

UNIVERZITA PARDUBICE

Fakulta elektrotechniky a informatiky

E-shop s podporou mezinárodního prodeje

Pavel Macháň

Bakalářská práce
2014

Prohlášení autora

Prohlašuji, že jsem tuto práci vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 30. 4. 2014

Pavel Macháň

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2013/2014

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Pavel Macháň**
Osobní číslo: **I09184**
Studijní program: **B2646 Informační technologie**
Studijní obor: **Informační technologie**
Název tématu: **E-shop s podporou mezinárodního prodeje**
Zadávací katedra: **Katedra informačních technologií**

Z á s a d y p r o v y p r a c o v á n í :

V úvodní části práce je nutné provést přehled a porovnání nejčastěji používaných PHP frameworků. Především pak frameworků CakePHP, Nette, QPHP, Symfony a Zend. Při porovnávání frameworků bude kladen důraz na podporu vývoje multijazyčných lokalizovaných webových aplikací.

Primárním cílem bakalářské práce je návrh a implementace softwarového řešení univerzálního multijazyčného lokalizovaného e-shopu za využití vhodného PHP frameworku. V rámci lokalizace je nutno mimo jiné počítat s variantou umožňující:

Používat v různých jazykových mutacích různou měnu,
nabízet konkrétní produkt pouze na vybraných jazykových mutacích,
zakázat zobrazování vybraných fotografií produktů ve vybraných jazykových mutacích, spravovat na různých jazykových mutacích odlišné kategorie produktů.

Produkt bude otestován a nasazen do ostrého provozu, kde nahradí stávající systém provozovaný na adrese: <http://mmproducts.cz>

Předpokládané využití technologií: PHP, HTML5, CSS3, JavaScript, relační databáze

Rozsah grafických prací:

Rozsah pracovní zprávy:

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

Seznam odborné literatury:

BAGUI, Sikha a Richard EARP. Database design using entity-relationship diagrams. 2nd ed. Boca Raton, FL: CRC Press, c2012, xxvii, 343 p. Foundations of database design series. ISBN 14-398-6176-5.

BORONCZYK, Ed a Steven D NOWICKI. PHP 6: programujeme profesionálně. Vyd. 1. Překlad Ondřej Gibl. Brno: Computer Press, 2010, 718 s. Programujeme profesionálně, no. 2. ISBN 9788025127674.

PILGRIM, Mark. Dive into HTML5. [online] 2011 [cit. 2012-10-11]. Dostupné z WWW: <http://diveintohtml5.info>, <http://kniha.html5.cz>

Vedoucí bakalářské práce:

Ing. Jan Hříděl


Katedra informačních technologií

Datum zadání bakalářské práce: **20. prosince 2013**

Termín odevzdání bakalářské práce: **9. května 2014**



prof. Ing. Simeon Karamazov, Dr.
děkan



Ing. Lukáš Čegan, Ph.D.
vedoucí katedry

V Pardubicích dne 31. března 2014

Poděkování

Chtěl bych na tomto místě poděkovat vedoucímu mé bakalářské práce Ing. Janu Hřídeli za rady při zpracování této práce a komunitě okolo Nette Framework, za školení v Nette Framework. Také bych chtěl poděkovat mé rodině za podporu během studia.

Anotace

Práce se zabývá implementací softwarového řešení univerzálního, vícejazyčného, lokalizovaného e-shopu, za využití vhodného PHP frameworku. Výběr vhodného frameworku proběhl na základě teoretické rešerše nejčastěji používaných PHP frameworků, s ohledem na podporu vývoje vícejazyčných lokalizovaných aplikací. Předkládané řešení bude nasazeno do ostrého provozu, kde nahradí stávající, zastaralý systém provozovaný na adrese: <http://mmproducts.cz>.

Klíčová slova

Internetový obchod, PHP, MySQL, Nette Framework, HTML, CSS

Title

E-shop support international sales

Annotation

The thesis deals with an implementation of a universal multilanguage-localized e-shop. The suitable PHP framework was selected on the base of theoretical research and used for realization. The suitability of the framework was assessed according to its support of multilingual application. Presented application will be used as a replacement of obsolete system at the address: <http://mmproducts.cz>.

Keywords

E-shop, PHP, MySQL, Nette Framework, HTML, CSS

Seznam zkratek

| | |
|------|------------------------------------|
| PHP | Hypertext Preprocessor |
| HTML | HyperText Markup Language |
| CSS | Cascading Style Sheets |
| LESS | Leaner CSS |
| JSON | JavaScript Object Notation |
| WWW | World Wide Web |
| URL | Uniform Resource Locator |
| OEM | Original Equipment Manufacturer |
| ISV | Independent Software Vendor |
| VAR | Value-Added Reseller |
| CVS | Concurrent Version System |
| DI | Dependency Injection |
| JS | JavaScript |
| IDE | Integrated Development Environment |
| DRY | Don't repeat yourself |
| KISS | Keep it simple, stupid |
| MVC | Model-View-Controller |
| XSS | Cross Site Scripting |
| CSRF | Cross-Site Request Forger |
| UML | Unified Modeling Language |
| SQL | Structured query language |
| ORM | Object-relational mapping |
| SSL | Secure Sockets Layer |
| ER | Entity Relationship |
| PK | Primary Key |
| FK | Foreign Key |

Seznam obrázků

| | |
|--|----|
| Obrázek 1 - Diagram návrhového vzoru MVC [3]..... | 12 |
| Obrázek 2 - Ukázka CakePHP: konzole Bake [5]..... | 13 |
| Obrázek 3 - Symfony: ukázka použití překladače..... | 16 |
| Obrázek 4 - Zend: paměťová náročnost [6]..... | 17 |
| Obrázek 5 - Zend: ukázka použití překladače | 18 |
| Obrázek 6 - Nette: ukázka formuláře | 19 |
| Obrázek 7 - Nette Framework: ukázka jednoduché překladové služby | 21 |
| Obrázek 8 - Ukázka HTML..... | 25 |
| Obrázek 9 – Aktuální zastoupení PHP | 25 |
| Obrázek 10 - Ukázka SQL dotazu | 26 |
| Obrázek 11 - Ukázka zápisu konfiguračního souboru..... | 27 |
| Obrázek 12 - LESS vs. CSS | 28 |
| Obrázek 13 - IDE: PhpStorm..... | 29 |
| Obrázek 14 - Rich Picture | 30 |
| Obrázek 15 - UML Activity Diagram | 31 |
| Obrázek 16 - Use Case Diagram | 32 |
| Obrázek 17 - Ukázka MySQL Workbench | 33 |
| Obrázek 18 - ER diagram | 34 |
| Obrázek 19 - Ukázka použití lokalizačního klíče..... | 35 |
| Obrázek 20 - ER diagram: lokalizace..... | 35 |
| Obrázek 21 - ER diagram: produkty | 38 |
| Obrázek 22 - ER diagram: měny | 39 |
| Obrázek 23 - ER diagram: účty | 40 |

| | |
|--|----|
| Obrázek 24 - ER diagram: platební služby..... | 41 |
| Obrázek 25 - ER diagram: dopravní služby | 42 |
| Obrázek 26 - ER diagram: DPH..... | 43 |
| Obrázek 27 - ER diagram: objednávky | 44 |
| Obrázek 28 - ER diagram: texty | 45 |
| Obrázek 29 - SQL: SELECT pro zobrazení kategorie | 45 |
| Obrázek 30 - SQL: UPDATE pro upravení adresy | 46 |
| Obrázek 31 - SQL: DELETE pro mazání účtu..... | 46 |
| Obrázek 32 - SQL: Procedura pro duplikaci lokalizace produktů..... | 46 |
| Obrázek 33 - SQL: Trigger pro volání procedury | 46 |
| Obrázek 34 – Wireframe | 47 |
| Obrázek 35 – Porovnání starého a nového vzhledu | 48 |
| Obrázek 36 - Registrace uživatele | 49 |
| Obrázek 37 - Nákupní košík..... | 50 |
| Obrázek 38 - Upravování platební služby | 51 |
| Obrázek 39 - Ukázka aplikace: veřejná část..... | 55 |
| Obrázek 40 - Ukázka aplikace: administrace | 56 |
| Obrázek 41 - Ukázka zdrojových kódu: bootstrap.php | 57 |
| Obrázek 42 - Ukázka zdrojových kódu: BaseAuthenticator.php | 58 |

Seznam tabulek

| | |
|---------------------------------------|----|
| Tabulka 1 - Porovnání frameworků..... | 23 |
|---------------------------------------|----|

Obsah

| | |
|---|----|
| Seznam zkratk..... | 7 |
| Seznam obrázků..... | 8 |
| Seznam tabulek..... | 9 |
| Úvod | 11 |
| 1 PHP Frameworky | 12 |
| 1.1 CakePHP | 13 |
| 1.2 QPHP | 14 |
| 1.3 Symfony..... | 14 |
| 1.4 Zend | 16 |
| 1.5 Nette..... | 18 |
| 1.6 Závěrečné srovnání | 22 |
| 2 Implementace multijazyčného lokalizovaného e-shopu..... | 24 |
| 2.1 Použité technologie..... | 24 |
| 2.2 Architektura aplikace | 29 |
| 2.3 Návrh databázového modelu..... | 32 |
| 2.4 Vzhled aplikace..... | 47 |
| 2.5 Popis a ukázky funkcí aplikace..... | 49 |
| Závěr..... | 52 |
| Literatura | 53 |
| Příloha 1 – Ukázka aplikace: Veřejná část | 55 |
| Příloha 2 – Ukázka aplikace: administrace..... | 56 |
| Příloha 3 – Ukázka zdrojových kódů | 57 |
| Příloha 4 – Obsah přiloženého CD | 59 |

Úvod

Cílem práce bylo vytvořit softwarové řešení univerzálního, vícejazyčného, lokalizovaného e-shopu. Ze zadání práce vyplývají základní požadavky na vypracování, především na teoretickou rešerši, která slouží k získání přehledu v oblasti tzv. „frameworků“. K dalším požadavkům, kladeným na praktickou část práce, patří např. možnost využití různých měn v závislosti na jazykové mutaci, omezení nabídky určitého produktu pouze pro danou jazykovou mutaci, nebo možnost zakázání zobrazení vybraných fotografií produktů v závislosti na konkrétní jazykové mutaci. Během vývoje základního konceptu aplikace vyvstala otázka, zda tato omezení vázat pouze na danou jazykovou mutaci, nebo spíše na region, ze kterého uživatel do e-shopu přistupuje

Na základě teoretické rešerše z kap. 1 byl vybrán PHP framework Nette, který splňuje základní požadavky z hlediska podpory vícejazyčných, lokalizovaných aplikací. Nezanedbatelnou výhodou je jeho funkční podpora a velká komunita vývojářů, kteří ho aktuálně využívají. Výběrem vhodného frameworku se zabývá kapitola 1. Jsou zde porovnány frameworky: CakePHP, QPHP, Symfony, Zend, Nette a diskutovány jejich základní vlastnosti, výhody a nevýhody pro danou aplikaci (kap. 1.6).

Kapitola 2 se zabývá praktickou implementací e-shopu. V části 2.1 je uveden výčet použitých technologií, mezi které patří HTML, LESS, MySQL a další. Ke každé technologii je uvedena stručná charakteristika a příklad jejího využití. Velká pozornost je v práci věnována návrhu samotného konceptu architektury aplikace. Z hlediska organizace jednotlivých myšlenek a nápadů do jednoho funkčního celku je vhodné použít nástroj, který tento proces usnadní. Při návrhu architektury práce byl využit „Use Case“ diagram, „Activity Diagram“ v modelovacím jazyku UML a „Rich Picture“. Stěžejním úkolem praktické části práce byl návrh databázového modelu, který je podrobně popsán v kapitole 2.3. Je zde uvedena struktura a vlastnosti databázových tabulek včetně jejich vazeb. Vzhledem aplikaci se zabývá kapitola 2.4. Aplikace byla navržena s ohledem na intuitivnost ovládání. Bylo zvoleno jednoduché rozložení ovládacích prvků a líbivý, funkční design, který uživatele podvědomě navádí správným směrem.

V závěru práce jsou shrnuty získané poznatky, funkčnost aplikace a navrženy možnosti pro budoucí vývoj aplikace.

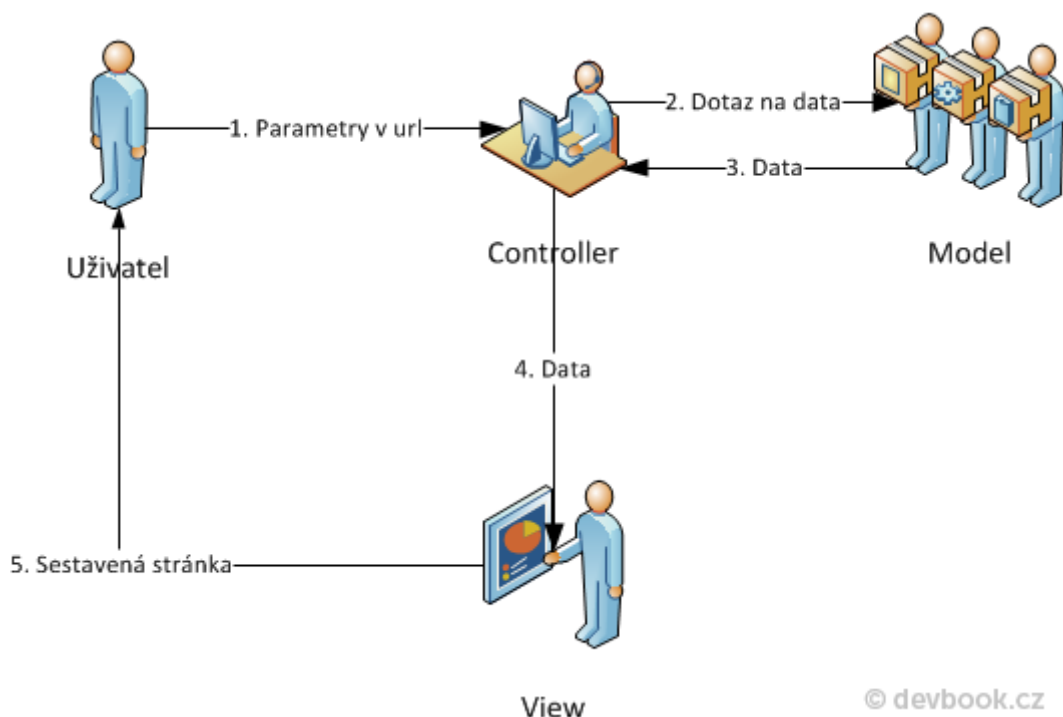
1 PHP Frameworky

Framework je softwarová struktura, která slouží jako podpora při programování, vývoji a organizaci jiných softwarových projektů. Může obsahovat podpůrné programy, knihovnu API, návrhové vzory nebo doporučené postupy při vývoji [1].

Cílem frameworku je převzetí typických problémů dané oblasti, čímž je vývoj usnadněn tak, aby se návrháři a vývojáři mohli soustředit pouze na svá zadání. Díky využití frameworku lze čas potřebný pro vývoj aplikace výrazně zkrátit, a tím uspořit náklady spojené s časovou dotací projektu.

Moderní PHP frameworky využívají architekturu MVC. Základní myšlenkou MVC je oddělení logiky od výstupu. Řeší tedy problém tzv. "špagetového kódu", kdy jsou v jednom souboru (třídě) logické operace a zároveň vykreslování výstupu. Soubor tedy obsahuje databázové dotazy, logiku (např. PHP operace) a různě rozmístěné HTML značky.

Architektura MVC dělí aplikaci na 3 logické části tak, aby je bylo možné upravovat samostatně a dopad změn jedné části na ostatní, byl co nejmenší. Tyto tři části jsou *Model*, *View* a *Controller*. *Model* reprezentuje data a business logiku aplikace, *View* zobrazuje uživatelské rozhraní a *Controller* má na starosti tok událostí v aplikaci a obecně aplikační logiku [2].



Obrázek 1 - Diagram návrhového vzoru MVC [3]

1.1 CakePHP

Jeden z nejnámějších a nejpoužívanějších PHP frameworků je CakePHP. Jeho první verzi vytvořil polský programátor Michal Tatarynowicz na počátku roku 2005. V prosinci téhož roku založili L. Masters a G. J. Woodworth nadaci Cake Software Foundation určenou pro jeho podporu a rozvoj. Verze 1.0 byla vydána v květnu roku 2006 a to pod svobodnou licencí MIT, která je jedna z nejvolnějších.

Ve své základní struktuře využívá velké množství vlastností z Ruby on Rails. Je zaměřen na rychlý vývoj celé aplikace při využití návrhového vzoru MVC. Jeho struktura je srozumitelně uspořádána a tím je zajištěna i snadná orientace v celém kódu. Poskytuje velké množství nástrojů používaných například pro jednoduché ověřování uživatelů nebo důkladnou validaci vstupních dat [4].

1.1.1 Konzole Bake

CakePHP obsahuje velice užitečný nástroj *Bake Console*. Jedná se o konzolu, která dokáže automaticky vygenerovat kusy kódu na základě databázového modelu a konfigurace aplikace. Umožňuje vygenerování modelů, pohledů, řadičů, testů a mnoho dalších částí aplikace (viz. Obrázek 2).

Díky tomuto nástroji můžeme za několik minut nejen vygenerovat hlavní kostru aplikace, ale i administrační rozhraní. Konzole Bake ušetří mnoho času, který by musel být vynaložen na ruční psaní kódu.

```
-----  
App : app  
Path: /path-to/project/app  
-----  
Interactive Bake Shell  
-----  
[D]atabase Configuration  
[M]odel  
[V]iew  
[C]ontroller  
[P]roject  
[F]ixture  
[T]est case  
[Q]uit  
What would you like to Bake? (D/M/V/C/P/F/T/Q)  
>
```

Obrázek 2 - Ukázka CakePHP: konzole Bake [5]

1.1.2 Lokalizace

Lokalizace je v CakePHP řešena velice jednoduše, pomocí komponenty *cakephp/localized*, která je dostupná pomocí nástroje Composer, jako samostatný balíček. Umožňuje překlad jednotlivých položek, databázových modelů a šablon. Využívá známé principy z GNU gettext, které se využívají pro psaní multijazyčných programů v Unixových operačních systémech. Jednotlivé jazykové mutace jsou uloženy v souborech s příponou ".po" (Portable Object).

1.1.3 Výhody a nevýhody

Výhody:

- velice jednoduchý framework,
- není vyžadována žádná konfigurace,
- kvalitní dokumentace,
- obrovská komunita,
- tisíce rozšíření a doplňků,
- rozdělen na jednotlivé komponenty

Nevýhody:

- nutnost dodržování konvencí frameworku,
- asociativních pole místo entit.

1.2 QPHP

QPHP začal v roce 2005 jako malý uzavřený dílčí projekt, pomocí kterého byla vytvořena administrace jedné osobní webové stránky. Na konci roku 2007 vyšla jeho poslední stabilní verze 0.9 vydaná pod licencí MIT. Do dnešního dne se nezachovala oficiální stránka, dokumentace ani zdrojové kódy.

QPHP Framework byl událostmi řízený, objektově orientovaný webový aplikační framework na bázi komponent, který byl vytvořen s cílem usnadnit rozvoj internetových stránek založených na PHP4 / PHP5. Využíval návrhový vzor MVC, podporoval technologii AJAX a obsahoval podporu pro více databázových systémů.

1.3 Symfony

Symfony je webový aplikační PHP framework, který využívá softwarovou architekturu MVC. Je aktuálně vydáván společností SensioLabs jako svobodný software pod licencí MIT. Symfony framework byl od začátku koncipován tak, aby byl rychlý, upřednostňoval výkon, byl flexibilní a plně konfigurovatelný. Společnost SensioLabs zajišťuje plnou podporu pro všechny „major“ verze po dobu 3 let.

Jedná se o světově velice rozšířený framework. Komunita je tvořena tisíci vývojáři, kteří framework rychle rozvíjejí a posunují vpřed. Má obsáhlou a podrobnou dokumentaci a mnoho oficiálních i neoficiálních rozšíření.

Symfony je mezinárodně uznávaná vývojářská platforma. Je podporováno jednak společností SensioLabs, která se více jak 13 let zabývá vývojem webových aplikací, ale také živou komunitou. Mnoho zlepšení, která zjednodušují práci, bylo buď převzato z jiných systémů (např. *dependency injection* z Javy), nebo bylo vyvinuto přímo pro Symfony (např. vývojářská lišta, webový profiler a další). Díky tomu, že se Symfony framework snaží co nejvíce dodržovat dohodnuté standardy, nenutí uživatele používat výhradně jeho produkty a komponenty. Uživatel si tak může pro svůj projekt vybrat knihovny, jaké chce [8].

Symfony používá šablonovací systém Twig, který má podobné funkce jako Latte v Nette včetně dědění šablon, automatického escapování atd. Výhodou Twigu je především lepší podpora v IDE. Jinak jsou oba systémy srovnatelné [7].

1.3.1 Konzole

Jedna z vynikajících součástí Symfony je komponenta konzole, která umožňuje vytvářet příkazy příkazového řádku. Jedná se o samostatný balík, který lze použít i v jiném PHP frameworku. Instalovat jej lze pomocí nástroje Composer, jako balíček dostupný pod názvem *symfony/console*, nebo lze použít oficiální repositář Git pro roční instalaci.

Konzole také umožňuje „ovládat“ Symfony aplikaci pomocí příkazů z příkazové řádky. Zapisují se následujícím způsobem: `$ app/console [název příkazu] [argumenty]` tedy např. `$ app/console cache:warmup`. Příklady nejpoužívanějších příkazů [8]:

- **cache:clear** – vymaže všechnu mezipaměť,
- **doctrine:database:create** – vytvoří databázi dle definic entit Doctrine2,
- **doctrine:schema:validate** – zkontroluje, jestli schéma DB odpovídá definicím entit Doctrine2,
- **assetic:dump –watch** – vytvoří různé předpřipravené javascripty, css, obrázky atd. a v případě změny je automaticky aktualizuje.

1.3.2 Lokalizace

Pro potřeby lokalizace obsahuje Symfony službu jménem *Translation*, která implementuje rozhraní *TranslatorInterface*. Tato služba je volně dostupná jako samostatný Composer balíček *symfony/translation*. Podporuje hlavní překladové formáty, jako jsou pole PHP, CSV, Gettext, INI a mnoho dalších.

```

use Symfony\Component\Translation\Translator;
use Symfony\Component\Translation\Loader\ArrayLoader;

$translator = new Translator('fr_FR');
$translator->addLoader('array', new ArrayLoader());
$translator->addResource('array', array(
    'Symfony2 is great!' => 'J\'aime Symfony2!',
), 'fr_FR');

echo $translator->trans('Symfony2 is great!');

```

Obrázek 3 - Symfony: ukázka použití překladače

1.3.3 Výhody a nevýhody

Výhody:

- dlouhá podpora „major“ verzí,
- velmi kvalitní dokumentace,
- flexibilní,
- snadno uchopitelný,
- obrovská komunita,
- tisíce rozšíření a doplňků,
- rozdělen na jednotlivé komponenty,
- podpora nástroje Composer.

Nevýhody:

- nepraktická komponenta ladění,
- složité vytváření formulářů pomocí asociativních polí,
- pomalé vývojové prostředí.

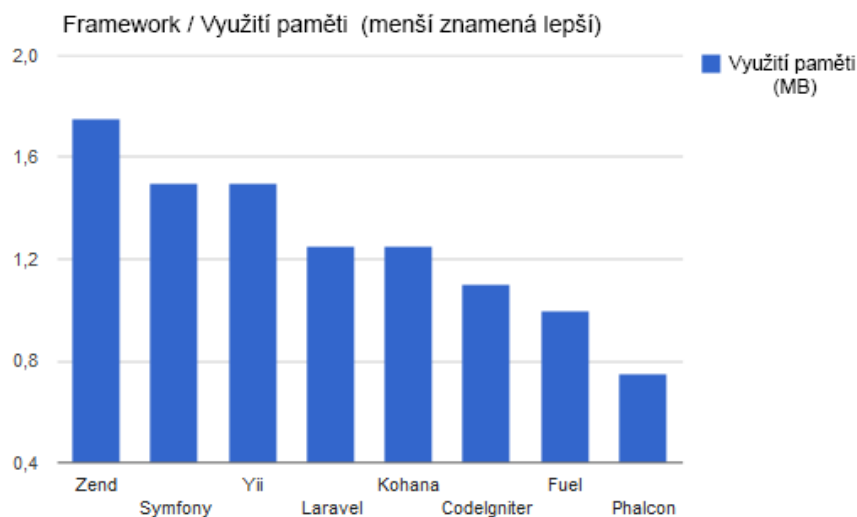
1.4 Zend

Zend Framework je vydáván firmou Zend Technologies jako svobodný software pod New BSD licencí pro vývoj webových aplikací a služeb pomocí PHP 5.3+. Využívá objektově orientovaný kód a většinu nových vlastností PHP 5.3 (jmenné prostory, pozdní statické vazby, lambda funkce a closure).

Nabízí propracovanou a robustní implementaci MVC, podporu pro multi-databázové systémy zahrnující MySQL, Oracle, IBM DB2 MSSQL Server, PostgreSQL, SQLite a Informix Dynamix Server. Obsahuje desítky komponent, jako jsou formuláře, filtry, validátory, stránkovače a mnoho dalších. Má předpřipravenou podporu pro mnoho známých služeb. Mezi nejznámější patří Amazon S2, Windows Azure, Flickr, ReCaptcha a Twitter.

Dokumentace je u Zend Framework velmi kvalitní a propracovaná, snadno se v ní orientuje. Je také doplněna velkým množstvím příkladů a video návodů. Oproti jiným frameworkům má Zend dokumentaci opravdu rozsáhlou. Pokud by nedostačovala dokumentace, jsou na knižním trhu desítky titulů, které uživatele provedou od úplného začátku.

Obsáhlost tohoto frameworku si vyžádalo svou daň. Oproti ostatním je znatelně pomalejší a má vyšší HW nároky a to především na paměť (viz. Obrázek 4).



Obrázek 4 - Zend: paměťová náročnost [6]

Zend neobsahuje žádný oficiální šablonovací nástroj, ale jej do něho snadno integrovat. Lze zvolit například Twig, Latte, SMARTY nebo jiný.

1.4.1 Lokalizace

Zend přichází s kompletní překladovou službou, která podporuje všechny hlavní formáty a zahrnuje oblíbené funkce, jako jsou překlady s podporou textových oblastí, které rozlišují množství. Lokalizační komponenta *Zend\I18n* je volně stažitelná z oficiálního webu frameworku, nebo jej lze stáhnout pomocí nástroje Composer jako balík *zendframework/zend-i18n*. Podporuje hlavní překladové formáty, jako jsou pole PHP, CSV, Gettext, INI a mnoho dalších. Použití překladové služby je velice jednoduché (viz. Obrázek 5).

```

$trans = new Zend_Translate(
    'array',
    '/www/lang/',
    'de_AT',
    array('scan' => Zend_Translate::LOCALE_DIRECTORY)
);

$trans->translate('I said hello');

```

Obrázek 5 - Zend: ukázka použití překladače

Služba umí ukládat překlady do mezipaměti. Tato vlastnost ušetření neustálé načítání a následnou analýzu jednotlivých formátů, ale také zaručí optimálnost zavádění procesu při každé žádosti o data. Povolení ukládání do mezipaměti lze docílit zavedením Zend cache adaptéru nacházející se ve jmenném prostoru *Zend\Cache\Storage\Adapter*.

1.4.2 Výhody a nevýhody

Výhody:

- velmi kvalitní dokumentace,
- flexibilní,
- obrovská komunita,
- tisíce rozšíření a doplňků,
- jednotlivé komponenty lze použít samostatně,
- kvalitně zpracovaný manuál se spoustou příkladů.

Nevýhody:

- pomalejší křivka učení,
- vyšší HW náročnost,
- pomalejší.

1.5 Nette

Nette Framework vytvořil a publikoval český vývojář David Grudl jako svobodný software, nabízený pod licencemi GNU GPL a New BSD. Tento český framework se zaměřuje především na eliminaci bezpečnostních rizik, na velkou znovupoužitelnost kódu, snadné a bezpečné použití formulářů. Podporuje AJAX, DRY, KISS a využívá softwarovou architekturu MVC. Výhodou tohoto frameworku je zejména využití událostmi řízeného programování.

Jak se Nette Framework liší od jiných frameworků? Nette Framework je stavěný tak, aby byl co nejpoužitelnější a nejvstřícnější. Jde o framework, s nímž je nejen snadné, ale i zábavné pracovat. Používá srozumitelnou a úspornou syntaxi, vychází vstříc při programování a debugování, nechává uživatele soustředit se na kreativní stránku vývoje a nepřiděluje mu vrásky. Eliminuje bezpečnostní rizika. Lze v něm tvořit e-shopy, wiki, blogy, CMS, atd.

Nette Framework používají významné společnosti jako třeba T-Systems, GE Money, Mladá fronta, VLTAVA-LABE-PRESS, Internet Info, DHL, Logio, ESET, Actum, Slevomat, Socialbakers, SUPRAPHON. V anketě serveru „Zdroják“ byl zvolen jako nepopulárnější a nejpoužívanější framework v České Republice [9].

1.5.1 Formuláře

Oproti ostatním PHP frameworkům má Nette velice propracované a uživatelsky přívětivé formuláře, které poskytují zabezpečení proti zranitelnostem. Ochrání aplikaci před útokem XSS i CSRF, odfiltruje ze vstupů kontrolní znaky, ověří validitu UTF-8 kódování a zjistí, jestli nejsou položky vybrané v select boxech podvržené (viz. Obrázek 6).

```
require 'vendor/autoload.php';

use Nette\Forms\Form;

$form = new Form;
$form->addText('name', 'Jméno:');
$form->addPassword('password', 'Heslo:');
$form->addSubmit('send', 'Registrovat');

echo $form;
```



Obrázek 6 - Nette: ukázka formuláře

Největší slabinou Nette formulářů bylo do verze 2.1 sestavování a validace dynamických formulářů u kterých není předem znám konečný počet komponent (často se využívají pro AJAXový DataGrid. Pro tyto účely vzniklo několik rozšíření, která dokázala replikovat jednotlivé položky formuláře, a tím umožnit dynamické sestavování formulářů. Jeden z nejznámějších formulářových replikátorů je „Kdyby/Replicator“ od Filipa Procházky, který je volně dostupný jako Composer balíček vydávaný pod licencí New BSD či GNU GPL.

Do verze 2.1 byla do formulářů implementována metoda *getHttpData*, umožňující práci s již ošetřenými nízko úroňovými daty formulářů, obsahující i komponenty, které byly přidány do formuláře až za běhu aplikace. Metoda *getHttpData* má jako první parametr typ

elementu, který udává, o jaký typ dat se jedná. Druhý parametr odpovídá HTML atributu *name* dané komponenty.

1.5.2 Lokalizace

Je-li zapotřebí lokalizovat aplikaci, musí být nejdříve zjištěno, jaký jazyk uživatel preferuje (v případě, že nebyl nastaven ručně). Zatím to účelem obsahuje třída *Request*, která je ve jmenném prostoru *Nette\Http*, metodu *detectLanguage*. Metoda *detectLanguage* zjišťuje z hlavičky HTTP požadavku *Accept-Language* preferované jazyky. Tyto jazyky jsou porovnávány s dostupnými jazyky aplikace, které jsou dodány parametrem *\$langs*. Je-li dostupných více jazyků, je vybrán jazyk s nejvyšší prioritou.

Nette Framework neobsahuje hotové řešení překladové služby, ale pouze rozhraní *ITranslator*, využívané pro implementaci překladových služeb, které se využívají například u šablon a formulářů. Rozhraní *ITranslator* se nachází ve jmenném prostoru *Nette\Localization* a obsahuje pouze jedinou metodu *translate()*.

```

<?php

class MyTranslator implements \Nette\Localization\ITranslator {

    /** @var int */
    private $langId;

    /** @param int $langId */
    public function __construct($langId) {
        $this->langId = $langId;
    }

    /**
     * Translates the given string.
     * @param string $message
     * @param int $count
     * @return string
     */
    public function translate($message, $count = NULL) {
        $translations = array(
            'hello' => array(
                1 => 'Hello world',
                2 => 'Ahoj světe'
            )
        );

        if (isset($translations[$message][$this->langId])) {
            return $translations[$message][$this->langId];
        }

        return 'Unknown message';
    }
}

```

Obrázek 7 - Nette Framework: ukázka jednoduché překladové služby

1.5.3 Výhody a nevýhody

Výhody:

- Velká a aktivní komunita v ČR,
- dokumentace v českém jazyce,
- strmá křivka učení,
- důmyslné zabezpečení před zranitelnostmi,
- intuitivní šablonovací systém,
- kvalitní ladící nástroje,
- propracované formuláře,
- automatické escapování dat,

- podpora nástroje Composer.

Nevýhody:

- Malá komunita v zahraničí,
- občasná zastaralost dokumentace,
- menší počet rozšíření,
- často nezachovávání zpětné kompatibility v nových verzích.

1.6 Závěrečné srovnání

CakePHP je velice zajímavý a výkonný framework, který obsahuje veliké množství funkcí. Jeho podpora pro lokalizaci aplikací je dostačující a využívá známé praktiky využívané v Unixových operačních systémech. Má kvalitní a podrobnou dokumentaci, velkou komunitní základnu a velký počet rozšíření a doplňků. Jeho největší nevýhodou je nutnost dodržování CakePHP konvencí, které dělají ze začátku velké problémy a tím degradují křivku učení. Entity se řeší jako asociativní pole, což dělá problémy vývojovému prostředí a blokuje inspekci psaného kódu.

QPHP byl na svou dobu velice dobrý framework využívající MVC architekturu. Bohužel se do dneška nezachoval a neexistuje už ani oficiální stránka nebo dokumentace. Informace o tomto frameworku jsou dostupné jen díky webovým archivačním službám, které znamenaly jen minimální část.

Oproti tomu Symfony má obrovskou komunitu a těší se z velké popularity, díky které vzniklo tisíce rozšíření a komponent. Je rozdělen na jednotlivé části, které se dají využít i v aplikacích postavených na jiném PHP frameworku. Podporuje šablonovací systém Twig, obsahuje generátor kódu a výkonnostně nezaostává. Obsahuje kompletní překladové služby, které podporují všechny hlavní formáty a jsou dostupné jako Composer balíček *symfony/translation*. Na rozdíl od Nette má nedokonalou ladící komponentu, která nezobrazuje všechny potřebné údaje. Vytváření formulářů pomocí asociativních polí není moc intuitivní a často to vede k vytvořením chyb.

Zend framework v sobě zahrnuje komponenty pro MVC aplikace, autorizaci a autentizaci, implementuje různé druhy vyrovnávacích pamětí, filtrů a validatorů pro uživatelská data, jazykové komponenty a mnoho dalších. Je často využíván pro tvorbu rozsáhlých webových aplikačních systémů pro svoji komplexnost. Zend přichází s kompletní překladovou službou, která podporuje všechny hlavní formáty a zahrnuje oblíbené funkce, jako jsou překlady s podporou textových oblastí, které rozlišují množství. Podporuje hlavní překladové formáty, jako jsou pole PHP, CSV, Gettext, INI a mnoho dalších. Zend framework se ne-

hodí pro malé a jednoduché aplikace, protože má jedny z nejvyšších HW nároků (především na paměť) a nepatří mezi rychlé frameworky.

Český Nette framework je stavěný tak, aby byl co nejpoužitelnější a nejvstřícnější. Jde o framework, s nímž je nejen snadné, ale i zábavné pracovat. Obsahuje důmyslné zabezpečení před zranitelnostmi jako je například automatické escapování dat. Součástí frameworku je velice kvalitní ladící nástroj *Tracy*, pomocí kterého budete vždy vědět, kde vznikl problém. Formuláře jsou v Nette velice důmyslné, propracované a oproti ostatním frameworkům až nadčasové. Nette neobsahuje hotové řešení překladové služby, ale pouze rozhraní *ITranslator*, ale není problém využít již hotové řešení z jiného frameworku. Nevýhodou Nette je malá komunita v zahraničí a občasná zastaralost dokumentace, ale to se postupem času čím dál více zlepšuje.

Vhodnost použití daného frameworku závisí především na požadavcích aplikace. Pro některé aplikace jsou důležitější kvalitní formuláře, pro jiné zase automatické generování dat. Výběr konkrétního frameworku tak do značné míry závisí na zkušenostech a potřebách programátora.

V příložené tabulce (Tabulka 1 - Porovnání frameworků) jsou znázorněny jednotlivé vlastnosti popisovaných frameworků.

Tabulka 1 - Porovnání frameworků

| | CakePHP | QPHP | Symfony | Zend | Nette |
|----------------------|----------|---------|----------|----------|-------------------|
| MVC | ano | ano | ano | ano | MVP |
| ORM | ano | ne | ano | ano | ne |
| formuláře | ano | ne | ano | ano | ano |
| podpora lokalizace | ano | ne | ano | ano | ano |
| šablonovací nástroje | ne | ne | Twig | ne | Latte |
| formuláře | ano | ano | ano | ano | ano |
| výkon | vysoký | neznámý | průměrný | nízký | průměrný |
| hw nároky | průměrné | neznámé | průměrné | velké | průměrné |
| generátor kódu | ano | ne | ano | ano | ne |
| náročnost | průměrná | neznámá | průměrná | velká | malá |
| PHP | >= 5.2 | 4.5 | >= 5.3.3 | >= 5.3.3 | >= 5.3.1 |
| licence | MIT | MIT | MIT | New BSD | New BSD, GPLv2(3) |

2 Implementace multijazyčného lokalizovaného e-shopu

Pro implementaci multijazyčného lokalizovaného e-shopu jsem zvolil Nette Framework, který používám úspěšně již několik let. Na českém trhu práce je velmi často vyžadován jako jedna z podmínek pro přijetí. Tímto se mi naskytla možnost vylepšit a procvičit své znalosti, a tím zvýšit své šance na trhu práce po dokončení studia.

Nette Framework neobsahuje kompletní překladové služby, ale pouze předpřipravené rozhraní, které je využíváno pro překlady ve formulářích a šablonách. Jako nejjednodušší řešení implementace překladů by bylo použít již hotové komponenty z jiných frameworků, například *Symfony/Translation* nebo *Zend\I18n*, ale já jsem se rozhodl vytvořit svoji jednoduchou překladovou službu.

2.1 Použité technologie

2.1.1 HTML

HTML je původní jazyk, který se ještě dnes v některých případech používá k vytváření základní obsahové kostry webových stránek. Dříve jazyk HTML sloužil i k formátování vzhledu (v současnosti se k tomu kvůli zachování přístupnosti webu používají kaskádové styly, které umožňují vytvářet vzhled jako druhou, na obsahu nezávislou vrstvu).

Název HTML je zkratkou od „*HyperText Markup Language*“ – textový značkovací jazyk (viz. Obrázek 8). Slovo „*HyperText*“ zde vyjadřuje možnost vzájemně propojovat texty na základě odkazů, „*Markup*“ označuje schopnost jazyka HTML dávat významy jednotlivým blokům textu s pomocí speciálních značek nazývaných tagy a elementy (např. vypsát část textu tučně nebo ji třeba určit jako nadpis).

Jazyk HTML patří do široké rodiny značkovacích jazyků SGML. Vznikl v roce 1990 ve Švýcarsku a postupně se vyvíjel v závislosti na nejpoužívanějších prohlížečích až k současné verzi HTML 4.01, u které byl vývoj ukončen, neboť na ni navazuje modernější jazyk XHTML [10].


```

<!DOCTYPE html>
<html lang="cs">
<head>
  <meta http-equiv="content-type" content="text/html; charset=utf-8">
  <meta name="author" content="Pavel Macháň">

  <link href="design.css" rel="stylesheet" media="screen">

  <title>My page</title>
</head>
<body>

<h1>Hello world</h1>

<p>Lorem ipsum dolor sit amet</p>

</body>
</html>

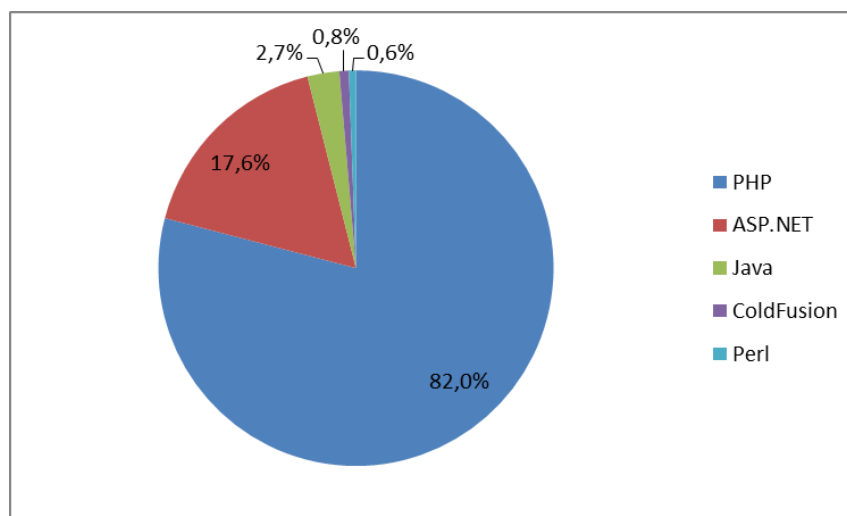
```

Obrázek 8 - Ukázka HTML

2.1.2 PHP

PHP je skriptový jazyk vytvořený právě pro web, který je umístěn na straně serveru. Do stránky HTML lze umístit kód PHP, který se vykonává pokaždé, když má být stránka zobrazena. Kód PHP je přeložen webovým serverem, který generuje HTML nebo jiný výstup. Ten je pak ze serveru zaslán (nejčastěji do webového prohlížeče) uživateli.

PHP je nejrozšířenějším skriptovacím jazykem pro web. Aktuálně 82% webových stránek pohání PHP (viz. Obrázek 9). Oblíbeným se stal především díky jednoduchosti použití, bohaté zásobě funkcí, ohebnosti jazyka a dostatečné rychlosti [11].



Obrázek 9 – Aktuální zastoupení PHP

Praktická část této bakalářské práce vyžaduje pro svůj chod alespoň PHP ve verzi 5.4, která umožňuje využívat *\$this* uvnitř anonymních funkcí, které jsou často využívány pro zpracování formulářů.

2.1.3 MySQL

MySQL je databázový systém, vytvořený švédskou firmou MySQL AB, nyní vlastněný společností Oracle Corporation. Jedná se o multiplatformní relační databázi. Komunikace s ní probíhá pomocí jazyka SQL (viz. Obrázek 10).

Je vydáván jako Open Source pod licencí GPL (GNU General Public License). Pokud licence GPL nevyhovuje, je možné zakoupit komerční licenci určenou pro OEM, ISV a VAR.

MySQL bylo od počátku optimalizováno především na rychlost, a to i za cenu některých zjednodušení (neumožňuje například vytváření sekvencí, dále je zde problém s „full-text“ indexy, které jsou do verze 5.6 podporovány pouze v úložišti MyISAM). Donedávna nepodporovalo pohledy, triggerly, a procedury.

```
SELECT `text_translation`.*, `text`.* FROM `text`  
LEFT JOIN `text_translation` USING(`id_text`)  
WHERE `id_language` = '2' AND (`text_translation`.`slug` = 'contact');
```

Obrázek 10 - Ukázka SQL dotazu

2.1.4 Composer

Composer je nástroj pro správu závislostí v PHP. Umožňuje deklarovat závislosti projektu na jiné knihovny či balíky. Composer je velice jednoduchý a intuitivní. Veškerá konfigurace probíhá v souboru *composer.json* (viz. Obrázek 11). Jak již z přípony souboru vyplývá, konfigurace se zapisuje ve formátu JSON.

Díky composeru není třeba závislosti ručně kopírovat z jiných míst, a tím odpadá problém s vyhledáváním jejich aktualizací. Composer umožňuje jednoduchým příkazem zaktualizovat veškeré závislosti. Dále se stará i o jejich automatické načtení.

Závislosti lze načítat z vlastních repositářů (CVS), nebo lze využít hlavní composer repositář dostupný na adrese <https://packagist.org/>, který obsahuje tisíce knihoven a balíků.

```

{
  "authors": [
    {
      "name": "Pavel Macháň",
      "email": "machan@hostbox.cz",
      "homepage": "http://www.hostbox.cz"
    }
  ],
  "config": {
    "vendor-dir": "Website/src/core/vendor"
  },
  "require": {
    "hostbox/lib-core": "@dev",
    "hostbox/lib-model": "@dev",
    "hostbox/cms-core": "@dev",
    "vojtech-dobes/nette-multi-authenticator": "v1.0.4",
    "kdyby/facebook": "v1.0.1",
    "tharos/leanmapper": "dev-develop"
  },
  "require-dev": {
    "nette/tester": "~1.0"
  }
}

```

Obrázek 11 - Ukázka zápisu konfiguračního souboru

2.1.5 LESS

LESS je CSS pre-processor, který rozšiřuje základní jazyk CSS (viz. Obrázek 12). Přidává možnost práce s proměnnými, lze definovat funkce a umožňuje mnoho dalších technik. LESS je velice přehledný a oproti klasickému CSS lehce znovupoužitelný. Často se využívá pro CSS frameworky jako je například Twitter Bootstrap (také využit v této aplikaci).

LESS může být zpracováván knihovnou JS za běhu aplikace nebo může být překompilován do klasického CSS. Je doporučeno používat překompilování do CSS, kde se lze vyhnout nadbytečné režii zpracováváním pomocí JS. Pro kompilaci lze využít hned několik programů například: SimpLESS, WinLESS, Koala a mnoho dalších. Některá vývojová prostředí podporují kompilaci LESS souborů přímo v základu či po doinstalování doplňků.

```

// LESS
#bundle {
  .button {
    display: block;
    border: 1px solid black;
    background-color: grey;
    &:hover {
      background-color: white
    }
  }
}

// CSS
#bundle .button {
  display: block;
  border: 1px solid black;
  background-color: grey;
}
#bundle .button:hover {
  background-color: #ffffff;
}

```

Obrázek 12 - LESS vs. CSS

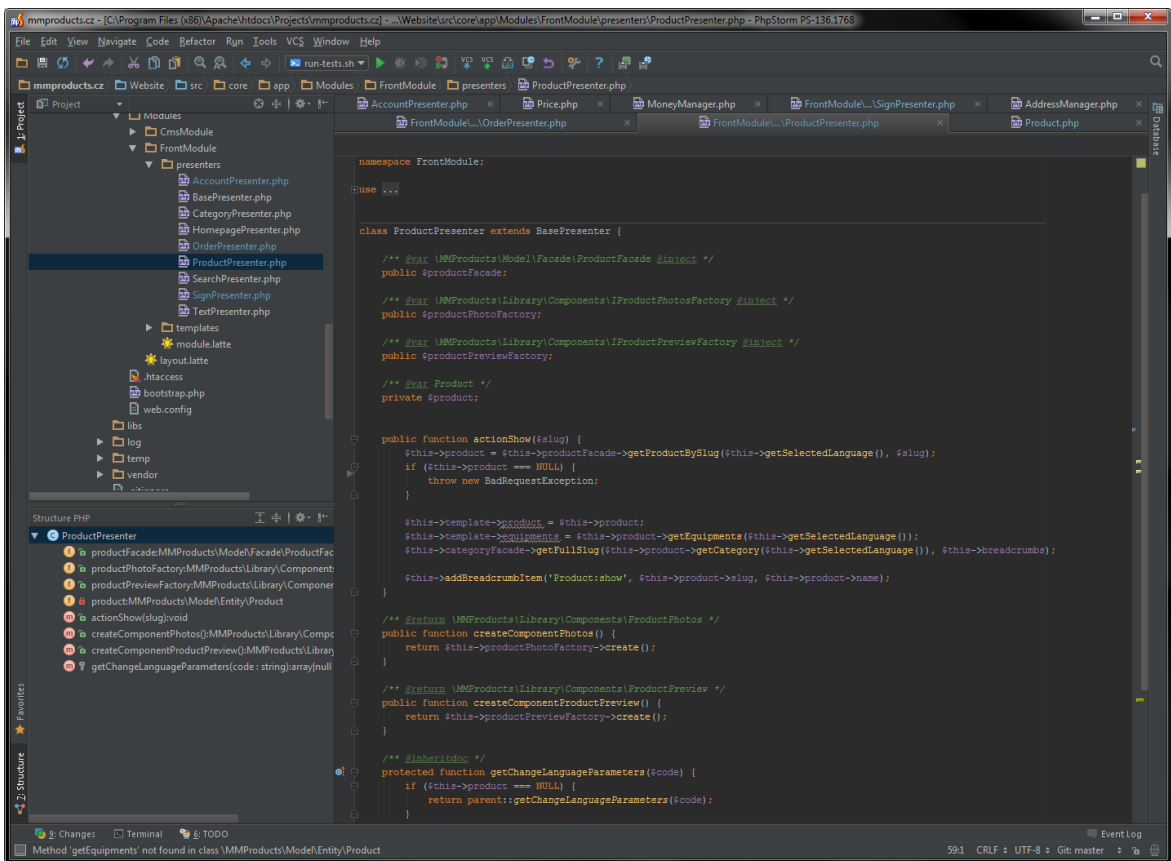
2.1.6 IDE: PhpStorm

PhpStorm je komerční, multiplatformní IDE pro PHP založené na platformě IntelliJ IDEA, které poskytuje editor pro PHP, HTML a JavaScript. Umožňuje analýzu kódu on-the-fly, prevenci chyb a automatizovaného refaktorování pro PHP a JavaScript. Podporuje PHP 5.3, 5.4 a 5.5, včetně generátorů, jmenných prostorů, „closure“, „trait“ a zkráceného zápisu polí. Obsahuje plnohodnotný editor SQL s výsledky požadovaných dotazů a mnoho dalších funkcí, které usnadňují vývoj webových aplikací (viz. Obrázek 13).

Licence je trvalá a zahrnuje jeden rok aktualizací produktu od data zakoupení, včetně hlavních verzí a upgradů. Cena licence se nyní pohybuje od 26 € do 179 € dle typu licence. Obnovení aktualizací po roce využívání stojí zhruba polovinu ceny licence klasické.

Dříve jsem používal NetBeans od firmy Oracle, který je zdarma vydávaný pod licencí GPL2, ale postupem času nedostačoval mým potřebám. NetBeans je oproti PhpStorm velice pomalý a neohrabaný.

Jakmile jsem PhpStorm otestoval pomocí 30 denní zkušební verze, která je zdarma, ihned jsem si ho oblíbil a druhý den následně i zakoupil. Nyní používám PhpStorm téměř rok a hodlám si stávající licenci po jejím vypršení prodloužit. Každému, kdo se zabývá vývojem webových aplikací PhpStorm doporučuji.



Obrázek 13 - IDE: PhpStorm

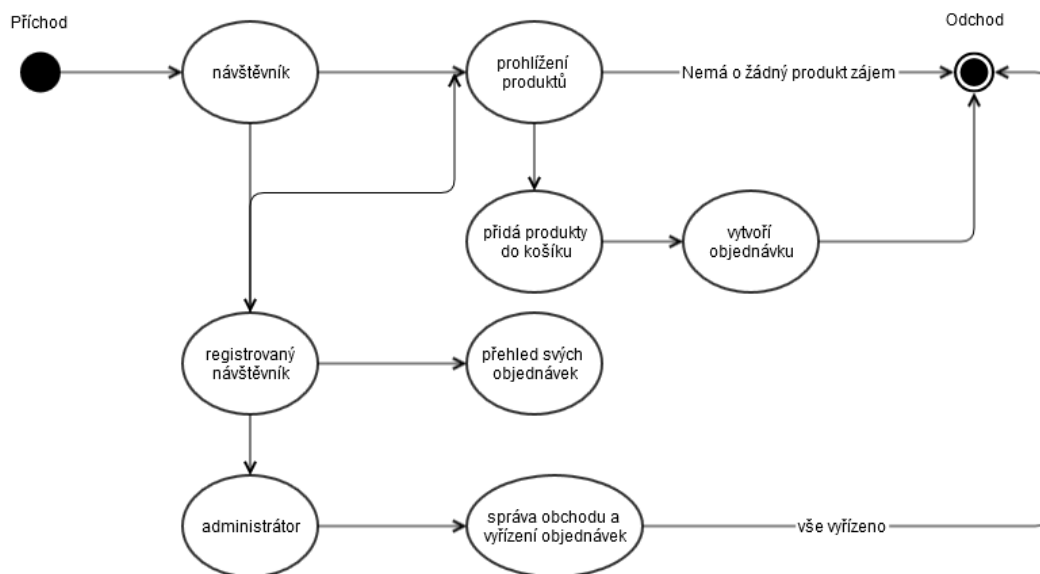
2.2 Architektura aplikace

2.2.1 Rich Picture

Rich Picture slouží k primitivnímu znázornění logiky navrhované aplikace. Zaznamenává vnitřní i vnější vlivy uživatele z pohledu aplikace (viz. Obrázek 14).

Popis logiky navržené aplikace

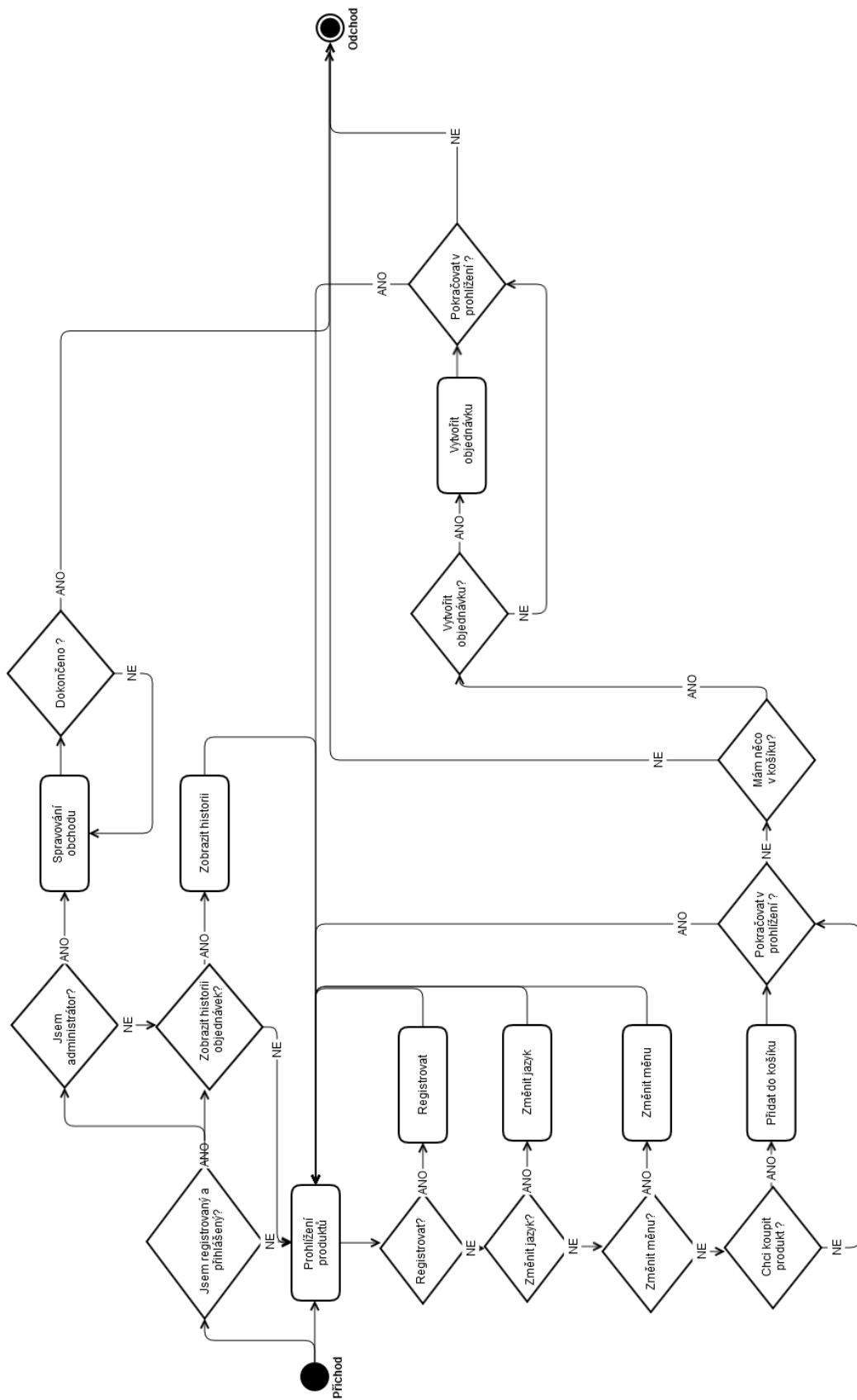
Návštěvník přijde do navrhované aplikace, kde si může prohlédnout produkty a následně je vložit do nákupního košíku. Po přidání požadovaných produktů do nákupního košíku je návštěvníkům umožněno vytvořit objednávku. Pokud je návštěvník zaregistrován, je mu povoleno zobrazit své již vytvořené objednávky. Administrátor má za úkol správu obchodu a vyřizování objednávek. Po dokončení plánovaných akcí návštěvník odchází z aplikace (zavře prohlížeč nebo panel prohlížeče).



Obrázek 14 - Rich Picture

2.2.2 UML Activity Diagram

UML Activity Diagram je jeden z diagramů UML, které popisují chování. Tento diagram je využíván pro modelování procedurální logiky, procesů a zachycení workflow. Každý proces v diagramu aktivit je reprezentován sekvencí jednotlivých kroků, které jsou v modelu zakresleny jako akce, nebo vnořené aktivity (viz. Obrázek 15).



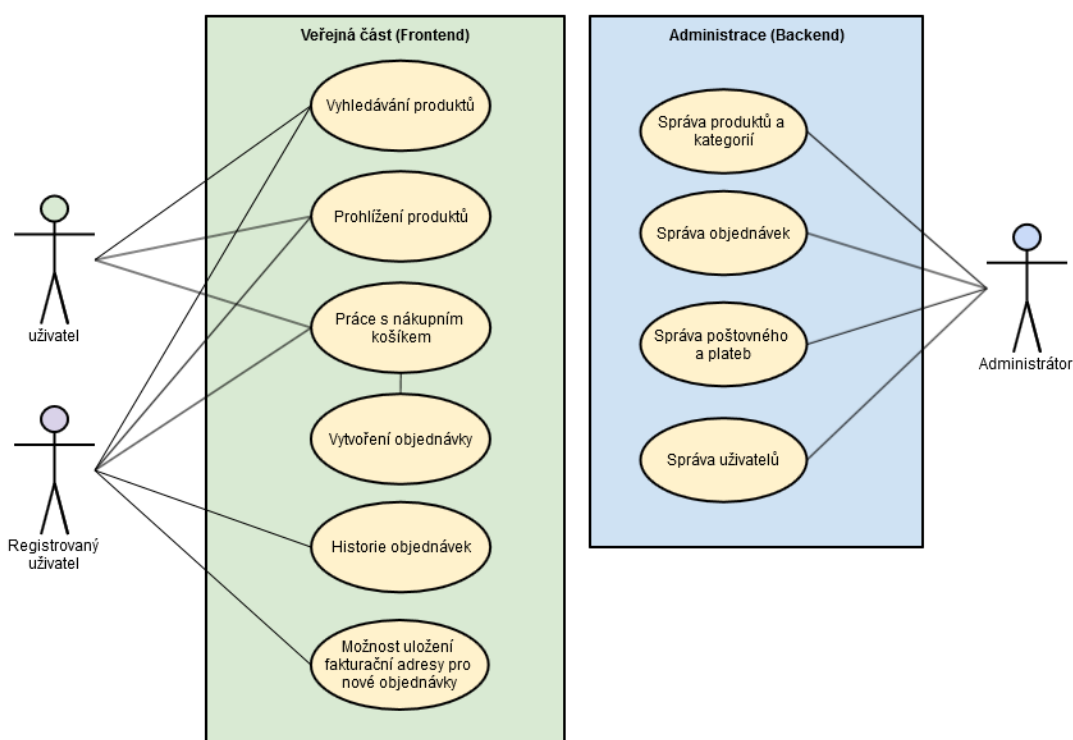
Obrázek 15 - UML Activity Diagram

2.2.3 Use Case Diagram

Navrhovaná aplikace obsahuje tři uživatelské role. První role je běžný uživatel, který nemá v aplikaci registraci a má jen základní oprávnění. Je mu umožněno vyhledávat produkty, prohlížet jednotlivé produkty a spravovat nákupní košík. Běžný uživatel může také vytvořit objednávku (viz. Obrázek 16).

Druhá role je registrovaný uživatel. Vlastní veškeré oprávnění běžného uživatele a navíc má povoleno zobrazit historii svých již vytvořených objednávek. Při registraci musí vyplnit fakturační údaje, které se budou automaticky vyplňovat při vytváření objednávek (oproti běžnému uživateli je nemusí neustále znovu a znovu vyplňovat).

Třetí role je administrátor. Má veškeré práva a je mu umožněn přístup do administrace aplikace, kde spravuje produkty a jejich kategorie, objednávky, poštovné, platby a uživatele.



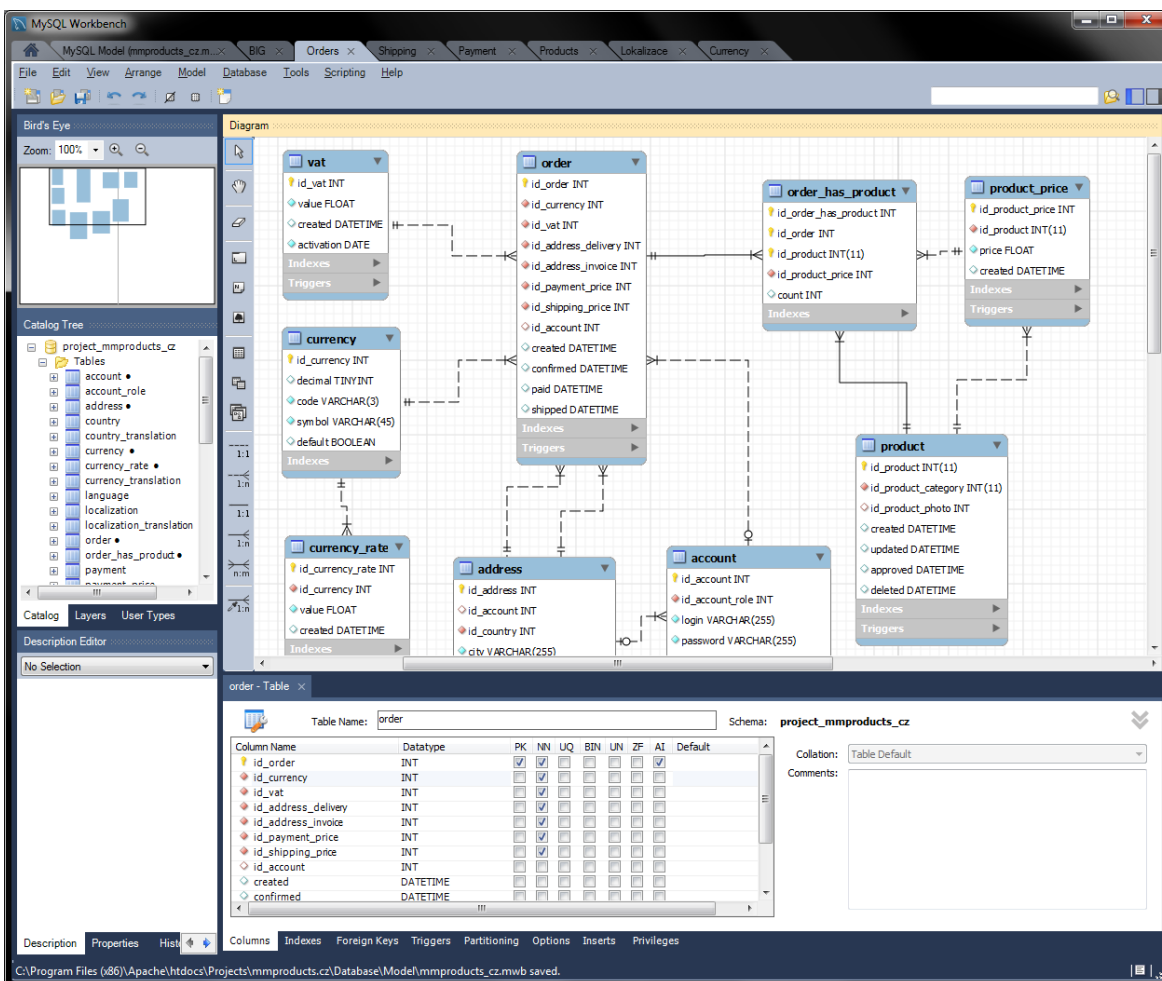
Obrázek 16 - Use Case Diagram

2.3 Návrh databázového modelu

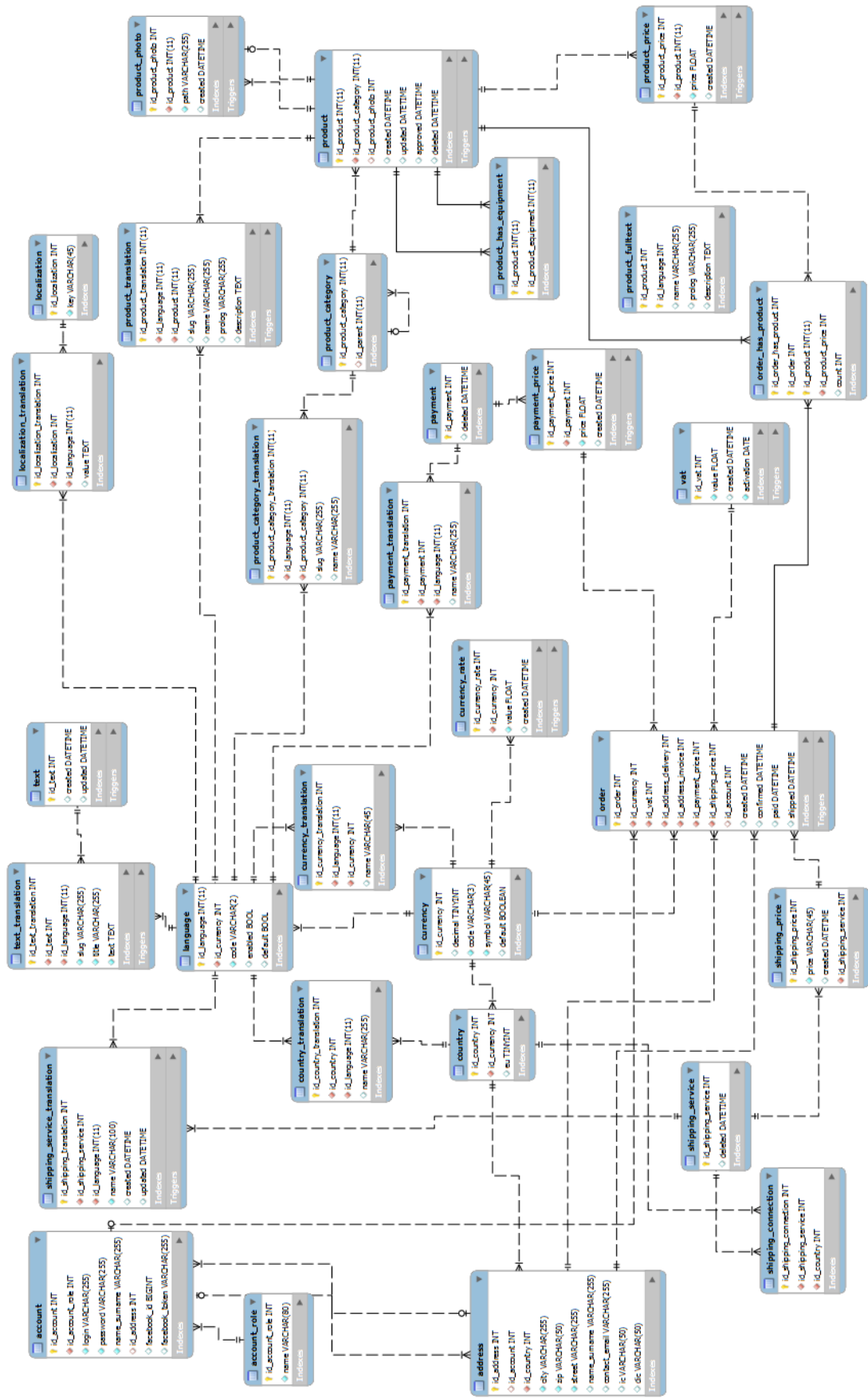
Návrh databázového modelu je jedna z nejdůležitějších částí při návrhu celé aplikace. Musí pokrývat požadavky aplikace a reflektovat i možné budoucí poslání. Při špatně navrhnutém modelu mohou například vznikat velké problémy ve výkonosti aplikace, a špatné budoucí rozšiřitelnosti (viz. Obrázek 18).

Pro návrh databázového modelu jsem použil program MySQL Workbench Community Edition ve verzi 6.0 od firmy Oracle, který se velice dobře používá, má intuitivní ovládání a zvládá i pokročilejší funkce (viz. Obrázek 17). MySQL Workbench je pro nekomerční použití zdarma a je vydáván pod Open Source licencí GPL.

MySQL Workbench používám již 2 roky a velice mě vyhovuje. Jedinou nevýhodou jsou občasné pády aplikace. Zkoušel jsem používat Toad Data Modeler, ale absolutně nevyhovoval mým potřebám.



Obrázek 17 - Ukázka MySQL Workbench



Obrázek 18 - ER diagram

2.3.1 Lokalizace

Lokalizace popisků, tlačítek a formulářů je řešena pomocí tří tabulek a to *language*, *localization* a *localization_translation* (viz. Obrázek 20).

Tabulka *language*

Tato tabulka obsahuje seznam jazyků, které aplikace podporuje. Sloupec *default* označuje který jazyk je výchozí.

Tabulka *localization*

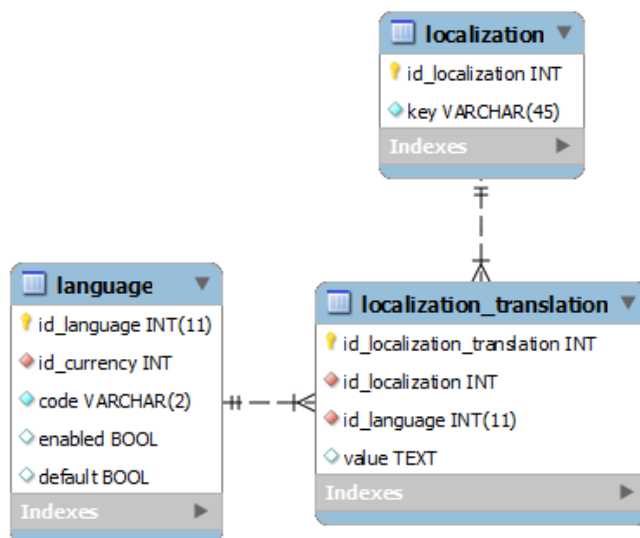
Tabulka *localization* je nositelem klíče daného překladu. Pomocí klíče je v aplikaci rozeznáváno, jaký překlad je z databáze požadován (viz. Obrázek 19).

```
$this->translator->translate('form.submit.save');
```

Obrázek 19 - Ukázka použití lokalizačního klíče

Tabulka *localization_translation*

V tabulce jsou obsaženy textové překlady pro daný lokalizační klíč a jazyk.



Obrázek 20 - ER diagram: lokalizace

2.3.2 Produkty

Nepostradatelnou částí datového modelu aplikace jsou produkty. Produkty mají několik tabulek, kde každá má svoji specifickou úlohu a funkci (viz. Obrázek 21).

Tabulka produkty

Každý produkt musí mít přiřazenou kategorii a může mít nastavenou hlavní fotografii, která se zobrazuje v náhledech produktu. Dále každý produkt obsahuje časové informace o vytvoření, upravení, schválení a smazání.

Při mazání produktu v administraci není jeho záznam smazán z tabulky, ale je pouze nastaven aktuální čas do sloupce *deleted* a to z důvodu zachování dat pro již provedené objednávky, které obsahují daný produkt. Pokud by se z tabulky produkt smazal, nebylo by možné udržovat historii objednávek.

Tabulka product_translation

Tabulka *product_translation* obsahuje textová data pro každý produkt. Tabulky využívají vazbu 1:N, kde každý produkt může obsahovat několik překladů. Je-li v aplikaci nastaven jazyk, pro který není vytvořen překlad, aplikace produkt nezobrazí. Tím je umožněno nabízet konkrétní produkt pouze ve vybrané jazykové mutaci.

Tabulka product_price

Produkty mají uložené ceny v tabulce *product_price*. Produkty využívají vazby 1:N, kde každý produkt může mít několik cen. Pro zobrazení aktuální ceny produktu je vždy použit poslední záznam z tabulky *product_price* pro daný produkt. Tato tabulka má za účel udržovat historii cen produktů. V případě, že by se cena produktu ukládala do tabulky *product*, nebylo by možné ji změnit, aniž by ovlivnila stávající objednávky.

Tabulka product_fulltext

MySQL úložiště InnoDB nepodporuje do verze 5.6 fulltext indexaci, která je v aplikaci využívána pro vyhledávání produktů. Z tohoto důvodu byla vytvořena tabulka *product_fulltext* s nastaveným úložištěm MyISAM, která tyto indexy podporuje. Nastavit úložiště MyISAM na tabulku *product_translation* nelze z důvodu absence cizích klíčů na tomto úložišti. Cizí klíče jsou potřeba pro vazbu mezi tabulkami *product_translation* a *product*.

Data jsou z překladové tabulky *product_translation* duplikována do tabulky *product_fulltext* pomocí procedury *product_fulltext_update*. Po vyhledání požadovaných produktů z tabulky *product_fulltext* jsou data ze sloupce *id_product* použita pro výběr původních data z tabulky *product_translation*.

Tabulka *product_photo*

Produkty mají cesty k fotografiím uloženy v tabulce *product_photo*. Tabulky využívají vazbu 1:N, kde každý produkt může mít hned několik fotografií. Pokud produkt nemá žádnou fotografii, je místo ní zobrazena oranžová ikonka fotoaparátu, která uživateli oznamuje neexistující fotografii.

Tabulka *product_photo_language_blacklist*

Z důvodu možnosti zakázat zobrazení určité fotografie pro vybraný jazyk musela být zavedena tabulka *product_photo_language_blacklist*, která tuto funkci umožňuje.

Tabulka *product_has_equipment*

Díky této tabulce je možné přiřadit každému produktu doporučené příslušenství. Pro tyto účely je použita identifikující vazba N:M mezi tabulkou *product* a *product_has_equipment*.

Uživatel u každého produktu vidí, jaké příslušenství se k danému produktu hodí a nemusí ztrácet čas jeho hledáním.

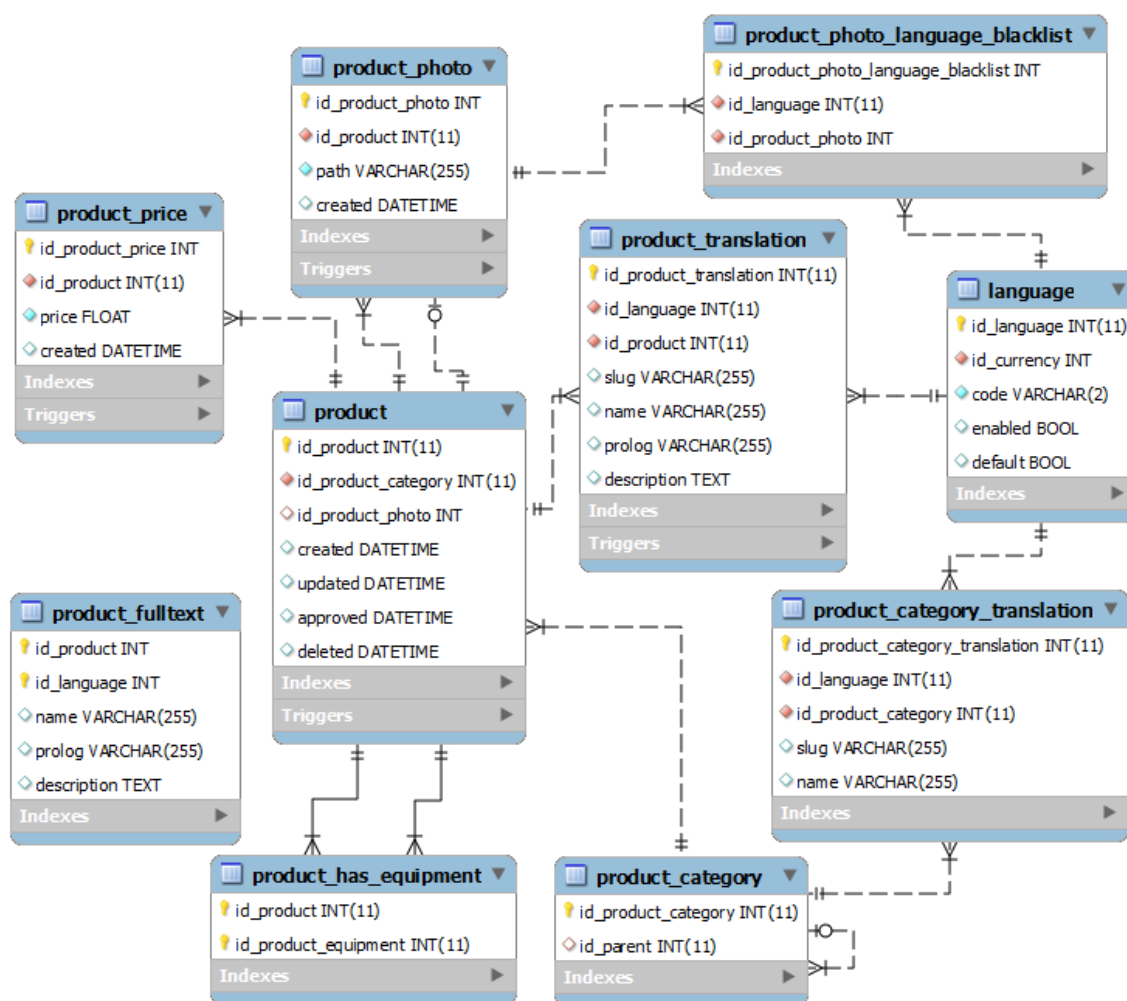
Tabulka *product_category*

Pro kategorie je použita stromová struktura, která je řešena rekurzivně procházeným seznamem. Rekurzivně procházený seznam obsahuje v tabulce sloupec *id_parent*, který udává její rodičovskou kategorii. Nejvyšší bod stromu nemá hodnotu rodiče (*id_parent* je NULL.)

Toto řešení je velice jednoduché, ale při velkém zanoření kategorií náročné. Pro získání všech potomků se musí použít rekurze, ale hlavně – pro každou úroveň, resp. každou položku, se musí spustit jeden dotaz SQL. Zvýšená náročnost se z části eliminuje zavedením mezipaměti (cache) na úrovni aplikace.

Tabulka *product_category_translation*

Tabulka *product_category_translation* slouží jako překladová pro tabulku *product_category* a obsahuje textová data pro jednotlivé kategorie. Tabulky využívají vazbu 1:N, kde každá kategorie může obsahovat několik překladů. Je-li v aplikaci nastaven jazyk, pro který není vytvořen překlad, aplikace kategorii nezobrazí. Tím je umožněno nabízet konkrétní produkt pouze ve vybrané jazykové mutaci.



Obrázek 21 - ER diagram: produkty

2.3.3 Měny

Má-li internetový obchod zasílat zboží do různých zemí, které nemají jednotnou měnu, je nutné podporovat více měn (viz. Obrázek 22).

Tabulka currency

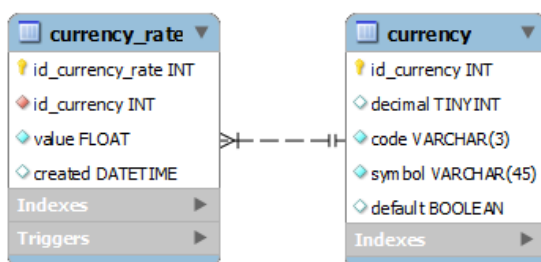
Tabulka currency obsahuje informace o dané měně a to kód, symbol a počet desetinných míst měny. Sloupec *default* označuje výchozí měnu obchodu.

Pro kód měny se používá normovaný mezinárodní standart ISO 4217, který obsahuje tři znaky. Je složený z označení země dle kódu ISO 3166-1 alpha-2 a počátečního písmena z názvu měny [12].

Symbol měny je grafický symbol, který je umístěn za hodnotou částky.

Tabulka `currency_rate`

Aby nebylo zadávání cen produktů složité a zdlouhavé, nezadávají se k produktu ceny pro každou měnu zvlášť, ale pouze částka měny výchozí. Pro přepočítání částky slouží tabulka `currency_rate`, která obsahuje sloupec `value`, kde je u každé měny uložen poměr vůči výchozí měně. Výsledná cena produktu dle vybrané měny se vypočítá pomocí vynásobení výchozí ceny a poměru vybrané měny vůči výchozí měně. Výsledná částka se zaokrouhlí dle hodnoty sloupce `decimal` v tabulce `currency`.



Obrázek 22 - ER diagram: měny

2.3.4 Účty

Aby bylo možné uživatelům umožnit zobrazení historie objednávek, je nutné je nějakým způsobem od sebe rozeznat. Za tímto účelem jsou v aplikaci zavedeny účty, díky kterým rozeznáme jednotlivé uživatele (viz. Obrázek 23).

Tabulka `account`

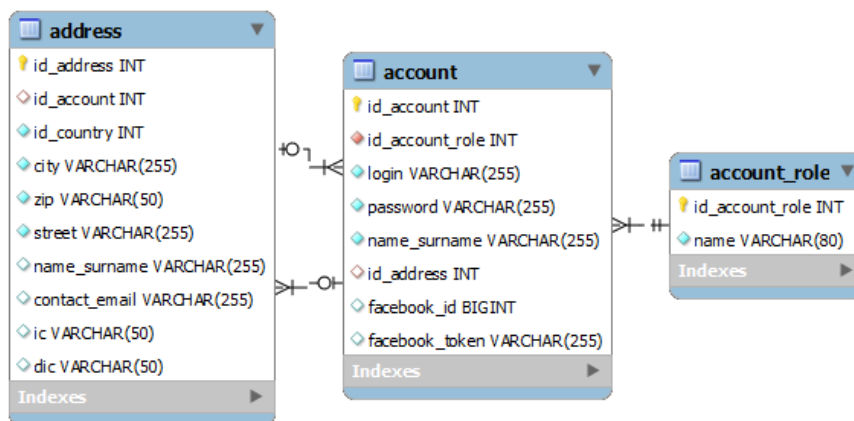
Tabulka obsahuje záznamy jednotlivých registrovaných uživatelů, kteří se mohou přihlásit pomocí přihlašovacích údajů, které vyplnili při registraci nebo mohou využít přihlašování pomocí sociální služby Facebook.

Tabulka `account_role`

Každý účet má přiřazenu roli, podle které uživatel dostává oprávnění. Aplikace využívá tři role a to běžný uživatel, registrovaný uživatel a administrátor (viz. kap. 2.2.3).

Tabulka `address`

Tabulka `address` uchovává adresní a fakturační informace k danému účtu, které jsou využívány pro objednávky.



Obrázek 23 - ER diagram: účty

2.3.5 Platební služby

Každý obchod potřebuje umožnit svým zákazníkům nějakým způsobem zaplatit za objednávky, a proto je nutné definovat platební služby. Z nich si nakupující zákazník jednu vybere a tou zaplatí (viz. Obrázek 24).

Tabulka payment

Tato tabulka obsahuje seznam platebních služeb, které aplikace podporuje.

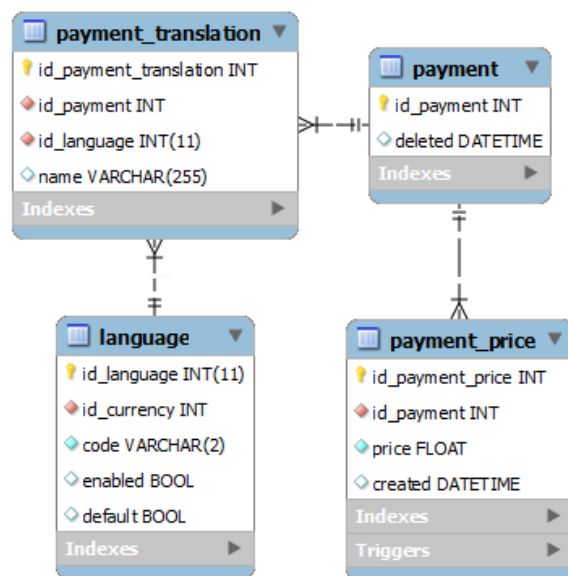
Při mazání platební služby v administraci není její záznam smazán z tabulky, ale je pouze nastaven aktuální čas do sloupce *deleted* a to z důvodu zachování dat pro již provedené objednávky, které obsahují danou platební službu. Pokud by se z tabulky platební služba smazala, nebylo by možné udržovat historii objednávek z důvodu neexistující služby.

Tabulka payment_translation

Tabulka *payment_translation* slouží jako překladová pro tabulku *payment* a obsahuje textová data pro jednotlivé služby. Tabulky využívají vazbu 1:N, kde každá platební služba může obsahovat několik překladů.

Tabulka payment_price

Platební služby mají uložené ceny v tabulce *payment_price*. Produkty využívají vazby 1:N, kde každá platební služba může mít několik cen. Pro zobrazení aktuální ceny služby je vždy použit poslední záznam z tabulky *payment_price* pro danou službu. Tato tabulka má za účel udržovat historii cen platebních služeb. V případě, že by se cena služby ukládala do tabulky *payment*, nebylo by ji možné změnit, aniž by se ovlivnily stávající objednávky, které by obsahovaly upravovanou službu.



Obrázek 24 - ER diagram: platební služby

2.3.6 Dopravní služby

Při vytváření objednávky musí zákazník vybrat dopravní službu, pomocí které internetový obchod doručí objednané zboží. Internetový obchod pro celosvětový prodej musí podporovat různé služby pro různé státy (viz. Obrázek 25).

Tabulka `shipping_service`

Tato tabulka obsahuje seznam dopravních služeb, které aplikace podporuje.

Při mazání dopravní služby v administraci není její záznam smazán z tabulky, ale je pouze nastaven aktuální čas do sloupce `deleted` a to z důvodu zachování dat pro již provedené objednávky, které obsahují danou dopravní službu. Pokud by se z tabulky dopravní služba smazala, nebylo by možné udržovat historii objednávek z důvodu neexistující služby.

Tabulka `shippingt_service_translation`

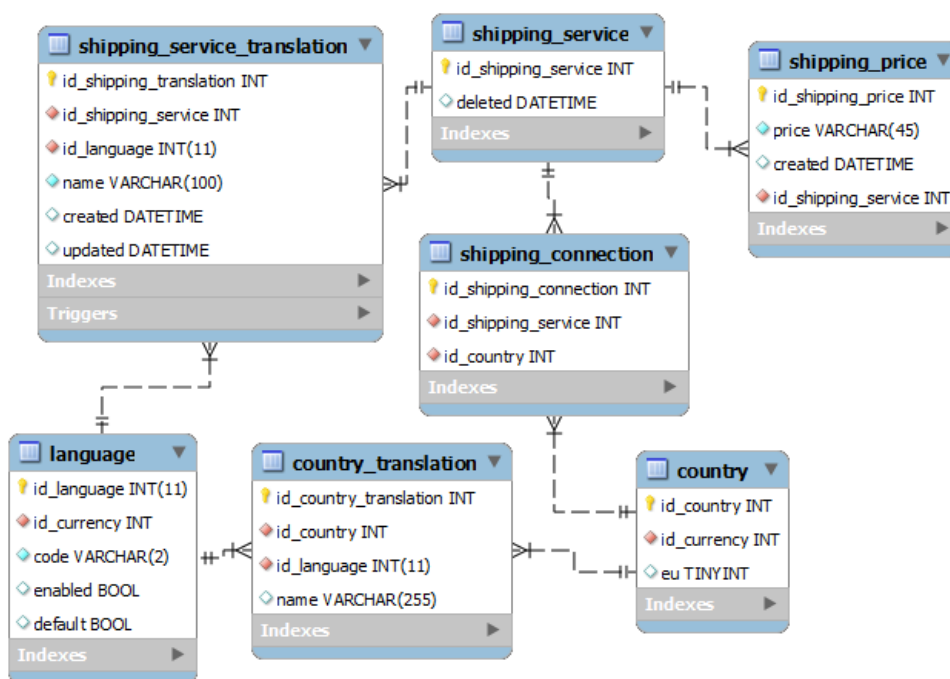
Tabulka `shippingt_service_translation` slouží jako překladová pro tabulku `shipping_service` a obsahuje textová data pro jednotlivé služby. Tabulky využívají vazbu 1:N, kde každá služba může obsahovat několik překladů.

Tabulka `shipping_service_connection`

Pro navázání dopravních služeb na určité státy byla vytvořena tabulka `shipping_service_connection`, kde každá dopravní služba může obsahovat hned několik států. Tímto je umožněno seskupit hned několik států pod jednu dopravní službu.

Tabulka `shipping_price`

Dopravní služby mají uložené ceny v tabulce `shipping_price`. Služby využívají vazby 1:N, kde každá dopravní služba může mít několik cen. Pro zobrazení aktuální ceny služby je vždy použit poslední záznam z tabulky `shipping_price` pro danou službu. Tato tabulka má za účel udržovat historii cen dopravních služeb. V případě, že by se cena služby ukládala do tabulky `shipping_service`, nebylo by možné ji změnit, aniž by se ovlivnily stávající objednávky, které by obsahovaly upravovanou službu.



Obrázek 25 - ER diagram: dopravní služby

2.3.7 DPH

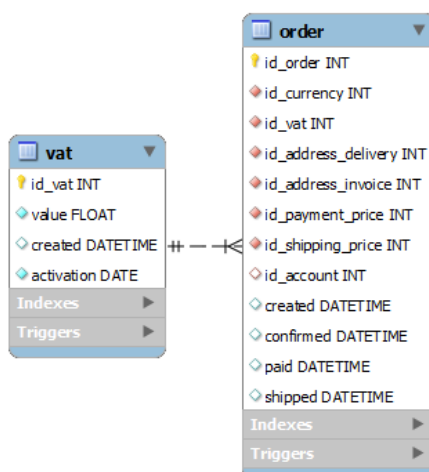
Každá objednávka musí zaznamenávat jaké DPH je fakturováno. Pokud je ve fakturační adrese uvedeno daňové identifikační číslo (DIČ) je fakturováno bez DPH pro ČR a státy patřící do Evropské unie. V jiných případech je fakturováno s DPH (viz. Obrázek 26).

Tabulka vat

Tabulka *vat* obsahuje hodnotu DPH ve sloupci *value* nikoliv jako procentuální hodnotu ale jako desetinné číslo s přičtenou jedničkou. Pro příklad DPH 21% je v tabulce vedeno jako 1,21. Pro výpočet ceny produktu s DPH stačí tuto hodnotu vynásobit cenou produktu.

Tento způsob ukládání hodnoty DPH je výhodný z důvodu ušetření neustálého převádění procentuální hodnoty DPH na každém místě v aplikaci, kde je požadován výpočet cen s DPH. Přesně opačný převod z procentuální hodnoty na tento tvar se provede pouze jednou při zadávání DPH v administraci.

Dále tabulka *vat* obsahuje sloupec *created* nesoucí časovou informaci o vytvoření záznamu. Sloupec *activation* udává datum od kdy je jednotlivé DPH aktivní.



Obrázek 26 - ER diagram: DPH

2.3.8 Objednávky

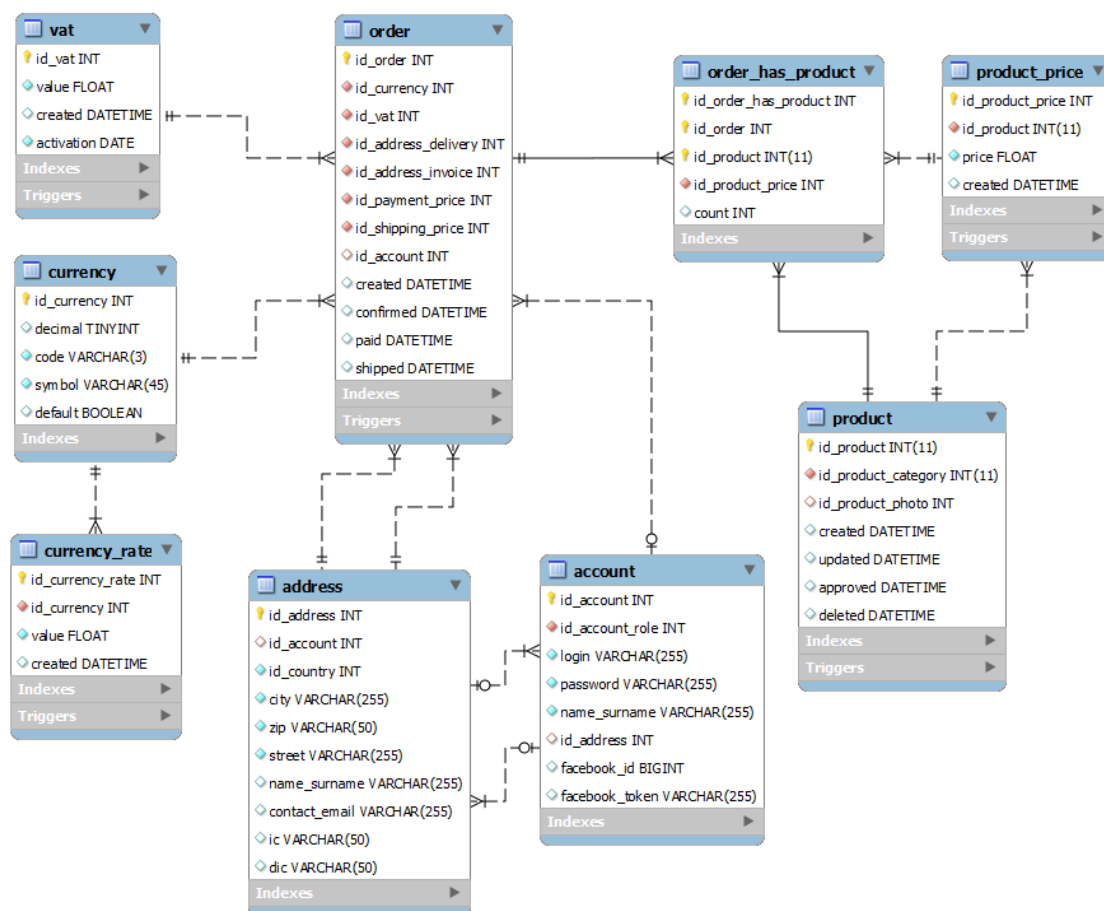
Objednávku může vytvořit registrovaný i neregistrovaný uživatel. Registrovaný uživatel je ale zvýhodněn předpřipravenou fakturační adresou, kterou si nastavil při registraci nebo ve správě účtu. Registrace pro jednorázový nákup není nutná, dokonce by kupujícího zdržovala, a tím by mohla odradit od nákupu (viz. Obrázek 27).

Tabulka order

Tabulka určená pro objednávky obsahuje informace o poštovní a fakturační adrese, o ceně platební a dopravní služby, daňovém zatížení, měně. Dále obsahuje časové informace ohledně vytvoření, potvrzení a zaplacení objednávky a jejím vyexpedování.

Tabulka order_has_product

K zaznamenání produktů, jejich množství a cen, které jsou obsažené v objednávce slouží tabulka *order_has_product*. Tabulka *order* má s tabulkou *order_has_product* identifikující vazbu 1:N, kde objednávka musí obsahovat alespoň jeden produkt.



Obrázek 27 - ER diagram: objednávky

2.3.9 Texty

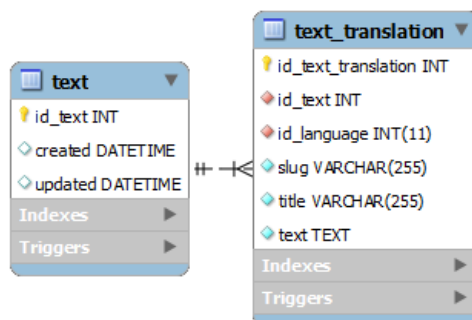
Texty jsou využívány pro obsahovou část, jako je například sekce kontakt, kde uživatel nalezne kontaktní informace obchodu (viz. Obrázek 28).

Tabulka text

V této tabulce jsou pouze časové informace o vytvoření a upravení textu.

Tabulka `text_translation`

Tabulka `text_translation` slouží jako překladová pro tabulku `text` a obsahuje textová data pro jednotlivé texty. Tabulky využívají vazbu 1:N, kde každý text může obsahovat několik překladů.



Obrázek 28 - ER diagram: texty

2.3.10 Ukázka vybraných SQL dotazů

SELECT

Pro vypsání kategorie produktů je použito vnitřní spojení tabulek `product_category` a `product_category_translation`, kde `product_category_translation` obsahuje lokalizaci kategorie. Požadovanou kategorii lze získat pomocí unikátního identifikačního čísla nastaveného jazyka a názvu kategorie, který aplikace získá z URL. Název kategorie musí být pro každý záznam pro daný jazyk unikátní (viz. Obrázek 29).

```
SELECT `product_category_translation`.*, `product_category`.*
FROM `product_category`
LEFT JOIN `product_category_translation`
  ON `product_category_translation`.`id_product_category` = `product_category`.`id_product_category`
WHERE `id_language` = '1' AND (`product_category_translation`.`slug` = 'modely')
LIMIT 1;
```

Obrázek 29 - SQL: SELECT pro zobrazení kategorie

UPDATE

Adresa je upravována příkazem UPDATE, kde jsou nastaveny sloupce na požadované hodnoty a pomocí restriktce na sloupci `id_address` je měněn jen požadovaný záznam (viz. Obrázek 30).

```
UPDATE `address` SET `id_country`= 59, `city`='Náchod', `street`='Kollárova 203',
`zip`='54701', `contact_email`='czeifel@gmail.com', `ic`='123456', `dic`='CZ123456789'
WHERE `id_address` = 84;
```

Obrázek 30 - SQL: UPDATE pro upravení adresy

DELETE

Smazání účtu je provedeno příkazem DELETE s požadovaným *id_account*. Pokud by nebyla nastavena restrikce na požadované unikátní identifikační číslo, byly by smazány veškeré účty (viz. Obrázek 31).

```
DELETE FROM `account` WHERE (`id_account`='10') LIMIT 1;
```

Obrázek 31 - SQL: DELETE pro mazání účtu

2.3.11 Ukázka vybraných procedur a triggerů

Aplikace využívá procedury pro možnost vyhledávání produktů. Aplikace je postavena na starší verzi MySQL 5.6, která nepodporuje (podpora od verze 5.7) fulltext indexaci nad úložištěm InnoDB.

Procedura *product_fulltext_update* duplikuje data produktů do tabulky *product_fulltext*, která má nastavené úložiště MyISAM a tím je zajištěna podpora fulltext indexů (viz. Obrázek 32).

```
CREATE PROCEDURE `product_fulltext_update` (IN `_id_product` int(11), IN `_id_language` int(11), IN `_name` TEXT,
IN `_prolog` TEXT, IN `_description` TEXT)
BEGIN
INSERT INTO `product_fulltext` (`id_product`, `id_language`, `name`, `prolog`, `description`)
VALUES (`_id_product`, `_id_language`, `_name`, `_prolog`, `_description`)
ON DUPLICATE KEY UPDATE `name` = VALUES(`name`), `prolog` = VALUES(`prolog`), `description` = VALUES(`description`);
END
```

Obrázek 32 - SQL: Procedura pro duplikaci lokalizace produktů

Procedura *product_fulltext_update* je spouštěna pomocí triggerů nad tabulkou *product_translation* (viz. Obrázek 33).

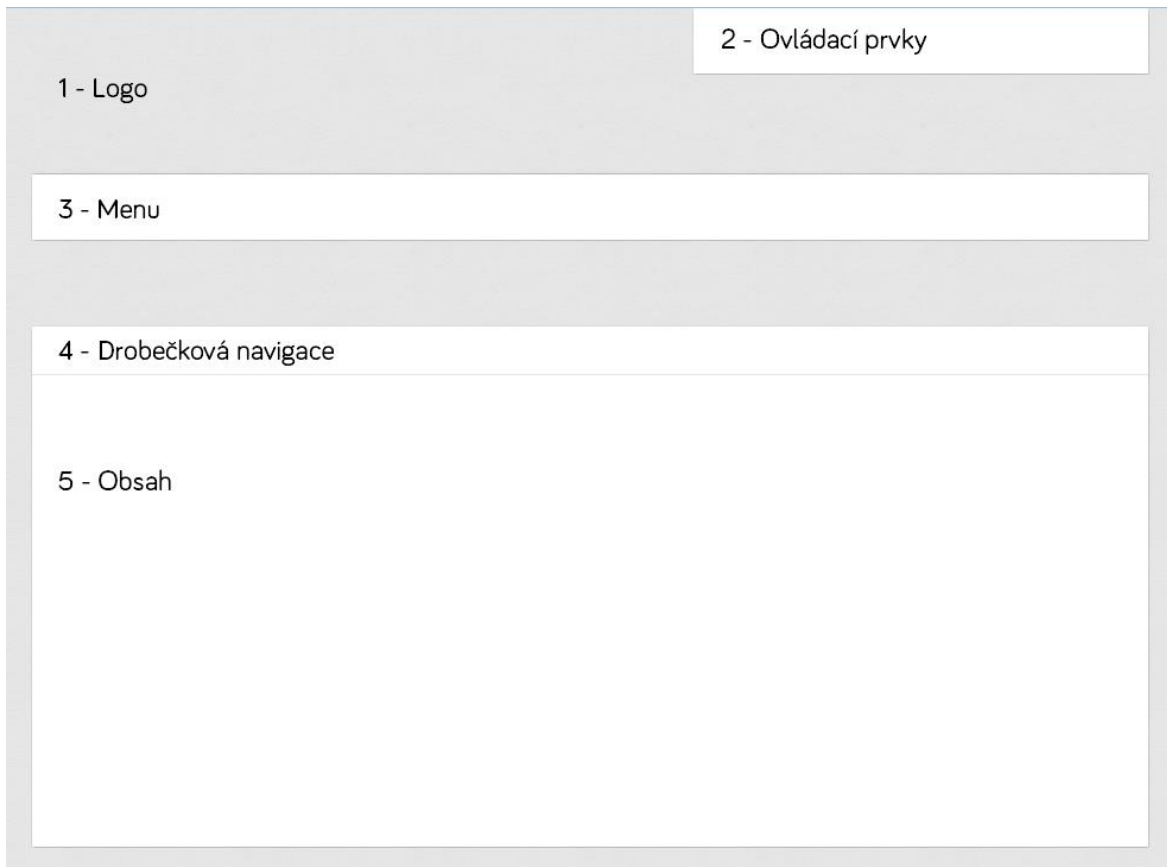
```
CREATE TRIGGER `product_translation_AINS` AFTER INSERT ON `product_translation` FOR EACH ROW
IF @disable_triggers IS NULL THEN
CALL product_fulltext_update(NEW.`id_product`, NEW.`id_language`, NEW.`name`, NEW.`prolog`, NEW.`description`);
END IF
```

Obrázek 33 - SQL: Trigger pro volání procedury

2.4 Vzhled aplikace

2.4.1 Wireframe

Cílem Wireframe je definovat rozložení jednotlivých funkčních prvků na stránce. Nejedná se o grafický návrh, neboť je tvořen pouze pomocí čar a textu.



Obrázek 34 – Wireframe

Aplikace je rozdělena na 5 funkčních částí, kde každá část plní svoji specifickou funkci (viz. Obrázek 34).

První část je určena pouze pro aktivní logo obchodu, které po kliknutí přesměruje návštěvníka na hlavní stránku obchodu. V praxi se přesměrovávání pomocí loga velice často využívá a návštěvníci stránek s tím více, či méně počítají.

Druhá část obsahuje ovládací prvky, které mají za úkol informovat o nastaveném jazyku, měně a také uživateli umožňují je přenastavit. Dále je v této části obsažen nákupní košík, který zobrazuje aktuální součtovou cenu vloženého zboží. Cena je zobrazována bez DPH. Tato část obsahuje i tlačítka pro přihlášení nebo registraci uživatele. Pokud je uživatel již zaregistrován a přihlášen je mu namísto těchto tlačítek zobrazováno tlačítko pro odhlášení a tlačítko pro přechod do svého účtu.

Třetí část obsahuje hlavní menu aplikace, kde jsou zobrazeny hlavní kategorie zboží. V pravé části menu je také pole pro vyhledávání, které po odeslání přesměruje uživatele do sekce vyhledávání a zobrazí mu vyhledané produkty.

Čtvrtá část je určena pro takzvanou „drobečkovou navigaci“, která zobrazuje aktuální zanoření uživatele v aplikaci. Tato navigace je zobrazována pouze u prohlíženého produktu nebo prohlížené kategorie, kde může být zanoření hluboké.

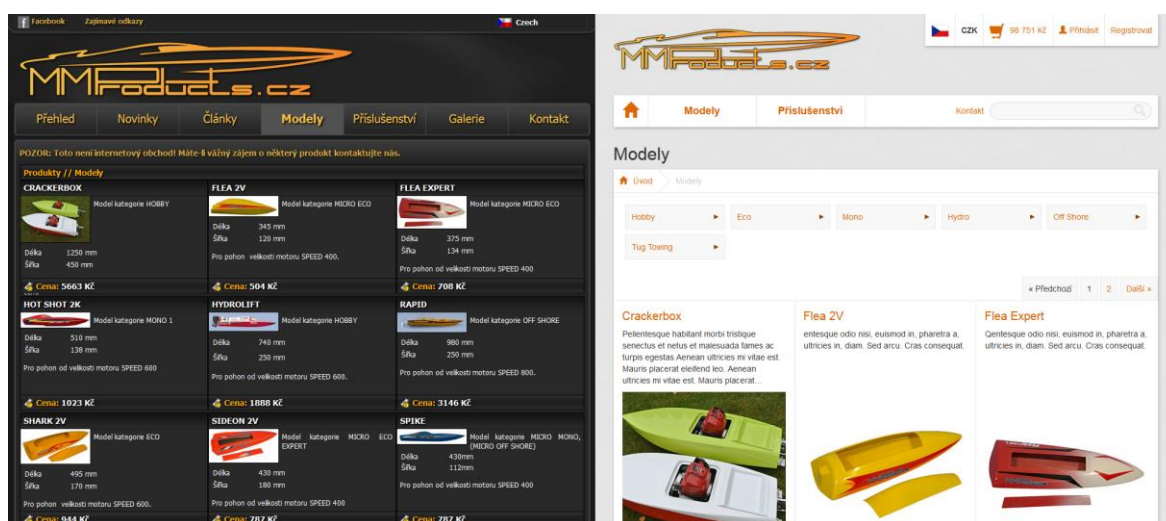
Poslední, pátá část, je určena pro zobrazení veškerého obsahu. Především jde o zobrazení produktů, kategorií produktů, informačních textů, objednávek atd.

2.4.2 Grafický návrh

Design stávající webové prezentace je velice tmavý a pro internetový obchod nevhodný. Díky tmavému pozadí zanikají ovládací prvky, fotografie produktů i obsahové texty, nebo naopak bijí moc do očí. Proto jsem se rozhodl navrhnout světlý a vzdušný design, který bude přehledný a nebude přeplněn zbytečnostími (viz. Obrázek 35).

Při vytváření návrhu jsem se inspiroval vzhledem loga stávající webové prezentace firmy. Na logu mě zaujala oranžová barva a rozhodl jsem se vytvořit grafický návrh, který bude využívat tuto barvu pro akční prvky, informační ikony a odkazy.

Pozadí stránky jsem zvolil lehce zašedlé, které na uživatele nepůsobí příliš agresivně a neodtahuje tak pozornost od obsahové části. Obsah má strhnout pozornost, a proto jsem pro jeho pozadí zvolil barvu bílou. Barva textu není úplně sytě černá, aby se příjemně četla a po delším čtení nebolely oči.



Obrázek 35 – Porovnání starého a nového vzhledu

2.5 Popis a ukázky funkcí aplikace

2.5.1 Registrace a přihlášení uživatele

Pokud uživatel chce využívat elektronický obchod plnohodnotně, musí se zaregistrovat pomocí formuláře, nebo může registraci provést pomocí sociální služby Facebook. V registračním formuláři vyplní uživatel přihlašovací a fakturační údaje. Uživatel je o chybně vyplněném políčku informován po odeslání formuláře. Chybová informace je zobrazena hned pod vstupním políčkem, kde chyba nastala.

Pro umožnění přihlášení pomocí sociální služby Facebook musí uživatel na svém Facebook účtu povolit přihlašovací aplikaci, dát jí základní oprávnění, které zpřístupní veřejný profil, seznam přátel a e-mailovou adresu. Pokud není v aplikaci tento Facebook účet veden, je automaticky vytvořen pomocí dostupných informací z veřejného profilu služby. Přihlašování sociální službou Facebook je velice jednoduché, rychlé a uživateli šetří čas (viz. Obrázek 36).

The image shows a registration form titled "Registrovat". At the top, there is a blue button with the Facebook logo and the text "Přihlásit se přes Facebook". Below this, the form is divided into two columns of input fields. The left column is titled "Přihlašovací údaje" and contains four fields: "Email", "Email znovu", "Heslo", and "Heslo znovu". The right column is titled "Fakturační údaje" and contains seven fields: "Jméno a příjmení (název firmy)", "Ulice", "Země" (with a dropdown menu showing "Česká republika"), "Město", "PSČ", "Kontaktní email", "IČ", and "DIČ". At the bottom center of the form is an orange button labeled "Zaregistrovat".

Obrázek 36 - Registrace uživatele





2.5.2 Nákupní košík a vytvoření objednávky

Nákupní košík obsahuje produkty, které si uživatel přidal za účelem nákupu. Před vytvořením objednávky je umožněno odstranit jednotlivé položky nebo měnit jejich počet. Uživatel nemusí změny nijak ukládat, vše je automatické. Při změně jsou přepočítány i celkové ceny produktů a celková částka k úhradě (viz. Obrázek 37).

Chce-li návštěvník zboží z košíku objednat, přesune se na další krok pomocí oranžového tlačítka pokračovat. Než bude objednávka vytvořena, musí uživatel vyplnit fakturační údaje a dodací údaje, podle kterých jsou mu nabídnuty dopravní služby. Při shodě fakturačních a dodacích údajů lze zaškrtnutím políčka „*Dodací adresa stejná jako fakturační*“ použít fakturační údaje jako dodací a uživatel takto ušetří čas, který by musel navíc vynaložit při vyplňování duplikátních dat.

V dalším kroku vybere uživatel platební službu, pomocí které uhradí danou objednávku. Také je nutné zvolit dopravní službu, která doručí objednané zboží. Seznam dopravních služeb je generován dle země uvedené v dodací adrese.

Před odesláním objednávky, je uživateli zobrazen celkový souhrn zahrnující cenu za dopravní a platební službu. Pokud vše souhlasí, potvrdí uživatel závazně objednávku a tím vše dokončí. Po odeslání objednávky je automaticky vyprázdněn nákupní košík.

| Produkt | Počet kusů | Cena za kus | Celková cena | |
|---|------------------------------------|-------------|-----------------|------------------|
|  Crackerbox | 4 + - | 4 680 Kč | 18 720 Kč | ✘ |
|  Flea 2V | 3 + - | 417 Kč | 1 251 Kč | ✘ |
|  Flea Expert | 4 + - | 585 Kč | 2 340 Kč | ✘ |
|  Shark 2V | 8 + - | 780 Kč | 6 240 Kč | ✘ |
| | | | Celkem bez DPH | 28 551 Kč |
| | | | DPH (21 %) | 5 996 Kč |
| | | | K úhradě | 34 547 Kč |

[Pokračovat](#)

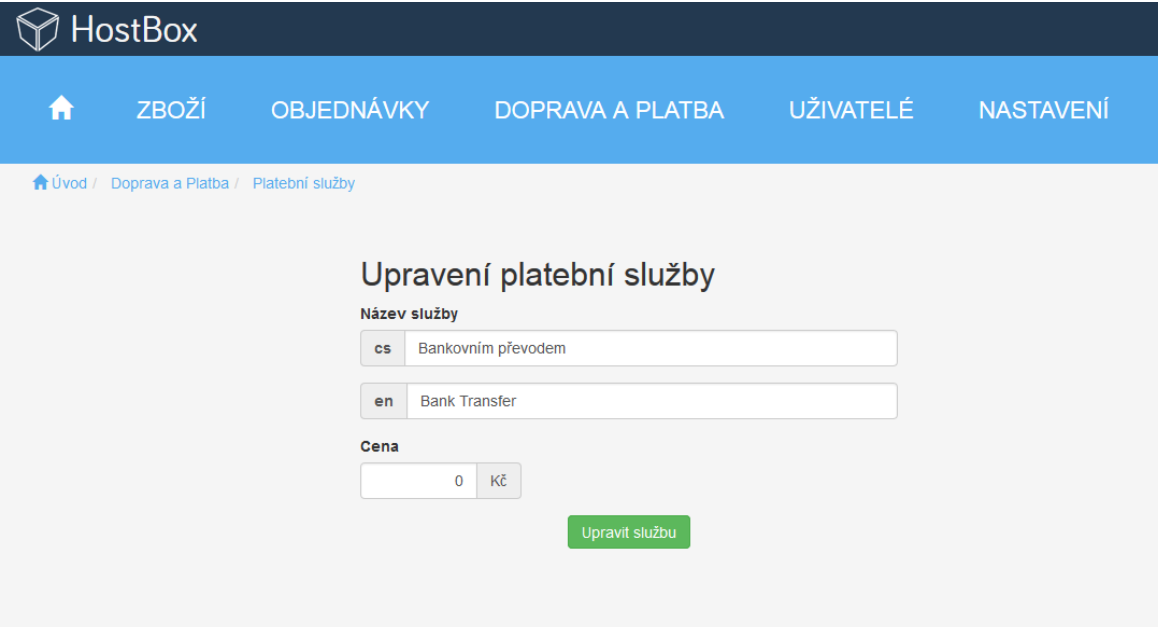
Obrázek 37 - Nákupní košík

2.5.3 Úprava platební služby

Spravování platebních služeb se nachází v administrační části aplikace, která je umístěna na subdoméně cms.mmproducts.cz. Pro přístup do této části aplikace je nutné provedení autentizace a autorizace. Přístup je povolen jen uživatelům s administrátorskými právy.

V hlavním menu administrace lze nalézt správu platebních služeb pod položkou „*doprava a platba*“. Po rozkliknutí této položky, se administrátor dostane na rozcestí sekcí, kde lze vybrat požadovanou správu platebních služeb.

Pro upravení platební služby stačí vybrat požadovanou službu ze seznamu a příslušným tlačítkem „*Upravit*“ se přesunout do editačního formuláře (viz Obrázek 38). Upravit lze název služby a její cenu. Cena se zadává v nastavené výchozí měně, která je pro tuto aplikaci nastavena na Koruny české (Kč). Název služby je nutné nastavit pro všechny dostupné jazyky, jinak není možné úpravu dokončit. Do položky cena lze zapsat pouze číselnou hodnotu. Pokud uživatel nesplní kritéria jednotlivých položek je o chybě informován po odeslání formuláře. Informace o chybě je zobrazena hned pod vstupním políčkem, kde chybu udělal. Dokončení úpravy služby provede uživatel zeleným tlačítkem „*Upravit službu*“. Po dokončení úpravy je uživatel přesměrován na seznam platebních služeb a je informován o úspěšnosti akce.



The screenshot shows the HostBox admin interface. At the top, there is a navigation bar with a home icon and menu items: ZBOŽÍ, OBJEDNÁVKY, DOPRAVA A PLATBA, UŽIVATELÉ, and NASTAVENÍ. Below the navigation bar, there is a breadcrumb trail: Úvod / Doprava a Platba / Platební služby. The main content area is titled 'Upravení platební služby'. Under the heading 'Název služby', there are two input fields: one for 'cs' with the value 'Bankovním převodem' and one for 'en' with the value 'Bank Transfer'. Below these is a 'Cena' section with an input field containing '0' and a 'Kč' unit selector. At the bottom right of the form is a green button labeled 'Upravit službu'.

Obrázek 38 - Upravování platební služby

Závěr

V rámci bakalářské práce vznikl produkt, který může být komerčně využit pro celou řadu firem a společností, které chtějí své zboží prodávat prostřednictvím internetového e-shopu. Vzhledem k různorodosti sortimentu, který může být prostřednictvím e-shopu nabízen, může nastat situace, kdy bude třeba některé části systému upravit. Jádro je však vytvořené pro univerzální použití a úprava se v tomto případě bude týkat pouze některých modulů.

Během práce na vývoji aplikace jsem zjistil, že definice univerzálního, vícejazyčně lokalizovaného e-shopu tak, jak je uvedena v zadání, je pro reálné nasazení v praxi nepřesná. Nabízení konkrétních produktů a kategorií pouze pro vybrané jazykové mutace není v praxi reálně využitelné. Důležité není nastavení jazykové mutace uživatele, které jde změnit, ale region kde se uživatel právě nachází. Zboží obchodu musí být pro všechny jazykové mutace stejné, ale nemusí být dostupné pro určité regiony, kde zboží nemusí splňovat místní zákony nebo licenční ujednání.

Aplikace splňuje všechny požadavky uvedené v zadání práce. Budoucí vývoj aplikace by mohl směřovat například k minimalizaci rizika chyby při zpracování objednávky zaměstnancem. Objednávky by mohly být generovány automaticky ve formátu PDF a zasílány do skladu firmy k vyřízení. V kopii mohou být zasílány zákazníkovi, který by tím byl informován, že se jeho objednávka právě zpracovává.

Před nasazením aplikace do ostrého provozu na adrese <http://www.mmproducts.cz> musí být provedeno několik úprav, o které bylo zažádáno až v průběhu tvorby této bakalářské práce. Především se jedná o implementaci služby PayPal, která je nejrozšířenějším webovým platebním nástrojem.

Podařilo se mi získat přehled v problematice lokalizace webových aplikací. Jeden z nejdůležitějších úkolů je správný návrh databázového modelu, který musí umožňovat plnou podporu pro překlad textových dat. Díky výběru Nette Framework pro implementaci aplikace se mi naskytla možnost procvičit si své znalosti tohoto nástroje. Vzhledem k tomu, že znalost Nette bývá jednou z podmínek pro přijetí na pracovní pozici vývojáře webových aplikací, byla tato moje zkušenost jednoznačným přínosem, který zvyšuje mé uplatnění na trhu práce v oblasti IT.

Literatura

- [1] **Jan Škrášek a Zdeněk Lehocký.** *PHP frameworky* [online] 21. 2. 2008 [cit. 20.4.2014]. Dostupné z: <http://programujte.com/clanek/2008022000-php-frameworky>
- [2] **David Čápka.** *Popis MVC architektury* [online] 2013. Dostupné z: <http://www.devbook.cz/objektovy-mvc-redakcni-system-v-php-popis-architektury>
- [3] Robustní architektonický základ. *Co je MVC.* [online] [cit. 15.4.2014]. Dostupné z: <http://webdev.felk.cvut.cz/~chlumma1/usi/raz/mvc.html>
- [4] **Jaroslav Jakoubě.** *ORM test PHP frameworků – CakePHP, CodeIgniter* [online] 10. 7. 2013 [cit. 14.4.2014]. Dostupné z <http://www.zdrojak.cz/clanky/test-php-frameworku-cakephp-codeigniter>
- [5] CakePHP: oficiální dokumentace. *Code Generation with Bake.* [online]. Dostupné z <http://book.cakephp.org/2.0/en/console-and-shells/code-generation-with-bake.html>
- [6] Fhalcon Documentation. *Hello World Benchmark* [online] 2013. Dostupné z: <http://docs.phalconphp.com/en/latest/reference/benchmark/hello-world.html>
- [7] Symfony: oficiální dokumentace. *Vysvětlení Symfony pro vývojáře* [online] 18. 3. 2013 [cit. 21.4.2014]. Dostupné z: <http://symfony.com/cs/explained-to-a-developer>
- [8] **Jiří Koutný.** *Symfony2, těší mě!* [online] 26. 6. 2013. Dostupné z: <http://www.zdrojak.cz/clanky/symfony2-tes-me>
- [9] Nette: oficiální dokumentace. *Seznámení s Nette Frameworkem* [online], 17. 1. 2014 [cit. 23.4.2014]. Dostupné z: <http://doc.nette.org/cs/2.1/getting-started>
- [10] **Adaptic,** *HTML* [online] [cit. 24.4.2014]. Dostupné z: <http://www.adaptic.cz/znalosti/slovnicek/html>
- [11] **Q-Success,** *Server-side Programming Languages* [online] 2014 [cit. 25.4.2014]. Dostupné z: <http://w3techs.com>
- [12] International Organization for Standardization . *Currency codes - ISO 4217.* [online] [cit. 26.4.2014]. Dostupné z: http://www.iso.org/iso/currency_codes

- [13] **LECKY-THOMPSON, Ed a Steven D NOWICKI.** *PHP 6: programujeme profesionálně.* Vyd. 1. Překlad Ondřej Gibl. Brno: Computer Press, 2010, 718 s. Programujeme profesionálně, no. 2. ISBN 978-80-251-3127-5.
- [14] **BORONCZYK, Ed a Steven D NOWICKI.** *PHP 6: programujeme profesionálně.* Vyd. 1. Překlad Ondřej Gibl. Brno: Computer Press, 2010, 718 s. Programujeme profesionálně, no. 2. ISBN 9788025127674.
- [15] **BAGUI, Sikha a Richard EARP.** *Database design using entity-relationship diagrams.* 2nd ed. Boca Raton, FL: CRC Press, c2012, xxvii, 343 p. Foundations of database design series. ISBN 14-398-6176-5.
- [16] **PILGRIM, Mark.** *Dive into HTML5.* [online] 2011 [cit. 2012-10-11]. Dostupné z: <http://diveintohtml5.info>, <http://kniha.html5.cz>
- [17] **Luke Welling, Laura Thomson.** *PHP and MySQL- rozvoj webových aplikací.* Překlad Martin Minář, Libor Jelínek. SoftPress s.r.o., 2002, 718s. ISBN 80-86497-20-8.
- [18] **Jakub Vrána.** *1001 tipů a triků pro PHP.* Vyd.1. Brno: Computer Press, 2010, 456s. ISBN 978-80-251-2940-1

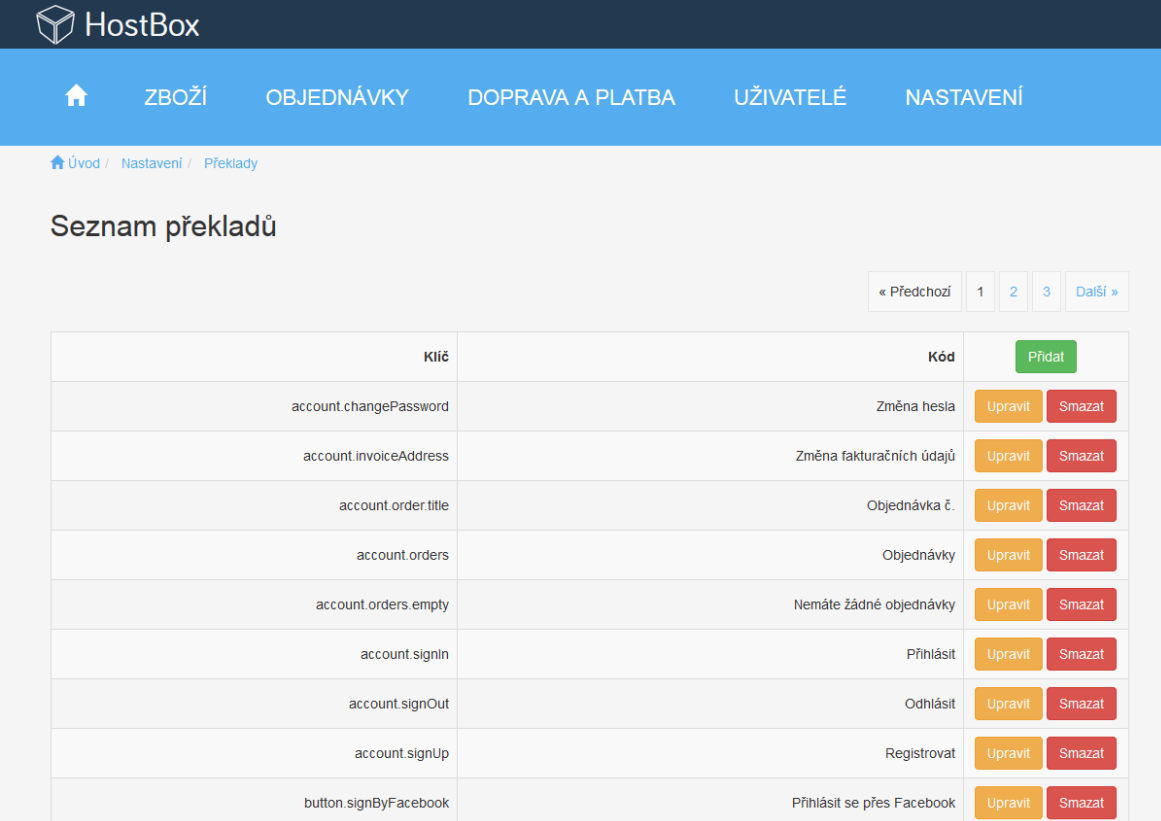
Příloha 1 – Ukázka aplikace: Veřejná část

The screenshot displays the public interface of the MMFodrats.cz website. At the top, there is a navigation bar with the company logo, a Czech flag, the currency 'CZK', a shopping cart icon labeled 'Nákupní košík', a user profile icon for 'Pavel Macháň', and a refresh icon. Below this is a secondary navigation bar with a home icon, 'Modely', 'Příslušenství', and 'Kontakt' links, along with a search bar. The main content area is titled 'Modely' and features a breadcrumb trail 'Úvod > Modely'. A category filter bar includes 'Hobby', 'Eco', 'Mono', 'Hydro', 'Off Shore', and 'Tug Towing'. A pagination control shows '« Předchozí 1 2 Další »'. Three product cards are displayed:

- Crackerbox**: Price 4 680 Kč (s DPH 5 663 Kč). Includes a 'Přidat do košíku' button. The image shows two motorboats, one green and one white.
- Flea 2V**: Price 417 Kč (s DPH 505 Kč). Includes a 'Přidat do košíku' button. The image shows a yellow motorboat hull.
- Flea Expert**: Price 585 Kč (s DPH 708 Kč). Includes a 'Přidat do košíku' button. The image shows a red motorboat hull.

Obrázek 39 - Ukázka aplikace: veřejná část

Příloha 2 – Ukázka aplikace: administrace



The screenshot displays the HostBox administration interface. At the top, there is a dark blue header with the HostBox logo and a light blue navigation bar with menu items: ZBOŽÍ, OBJEDNÁVKY, DOPRAVA A PLATBA, UŽIVATELE, and NASTAVENÍ. Below the navigation bar, a breadcrumb trail shows 'Úvod / Nastavení / Překlady'. The main content area is titled 'Seznam překladů' and features a table of translations. The table has three columns: 'Klíč', 'Kód', and 'Přidat'. The 'Přidat' column contains a green 'Přidat' button for the first row and yellow 'Upravit' and red 'Smazat' buttons for the subsequent rows. The table lists various translation keys and their corresponding codes, such as 'account.changePassword' for 'Změna hesla' and 'button.signByFacebook' for 'Přihlásit se přes Facebook'.

| Klíč | Kód | Přidat |
|------------------------|----------------------------|--|
| account.changePassword | Změna hesla | <input type="button" value="Upravit"/> <input type="button" value="Smazat"/> |
| account.invoiceAddress | Změna fakturačních údajů | <input type="button" value="Upravit"/> <input type="button" value="Smazat"/> |
| account.order.title | Objednávka č. | <input type="button" value="Upravit"/> <input type="button" value="Smazat"/> |
| account.orders | Objednávky | <input type="button" value="Upravit"/> <input type="button" value="Smazat"/> |
| account.orders.empty | Nemáte žádné objednávky | <input type="button" value="Upravit"/> <input type="button" value="Smazat"/> |
| account.signIn | Přihlásit | <input type="button" value="Upravit"/> <input type="button" value="Smazat"/> |
| account.signOut | Odhlásit | <input type="button" value="Upravit"/> <input type="button" value="Smazat"/> |
| account.signUp | Registrovat | <input type="button" value="Upravit"/> <input type="button" value="Smazat"/> |
| button.signByFacebook | Přihlásit se přes Facebook | <input type="button" value="Upravit"/> <input type="button" value="Smazat"/> |

Obrázek 40 - Ukázka aplikace: administrace

Příloha 3 – Ukázka zdrojových kódů

```
<?php

require __DIR__ . '/../vendor/autoload.php';

$configurator = new \Nette\Configurator;
$configurator->setDebugMode(array('127.0.0.1'));
$configurator->enableDebugger(__DIR__ . '/../log');
$configurator->setTempDirectory(__DIR__ . '/../temp');

$configurator->addConfig(__DIR__ . '/config/config.neon');
if (file_exists(__DIR__ . '/config/config.local.neon')) {
    $configurator->addConfig(__DIR__ . '/config/config.local.neon');
}

$configurator->createRobotLoader()
    ->addDirectory(__DIR__)
    ->addDirectory(__DIR__ . '/../libs/')
    ->register();

return $configurator->createContainer();
```

Obrázek 41 - Ukázka zdrojových kódů: bootstrap.php

```

<?php

namespace MMProducts\Library\Authenticators;

use MMProducts\Model\Entity\Account;
use MMProducts\Model\Facade\AccountFacade;
use Nette\Security\Identity;
use Nette\Utils\Strings;

abstract class BaseAuthenticator {

    /** @var AccountFacade */
    protected $accountFacade;

    public function __construct(AccountFacade $accountFacade) {
        $this->accountFacade = $accountFacade;
    }

    protected function createIdentity(Account $account) {
        $identity = $account->getRowData();
        unset($identity['password']);

        return new Identity($account->id, $account->role->name, $identity);
    }

    /**
     * Computes salted password hash.
     * @param string $login
     * @param string $password
     * @param string|null $salt
     * @return string
     */
    public static function calculateHash($login, $password, $salt = NULL) {
        if ($password === Strings::upper($password)) {
            $password = Strings::lower($password);
        }

        return crypt($login . $password, $salt ? : '$2a$07$' . Strings::random(22));
    }
}

```

Obrázek 42 - Ukázka zdrojových kódu: BaseAuthenticator.php

Příloha 4 – Obsah příloženého CD

Příložené CD obsahuje text bakalářské práce v elektronické podobě a kompletní programový systém pro e-shop s podporou mezinárodního prodeje včetně všech zdrojových kódů, diagramů a použitých obrázků.