme v akcích vytvoření nového objektu nebo úpravě stávajícího objektu. Tím si ušetříme spoustu zbytečného kódu. Ten by vznikl, vytvořením formuláře pro každou akci zvlášť

5.3.1.2 Nová (`new.html.haml`) Stará se o šablonu pro akci new. Ta se ve většině případů užívá k vytvoření nového objektu. Nová akce nám tedy vytvoří prázdný objekt, který pošle šabloně. Ta pro něj vytvoří formulářová pole. Odesláním formuláře se aktivuje akce create, ta dostane parametry z formuláře. Pokud vše projde validací, kterou máme v modelu, je vytvořen nový objekt. Pokud neprojde validací, vrací objekt zpět, s našimi vyplněnými daty a chybovým hlášením.

5.3.1.3 Úprava (`edit.html.haml`) Stará se o šablonu pro akci edit. Ta se stará o úpravu stávajících objektů. Proto očekává již vytvořený objekt. Pro ten následně zase vytvoří formulář. Pokud najde ve stávajícím objektu již zadanou hodnotu, vyplní ji do příslušného pole. Po odeslání opět prochází validací a ukládá. Pokud validací neprojde, opět je vrácen zpět s chybovým hlášením.

5.3.1.4 Přehled (`index.html.haml`) Nejčastěji se využívá k přehledu všech dat, v dané tabulce. Můžeme z ní dále odkazovat na příslušné pohledy zobrazených objektů.

5.3.1.5 Zobrazení (`show.html.haml`) Zobrazuje předaný objekt. Neslouží k odesílání informací.

5.3.1.6 Hlavní kostra (`application.html.haml`) Představuje základní kostru webové aplikace. Můžeme do ní přidat námi vytvořené šablony. Nejčastější rozdělení na záhlaví, boční panel, tělo a zápatí.

Závěr

Moje práce se snaží cílit na menší podnikatele či firmy, kteří by rádi začali své podnikání a nechce se jim platit velké sumy za aplikaci na míru. Rádi by využili nějaký systém zdarma, který ovšem není příliš robustní. Manipulace s ním by měla být rychlá, intuitivní a jednoduchá. Použití moderních technologií zaručuje velkou rychlost a potenciál růstu do budoucna.

Uživatelům je k dispozici přehledný a jednoduchý administrační systém, ve kterém jdou přehledně spravovat všechny části jak administrační, tak zákaznické sekce. K dispozici jsou jazykové mutace a do budoucna plánuji plně implementovat podporu cizích měn, které jistě ocení každý kdo by rád cílil i na zákazníky z jiných zemí.

Aplikace má před sebou ještě dlouhou cestu vývoje. Do budoucna bych rád přidal rozdělení zákazníků do skupin s odlišnými cenami. Další velice důležitou částí, kterou bych rád implementoval, jsou testy. Testování je dnes nedílnou součástí vývoje aplikace. Pomůže nám vyvarovat se chybám, které se mohou projevit po implementaci nových funkcí a nemusíme si jich na první pohled všimnout. Vytvoření testů, pro otestování celé aplikace nám mnohdy zabere více času, než vytvoření aplikace samotné. Určitě bych rád pokračoval ve vývoji aplikace.

Conclusions

My thesis is trying to target on a smaller businessmen which would start their own business. They dont want pay really big prices for commercial eshop solution. They would like to try an open source system which is not so big. Work with this system should be fast, intuitive and simple. Usage of modern technologies guarantees high speed and future growth potential.

Users have available well-arranged and easy administration system. This system can manage all parts of administration or customer site. Also you can use a different languages. Future idea is to implement different currencies and target on foreign customers.

It takes some time to improve this application. Future ideas are to add dividing to groups with differents prices. The next important part is testing. Testing is very helpful part of application development. It should help us to avoid errors which can appear after implementation of new parts or after bug fixes. Unfortunately it takes a lot of time to create tests for whole application. I would like to continue with eshop development.

Reference

- [1] *Oficiální web jazyka ruby.*
Dostupné z: <https://www.ruby-lang.org>
- [2] *Oficiální dokumentace k frameworku Ruby on Rails.*
Dostupné z: <http://guides.rubyonrails.org>
- [3] *Railscasts video návody*
Dostupné z: <http://railscasts.com>
- [4] *Oficiální dokumentace PostgreSQL*
Dostupné z: <http://www.postgresql.org/docs/>
- [5] *W3schools návody*
Dostupné z: <http://www.w3schools.com>
- [6] *Bootstrap framework*
Dostupné z: <http://www.getbootstrap.com>
- [7] *Dokumentace k HAML*
Dostupné z: <http://haml.info/docs.html><http://coffeescript.org>
- [8] *Dokumentace k HStore doplňku*
Dostupné z: <http://www.postgresql.org/docs/9.1/static/hstore.html>
- [9] *Dokumentace ke CoffeeScriptu*
Dostupné z: <http://coffeescript.org>

A Obsah přiloženého CD/DVD

bin/

Kompletní adresářová struktura webové aplikace (v ZIP archivu) pro zkopírování na webový server. Adresář obsahuje i všechny runtime knihovny a další soubory potřebné pro bezproblémový provoz webové aplikace na webovém serveru.

doc/

Text práce ve formátu PDF, vytvořený s použitím závazného stylu KI PřF UP v Olomouci pro závěrečné práce, včetně všech příloh, a všechny soubory potřebné pro bezproblémové vygenerování PDF dokumentu textu (v ZIP archivu).

src/

Kompletní zdrojové texty webové aplikace se všemi potřebnými (příp. převzatými) zdrojovými texty, knihovnami a dalšími soubory potřebnými pro adresářové struktury pro zkopírování na webový server.

readme.txt

Instrukce pro nasazení webové aplikace na webový server, včetně všech požadavků pro její bezproblémový provoz, a webová adresa, na které je aplikace nasazena pro účel testování při tvorbě posudků práce a pro účel obhajoby práce.