UNIVERZITA PARDUBICE

FAKULTA ELEKTROTECHNIKY A INFORMATIKY

DIPLOMOVÁ PRÁCE

2013 Bc. Jan Kočí

UNIVERZITA PARDUBICE
Fakulta elektrotechniky a informatiky

Návrh a implementace internetového obchodu v modelu SaaS

Jan Kočí

Diplomová práce
2013

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: **2012/2013**

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Bc. Jan Kočí**
Osobní číslo:	**I09366**
Studijní program:	**N2646 Informační technologie**
Studijní obor:	**Informační technologie**
Název tématu:	**Návrh a implementace internetového obchodu v modelu SaaS**
Zadávající katedra:	**Katedra softwarových technologií**

Zásady pro vypracování:

Hlavním cílem diplomové práce je návrh a implementace internetového obchodu v modelu SaaS. V úvodní, teoretické části bude shrnut význam internetového obchodu a přehled základních požadavků na aplikaci a bude vymezen byznys model SaaS (Software as a Service). Dále budou vymezeny základní charakteristiky a požadavky na funkčnost navrhovaného a implementovaného internetového obchodu, které budou zohledňovat konkurenceschopnost systému (je třeba jej vymezit k dostupným systémům stejného typu). Součástí návrhu a implementace bude analýza požadavků prostřednictvím UML diagramů pomocí softwarového nástroje Enterprise Architect, návrh datových tříd a jejich grafické vizualizace a vytvoření konkrétní databáze (MySQL). Aplikace bude naprogramovaná v jazyce PHP 5 pomocí Nette frameworku a při implementaci budou dodržovány principy a zásady tvorby webových aplikací. Při implementaci systému je třeba důsledně dodržet základní požadavek vhodného uživatelského rozhraní a požadavky na optimalizaci a bezpečnost spravované webové aplikace.

Rozsah grafických prací:

Rozsah pracovní zprávy:

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

Seznam odborné literatury:

GÁLA, Libor, POUR, Jan, TOMAN, Prokop.Podniková informatika.Mgr. Petr Mušálaek.1. vyd. Praha: Grada Publishing, 2006. 484 s. Managment v informační společnosti. ISBN80-247-1278
Gutmans, A.: Mistrovství v PHP5. 1. vydání. Nakladatelství ComputerPress,a.s., 2005. ISBN:80-251-1519-0
Smička, R.: Optimalizace pro vyhledávače SEO. 1. vydání. Jaroslava Smicková, Dubany, 2004. ISBN:80-239-2961-5
Dokumentace Nette framework. [online]. [cit. 2012-10-26]. Dostupné z: http://doc.nette.org/cs/
Kofler, M.: Mistrovství v MySQL5 Kompletní průvodce webového vývojáře. Nakladatelství ComputerPress, a.s., 2007. ISBN:978-80-251-1502-2

Vedoucí diplomové práce: **Ing. Jana Holá, Ph.D.**
Katedra informatiky, managementu a radiologie

Datum zadání diplomové práce: **31. října 2012**
Termín odevzdání diplomové práce: **17. května 2013**

Univerzita Pardubice Česká republika

L.S.

prof. Ing. Simeon Karamazov, Dr.
děkan

prof. Ing. Antonín Kavička, Ph.D.
vedoucí katedry

V Pardubicích dne 15. listopadu 2012

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.

Nesouhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 17. 3. 2013 Bc. Jan Kočí

Poděkování

Rád bych na tomto místě poděkoval Ing. Janě Holé, Ph.D. za vedení mé diplomové práce a za podnětné návrhy, které ji obohatily.

Anotace

Tato práce se zaměřuje na tvorbu internetového obchodu v modelu SaaS. Systém je v práci analyzován, navržen, implementován a následně i nasazen do ostrého provozu. Výsledkem je modulární systém schopný konkurence na současném trhu.

Klíčová slova

internetový obchod, SaaS, Nette framework, webová aplikace

Title

Design and implementation of online store in a SaaS model

Annotation

The thesis focuses on complete development of online store in a SaaS model. The system is analyzed, designed, implemented and subsequently deployed into full operation. The result is a modular system capable of competition in the market today.

Keywords

online store, SaaS, Nette framework, web application

Obsah

Úvod **14**

1 Definice konkurenceschopného internetového obchodu **15**

1.1. Rozdělení obchodů dle segmentu trhu *15*
1.1.1 Business-to-consumer (B2C) 15
1.1.2 Business-to-businnes (B2B) 15
1.1.3 Hybridní B2C a B2B systémy 15

1.2 Základní funkcionality internetového obchodu *16*
1.2.1 Veřejná část systému 16
1.2.2 Administrační část systému 21

1.3 SEO *21*

1.4 Marketingové nástroje *22*
1.4.1 Srovnávače cen 22
1.4.2 Analytické nástroje 24
1.4.3 E-mailing 25

1.5 Přístupnost a použitelnost *25*
1.5.1 Responzivní web design 26

1.6 Platební metody *26*
1.6.1 Off-line metody 27
1.6.2 On-line metody 28
1.6.3 Shrnutí 31

1.7 Způsoby dopravy *31*
1.7.1 Dopravci 32
1.7.2 Odběrná místa 32
1.7.3 Shrnutí 33

1.8 Rozšiřitelnost aplikace *33*
1.8.1 Propojení s ekonomickými a informačními systémy 33
1.8.2 Napojení na dodavatele 34

1.9 Servis a podpora *34*

2 Možnosti pořízení internetového obchodu **35**

2.1 Open source systémy *35*

2.2 Pronájem formou SaaS *35*
2.2.1 Software as a Service 36

2.3 Komerční řešení *37*

2.4 Řešení na klíč *37*

3 Srovnání konkurenčních systémů 38
3.1.1 Kritéria výběru porovnávaných systémů 38
3.1.2 Vybrané konkurenční systémy 38
3.1.3 Porovnání se systémem Weseus III 39

4 Vývoj a implementace aplikace 42
4.1 Analýza současného stavu systému 42
4.2 Analýza požadavků 43
4.3 Technologie pro návrh a implementaci systému 44
4.3.1 Platforma LAMP 44
4.3.2 Nette framework 46
4.3.3 Analytické a vývojové nástroje 47
4.3.4 Verzovací systém Git 47
4.3.5 Správce závislostí Composer 48
4.4 Návrh případů užití 50
4.4.1 Aktéři 51
4.5 Návrh datových tříd a architektury systému 51
4.5.1 Architektura MVP 52
4.5.2 Návrhový vzor Dependency Injection 53
4.5.3 Komponentový přístup 54
4.6 Návrh databáze 54
4.7 Implementace systému 56
4.7.1 Adresářová struktura 56
4.7.2 Front-end 57
4.7.3 Back-end 60
4.8 Testování 62
4.8.1 Uživatelské testování 62
4.8.2 Testování použitelnosti 62
4.9 Nasazení 63

Závěr 66

Seznam zkratek

SaaS	Software as a Service
PPC	Pay per click
B2C	Business to consumer
B2B	Business to business
ERP	Enterprise resource planning
CRM	Customer relationship management
AJAX	Asynchronous JavaScript and XML
SEO	Search Engine Optimization
URL	Uniform Resource Locator
XML	Extensible Markup Language
HTML	HyperText Markup Language
CPC	Cost Per Click
APEK	Asociace Pro Elektonickou Komerci
CSS3	Cascading Style Sheets 3
EAN	European Article Number
UX	User Experience
POS	Point of sale
API	Application Programming Interface
SMS	Short message service
PIN	Personal identification number
ERP	Enterprise Resource Planning
EIS	Enterprise information systém
SLA	Service Level Agreement
LAMP	Linux Apache MySQL PHP
PHP	PHP: Hypertext Preprocessor

OOP	Objektově orientované programování
MySQL	My Structured Query Language
SQL	Structured Query Language
IDE	Integrated Development Environment
UML	Unified Modeling Language
ORM	Object-relational mapping
CSRF	Cross-site request forgery
XSS	Cross-site scripting
CASE	Computer-aided software engineering
MVC	Model-View-Controller
MVP	Model-View-Presenter
DI	Dependency Injection
HTTP	Hypertext Transfer Protocol

Seznam obrázků

Obrázek 1- Dynamické filtrování dle parametrů. Zdroj alza.cz. 17

Obrázek 2- Vyhledávání s našeptávačem. Zdroj www.euronics.cz. 18

Obrázek 3- Příklad Cross-sellingu v obchodě Kasa.cz. 19

Obrázek 4 - Heat mapa před redesignem e-shopu [5] 20

Obrázek 5 - Heat mapa e-shopu po redesignu [5] 20

Obrázek 6 - Graf znázorňuje rozdíly v návštěvnostech u zbožových srovnávačů a katalogů sledovaných nástrojem Effectix Analyzer. [11] 24

Obrázek 7- Graf konverzního poměru Google Analytics. 25

Obrázek 8 - Přístupová zařízení v roce 2012 [15] 26

Obrázek 9 - Nejčastější způsoby platby Čechů v roce 2012 [20] 27

Obrázek 10 -Srovnání oblíbenosti platebních metod v letech 2010 až 2012. [20] 29

Obrázek 11- Nejvyužívanější dopravci v ČR v roce 2012 [21] 31

Obrázek 12- Diagram základních požadavků systému 44

Obrázek 13- Schéma architektury platformy LAMP. Zdroj [27]. 45

Obrázek 14- Grafické znázornění příkazů verzovacího systému Git. Zdroj [28]. 48

Obrázek 15 - Use Case diagram internetového obchodu. 50

Obrázek 16- Schéma architektury Model-View-Presenter. Zdroj[29]. 52

Obrázek 17- Výřez ER diagramu zobrazující produkt a dynamické parametry. 55

Obrázek 18- Základní adresářová struktura aplikace. 56

Obrázek 19- Životní cyklus presenteru Nette frameworku. Zdroj [30]. 58

Obrázek 20 -Příklad komponenty SearchInput. 60

Obrázek 21- Uživatelské rozhraní administrace, konkrétně seznam objednávek. 61

Obrázek 22 - Ukázka filtrování produktů na výpisu kategorie. 64

Obrázek 23 - Ukázka nasazení systému v praxi. Zachycuje implementaci unikátní grafické šablony. 65

Seznam tabulek

Tabulka 1- Jakým způsobem Češi vybírají e-shopy? [10] 23

Tabulka 2- Porovnání cen a základních vlastností tarifů konkurenčních systémů. 39

Tabulka 3- Porovnání funkcionalit konkurenčních systémů. 40

Tabulka 4 -Porovnání podpory platebních metod a dopraních služeb. 41

Úvod

Diplomová práce je zaměřená na návrh a implementaci internetového obchodu určeného pro komerční využití. Výsledný produkt by měl být v praxi distribuován jako služba a měl by komplexně pokrýt všechny požadavky na systém obsluhující proces on-line obchodování.

V teoretické části práce bude produkt nejprve vymezen pro určitý segment trhu a dále budou zmapovány požadavky na moderní internetový obchod. Pro konkurenceschopnost vyvíjeného systému je nutné popsat současné trendy spojené s internetovým obchodováním a on-line marketingem. Další důležitou částí je vymezení byznys modelu SaaS (Software as a Service) a porovnání konkurenčních systémů, které pomůže vyspecifikovat požadavky na konečnou podobu systému.

Výstupem této práce bude funkční systém, který bude v praktické části nejprve analyzován a modelován pomocí CASE (Computer-Aided Software Engineering) nástrojů a jazyka UML (Unified Modeling Language). Pro urychlení implementace webové aplikace bude použit Nette framework, který klade důraz na znovupoužitelnost kódu a eliminuje bezpečnostní rizika. Před nasazením v praxi bude systém řádně otestován a poté bude popsán proces jeho nasazení jako služby.

Internetový obchod, jehož implementace je v práci popsána se nazývá Weseus III. Název je odvozen od redakčního systému společnosti Webový servis, pro kterou je systém vytvářen. Hlavními požadavky je zachování původních funkcí redakčního systému, rozšíření o modul internetového obchodu a nasazení v byznys modelu SaaS.

1 Definice konkurenceschopného internetového obchodu

1.1. Rozdělení obchodů dle segmentu trhu

Internetové obchody můžeme rozdělit na dva základní typy podle toho, pro jaký segment trhu jsou primárně určené.

1.1.1 Business-to-consumer (B2C)

E-shopy typu B2B jsou určené pro prodej koncovým spotřebitelům (fyzické osoby, domácnosti, maloodběratelé apod.). Cílem takového obchodu je v první řadě získání a udržení zákazníka. Většinou se zaměřují se na oslovení určité cílové skupiny (ženy, muži sportovci, maminky apod.) a grafickým zpracováním reprezentují druh nabízeného zboží. Z důvodu vysokého konkurenčního prostředí musí být nákupní proces snadný, rychlý a optimalizovaný pro maximální míru konverze návštěvníka na zákazníka. Úspěšný B2C obchod se v dnešní době neobejde bez optimalizace pro vyhledávače, porovnávačů zboží, sociálních sítí nebo třeba PPC reklamy.

1.1.2 Business-to-businnes (B2B)

B2B je zkratka označující obchody zaměřené na prodej svým obchodním partnerům, většinou dealerům nebo velkoodběratelům. Zákazníci se musí téměř vždy zaregistrovat a po přihlášení mohou využívat speciálních podmínek, jako například velkoobchodní ceny dle obratu, individuální slevy, dodací lhůty, dodací a fakturační údaje apod. Grafické zpracování bývá oproti B2C obchodům střízlivější a důraz bývá kladen na jednoduchost bez funkcí pro podporu prodeje. Tyto systémy se téměř neobejdou bez napojení na ERP, CRM, účetní nebo skladové systémy dodavatele. [1]

1.1.3 Hybridní B2C a B2B systémy

Na trhu lze najít systémy, které jsou určeny pro oba segmenty trhu. Dokážou implementovat výše uvedené vlastnosti z obou obchodních modelů a personifikují obsah jednotlivým skupinám uživatelů dle nastavení jejich účtu. Vzhled i struktura grafických prvků může být měněna dle druhu přihlášeného klienta. A zvláště cenotvorba bývá u těchto systémů dosti flexibilní a přistupuje ke každému individuálně.

Cílem této práce je vytvořit konkurence schopný internetový obchod. Pro naplnění cíle se nejlépe hodí hybridní systémy, které mají největší obchodní potenciál a funkčně mohou vyhovovat většímu množství obchodníků. V dalších kapitolách budou porovnávány pouze B2C a hybridní systémy. [2]

1.2 Základní funkcionality internetového obchodu

Základem internetového obchodu může být jednoduchá webová aplikace nebo naopak velmi rozsáhlý systém s velkým množstvím funkcí. Přestože se jednotlivá řešení mohou výrazně lišit, jsou vždy tvořena základními funkcemi, které lze definovat jako:

- katalog produktů,
- proces vytvoření objednávky,
- správa objednávek.

Dále je třeba rozlišit tyto základní funkcionality z pohledu zákazníka a obchodníka. Proto jsou internetové obchody rozděleny na dvě samostatné logické části:

1. veřejnou část,
2. administrační část.

1.2.1 Veřejná část systému

Veřejná část slouží primárně pro prezentaci katalogu produktů, nejčastěji formou webových stránek. Jejím hlavním cílem je dovést zákazníka k vytvoření objednávky, uzavření kupní smlouvy.

Produkt

Základní entitou katalogu produktů je produkt, který reprezentuje prodávané zboží nebo službu. Má základní parametry jako je cena, název, textový popis nebo třeba fotografie a videa sloužící k vizualizaci produktu zákazníkovi. Důležité jsou unikátní identifikátory produktu (EAN kód, kód výrobce, volitelný interní kód obchodníka apod.), které slouží například pro vyhledávání, párování se skladem nebo se systémy třetích stran. Každý produkt má v zásadě vlastní produktovou stránku s unikátní URL adresou.

Kategorie

Standardem dnešních systémů je řazení produktů do stromu kategorií, kde je každá kategorie reprezentována vlastní URL adresou. Zákazník na této stránce vidí pouze produkty z dané kategorie a běžně může využívat stránkování nebo např. řazení dle ceny. Propracovanější systémy umožňují přiřadit produktu více než jednu kategorii.

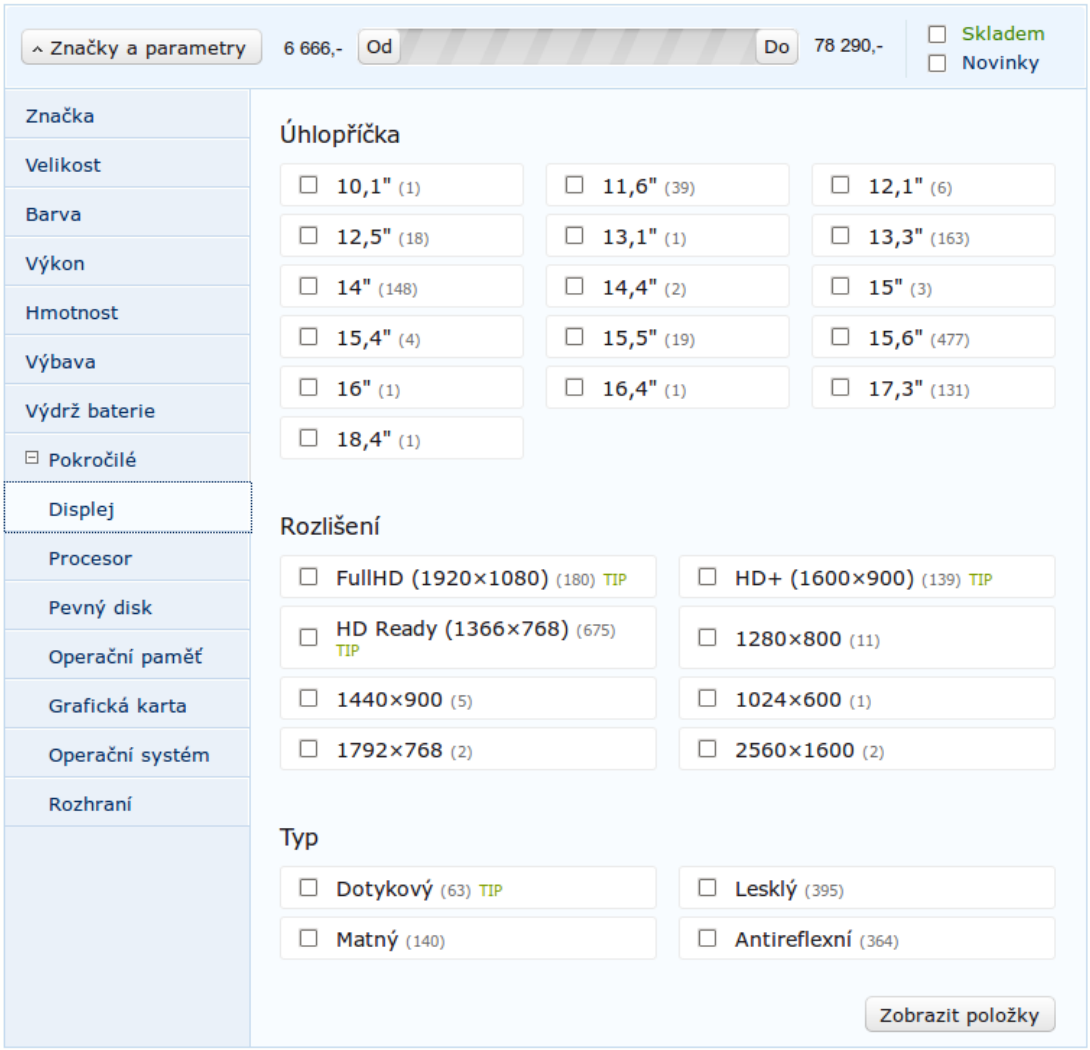
Výrobce

U obchodů, které prodávají zboží od více výrobců, je požadavek na vyhledávání dle značky (zejména u oblečení, sportovní potřeby apod.). Funkcionalita je v podstatě shodná s výpisem na stránce kategorie. V systémech, které nemají implementován modul výrobce, často vznikne špatná informační struktura vytvořením kategorie s názvem značky.

Parametrické filtrování

Uživatelsky přívětivý a moderní systém by kromě základních parametrů produktu měl umožňovat dynamickou správu libovolných parametrů. Na základě těchto dat lze poté produkty vyhledávat a konkretizovat výběr například na stránce jednotlivé kategorie. Na implementaci filtrování dle parametrů se velice hodí využití technologie AJAX, která nám umožní překreslit výběr bez znovunačtení stránky.

Obrázek 1- Dynamické filtrování dle parametrů. Zdroj alza.cz.



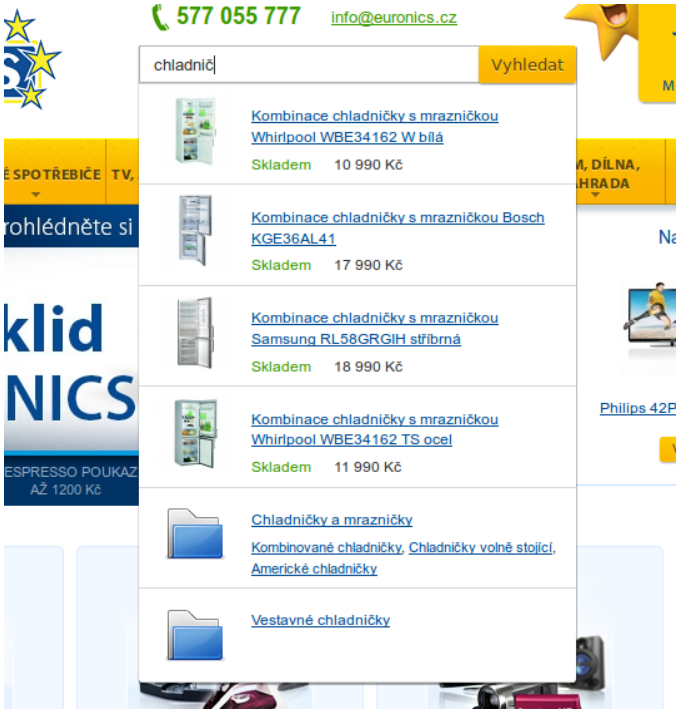
Porovnání produktů

U sortimentu technického charakteru, ale nejen tam, je velice přínosná funkce porovnávání produktů. Zákazník má možnost při procházení katalogem označit produkty, které chce porovnat. Po přechodu na stránku s porovnáním jsou mu vybrané produkty prezentovány v přehledné tabulce včetně všech jejich parametrů.

Vyhledávání

Nepostradatelnou funkcí, která dokáže snížit míru opuštění obchodu, je vyhledávání. Pomocí textového pole, které je nejvhodnější integrovat přímo do záhlaví layoutu stránky, umožní zákazníkovi vyhledat produkty obsahující zadanou frázi.

Obrázek 2- Vyhledávání s našeptávačem. Zdroj www.euronics.cz.



Uživatelsky je hodně přívětivá verze s tzv. našeptávačem, kdy se po každém stisku klávesy omezí výběr a zobrazí nabídka nalezených produktů, kategorií a výrobců. Mohou se zobrazit i libovolné informace jako fotografie produktů a loga výrobců, ale i ceny nebo skladové zásoby.

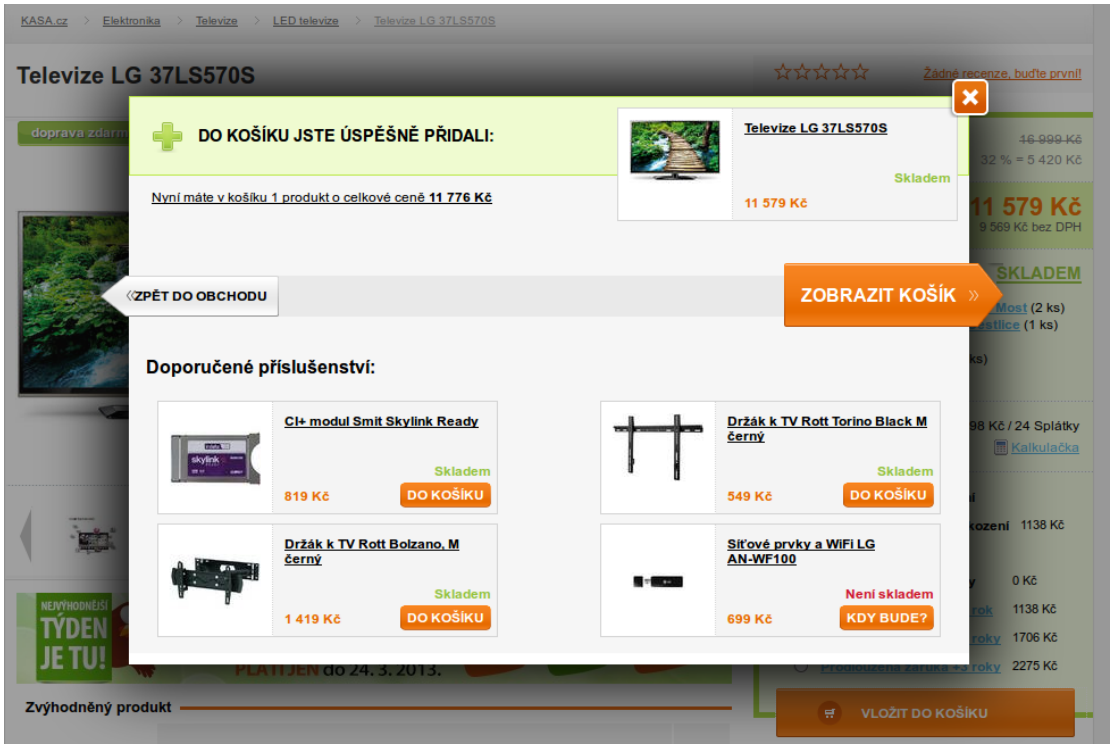
Produktové vazby

Pomocí vazeb mezi produkty lze do systému poměrně snadno implementovat Cross-selling (křížový prodej), který slouží pro nabídku ve formě příslušenství k produktu, podobného zboží nebo nabídku doplňkových služeb. Je to ideální cesta pro zvýšení prodeje příslušenství. Provázané produkty se poté zobrazují např. v košíku nebo na detailu produktu.

Dále se vazby mezi produkty dají použít na vytváření variant produktu. Varianty se běžně používají pro odlišení barvy např. u oděvů, kde mají produkty shodné všechny parametry, ale potřebují mít pouze jiné obrázky.

Sety jsou příkladem použití produktových vazeb pro sestavení výhodnější nabídky, kdy při zakoupení více produktů najednou v tzv. setu získá zákazník výhodnější cenu než při koupi totožných produktů samostatně.

Obrázek 3- Příklad Cross-sellingu v obchodě Kasa.cz.



Proces objednávky

Nejdůležitější částí internetového obchodu, na které je přímo závislá jeho úspěšnost, je proces vytvoření objednávky, takzvaný průchod nákupním košíkem. Zpravidla mívá 4 základní kroky, při kterých musí zákazník vyplnit všechny potřebné informace:

1. Potvrzení nákupního seznamu a počtu kusů.
2. Výběr způsobu dopravy a platební metody.
3. Formulář pro vyplnění osobních a kontaktních údajů o zákazníkovi, volba doručovací adresy, případně firemní údaje a potvrzení souhlasu s obchodními podmínkami.
4. Shrnutí a potvrzení objednávky.

Špatné uspořádání ovládacích prvků na stránkách košíku může mít fatální následky ve formě nedokončených objednávek a velké míry opuštění obchodu. Zákazník zde nesmí hledat, jakým způsobem objednat. Musí na první pohled být jasné, kdy objednává, kdy je objednávka závazná, proč požadované informace vyplňuje a co ještě zbývá k dokončení

objednávky vyplnit. V praxi se pro odhalení nedostatků nákupního procesu a jejich optimalizaci využívá uživatelské testování.

„Uživatel se i ve virtuálním světě rozhoduje (k nákupu) pod vlivem emocí, úplně stejně, jako ve světě reálném. A o tom, jaký prožitek si z návštěvy odnese, rozhoduje celá řada faktorů – z těch nejdůležitějších jmenujme srozumitelnost, přehlednost, ovládání, konzistentnost, užitnou hodnotu, design a obsah.“ [3]

Uživatelské testování

Používá se pro odhalení toho, jak návštěvníci vnímají testovaný web a jestli plní jejich očekávání, zda se snadno orientují nebo zda jsou pro ně srozumitelné texty. V případě košíku dokáže odhalit překážky, na které zákazník narazí a které ho případně odradí od nákupu. [6]

Obrázek 4 - Heat mapa před redesignem e-shopu [5]

Obrázek 5 - Heat mapa e-shopu po redesignu [5]

Pomocí metody Eye Trackingu (sledování pohybu očí) dostaneme pomocí speciálního softwaru výsledek ve formě Heat mapy, na které je jasně vidět, kam směřuje pozornost sledovaného uživatele. Správně navržený UX (User experience) design by měl uživatele vždy v co největší míře směřovat na tzv. tlačítka Call-to-action, pomocí kterého vyvolá námi požadovanou akci. Ostatní prvky zbytečně odvádějící pozornost je vhodné skrýt nebo barevně upravit, aby nebyly tolik výrazné. [5]

1.2.2 Administrační část systému

Administrační část obsluhuje kompletní správu obsahu, především objednávek a katalogu produktů. Obsahuje funkce pro komunikaci se zákazníky, souhrnné statistky apod. Může být reprezentována klasickým počítačovým programem, ale zpravidla je přístupná také přes internetový prohlížeč.

Správa objednávek

Přehledná správa objednávek, která umožní jejich snadné a rychlé vyřizování, musí být základem každého systému, jež chce být úspěšný. Objednávky by měly být barevně odlišeny dle stavů, ve kterých se nacházejí. Každá změna stavu by měla být zákazníkovi oznámena e-mailem. Velké množství času při správě objednávek ušetří implementace hromadných operací.

Správa katalogu produktů

Administrační část systému musí umožňovat editaci všech atributů nabízených produktů, jejich řazení do stromu kategorií a pokročilou cenotvorbu.

Přehled o nakupujících

Administrátor obchodu by měl mít dokonalý přehled o nakupujících a registrovaných zákaznících. Měl by mít možnost s jejich daty pracovat a segmentovat je pro další marketingové aktivity.

Fakturační systém

Aplikace by měla fungovat samostatně bez nutnosti napojení na účetní systém. Měla by umožňovat generovat, tisknout nebo třeba stornovat faktury dle platné legislativy.

1.3 SEO

SEO je zkratka z anglického Search Engine Optimization, což v češtině znamená optimalizace pro vyhledávače. Jedná se o úpravy stránek tak, aby byly vhodné pro algoritmy vyhledávačů a aby bylo dosaženo získání lepších pozic ve fulltextovém vyhledávání na důležitá klíčová slova. [7]

Požadavky SEO specialistů na systémy internetových obchodů jsou následující a u kvalitních systémů standardně bývají obsaženy již v základních verzích:

- Generování validního HTML kódu dle všech standardů s ohledem na principy indexace vyhledávačů.
- Možnost tvorby vlastních tzv. „čistých URL“ adres u produktů, kategorií a výrobců.
- Definování vlastních titulků, případně jejich skládání dle hierarchické struktury.

- Plnou kontrolu nad metatagy (*description*, *keywords*), veškerým obsahem pro možnost nastavení nadpisů *h1-h6*, zvýraznění klíčových slov v textu apod..
- Optimalizace obrázků pro výsledky vyhledávání. Atributy *title* a *alt* u elementu *img*.
- Poskytnutí generovaného seznamu URL adres ve formátu XML.
- Optimalizovat rychlost načítání stránek, která je pro vyhledávače jedním z rozhodujících parametrů.
- Použití kanonických URL sloužících ke zjednodušení struktury adres. Díky nim fulltextoví roboti přesněji vědí, která stránka je z pohledu webmastera důležitější.
- Využití Facebook Open Graph Meta Tags, pomocí kterých lze ovlivnit prezentaci stránek při odkazování a sdílení v sociální síti Facebook.[8]

1.4 Marketingové nástroje

Cílem každého internetového obchodu je získat co největší množství návštěvníků, kteří najdou to, co hledají a úspěšně dokončí objednávku. Pro získání kvalitních návštěvníků a potřebné analýzy jejich chování je zapotřebí, aby systém podporoval využití následujících marketingových nástrojů, pomocí kterých lze dosáhnout zvýšení obratu.

1.4.1 Srovnávače cen

Účelem srovnávačů zboží je umožnit uživatelům porovnat nabídku stejného zboží v různých e-shopech. Návštěvník zadá název hledaného zboží a srovnávač mu v registrovaných obchodech zboží vyhledá. Zákazníkovi je předložena nabídka z různých obchodů a ten si může jednoduše porovnat parametry jako je cena, skladová dostupnost, zkušenosti ostatních nakupujících s daným e-shopem apod., v závislosti na konkrétním srovnávači. [9]

Registrace obchodů do těchto portálů bývají zdarma, ale obecně obchodník platí za určitá zvýhodnění formou PPC reklamy. Zaplaceným proklikem získá na stránky návštěvníka téměř rozhodnutého k nákupu, a proto se mu tato investice vyplatí.

Z průzkumu agentury NMS Market Research mezi 6500 respondenty vyplývá, že počet zákazníků vybírajících e-shop na cenových srovnávačích roste. Zatímco v roce 2009 využívalo jejich služby pouze 19% lidí, o dva roky později jejich počet vzrostl na 44%. Naopak klesá tendence vyhledávat obchody přes klasické fulltextové vyhledávače jako Google nebo Seznam.

"Situace na internetových vyhledávačích se obrátila. Zatímco v roce 2009 vybírala obchody přes vyhledávače téměř polovina uživatelů, o dva roky později už to byla necelá třetina," dodal Michal Osuský z NMS Market Research. [10]

Tabulka 1- Jakým způsobem Češi vybírají e-shopy? [10]

	Rok 2011	Rok 2012
K výběru používám nákupní rádce nebo srovnávače cen	44 %	52 %
Obchody vybírám převážně přes internetové vyhledávače (Google, Seznam.)	30 %	20 %
Mám své oblíbené obchody a v těch nakupuji	18 %	14 %
Informace hledám na internetových fórech	3 %	5 %
Spoléhám se na doporučení přátel nebo na své vlastní zkušenosti	6 %	9 %

"Téměř trojnásobný nárůst obliby srovnávačů cen je dán citlivostí zákazníků jak na cenu zboží, tak i na kvalitu obchodu. Zákazníci na cenových srovnávačích mohou velice rychle a jednoduše porovnat nabídky tisíce obchodů a ušetřit značnou část ceny zboží," uvedl Kamil Demuth z portálu Heureka.cz. [10]

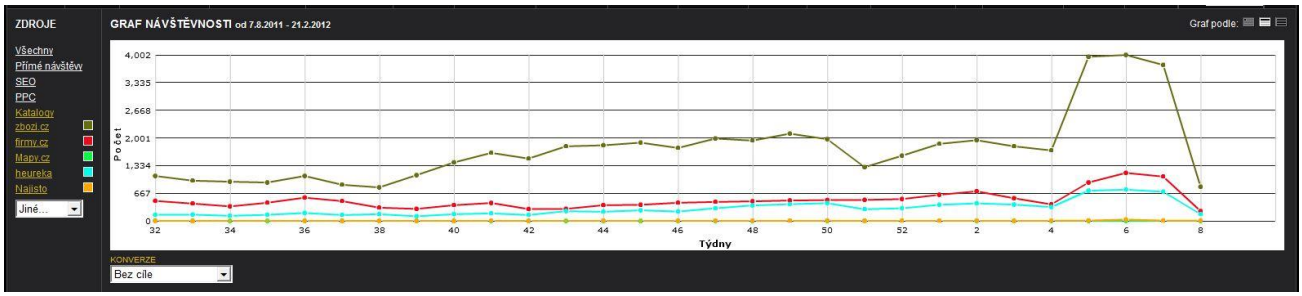
Z výše uvedených dat vyplývá, že napojení systému na srovnávače cen je pro jeho konkurenceschopnost nutností. Pro registraci obchodu do srovnávače je nutno uvést URL adresu na XML soubor (feed) obsahující informace o zboží, které propagovaný e-shop nabízí. Systém tedy musí implementovat export dat pro každý srovnávač individuálně, protože se liší specifikací a nabízenými službami. Obchodník by měl mít možnost exportovaná data v administraci ovlivnit. Zvláště pak hodnotu MAX CPC, která udává maximální hodnotu, kolik je ochoten obchodník zaplatit za proklik u konkrétního zboží.

V současnosti jsou nejvýznamnější tyto srovnávače:

- Google Nákupy,
- Heureka.cz – poskytuje cca 2-3krát větší přínos než Google Nákupy,
- Zboží.cz – poskytuje cca 5-7krát větší přínos než Heureka.cz.

“Přínosem je myšlen reálně uskutečněný obrat při využití placených možností zvýhodnění před konkurenčními nabídkami. Práce s ostatními srovnávači přináší běžně minimální obraty, proto vynaložená práce je vyšší než potenciální přínos.“ [11]

Obrázek 6 - Graf znázorňuje rozdíly v návštěvnostech u zbožových srovnávačů a katalogů sledovaných nástrojem Effectix Analyzer. [11]



1.4.2 Analytické nástroje

Pro efektivní marketing internetových obchodů je velice důležité použití analytických nástrojů, které pomocí vložených trackovacích kódů ukládají data o návštěvách, chování zákazníků a provedených objednávkách. Následně umožňují zjistit například:

- co a za kolik nakoupili jednotliví zákazníci,
- příjem v závislosti na zdroji, klíčovém slovu nebo produktu,
- průměrnou hodnotu objednávky,
- čas mezi první návštěvou webu a nákupem,
- efektivitu PPC kampaně dle poskytovatele. [12]

Dnes je bezesporu nejrozšířenějším analytickým nástrojem Google Analytics, který je zdarma a po aktivaci modulu E-commerce lze měřit všechny výše popsané metriky. Dále je vhodné implementovat napojení na systémy měření konverzí srovnávačů cen, které poskytují oba největší hráči Heureka.cz i Zboží.cz. Protože je analytických nástrojů celá řada, měl by mít marketingový specialista možnost implementovat libovolný nástroj a upravovat měřící kódy nejlépe přímo z administrace obchodu.

Konverzní poměr

Nejdůležitější metrikou webové analytiky spojenou s internetovými obchody je konverzní poměr. Jedná se o vyjádření, kolik procent návštěvníků ukončí svou návštěvu objednávkou. [13]

Obrázek 7- Graf konverzního poměru Google Analytics.

1.4.3 E-mailing

Efektivní E-mail marketing je jednou z nejsilnějších marketingových metod. Je důležitou součástí obchodování na internetu, pomáhá budovat povědomí o značce a přivést zákazníka k opětovnému nákupu. Využití newsletteru je vhodné pro metody Up-sellingu i Cross-sellingu.

Systém by měl umožňovat zákazníky segmentovat dle cílových skupin a následně umožňovat naplánování hromadného rozesílání včetně zakomponování slevových kupónů. E-maily by měly být graficky líbivé, proto je nutné umožnit přímou editaci HTML obsahu pomocí WYSIWYG editoru.

1.5 Přístupnost a použitelnost

Optimalizace webových stránek pro přístup ze všech možných počítačových platforem a internetových prohlížečů se stal běžným standardem. Trendem současnosti je přístup z mobilních zařízení, jako jsou tzv. chytré telefony (smartphony) a tablety.

Není se čemu divit, protože prodeje těchto zařízení se neustále zvyšují. Pro ilustraci studie serveru Mashable.com uvádějící, že si 73% amerických zákazníků vlastnících chytrý telefon plánuje do šesti měsíců koupit nový. Z dnešních cca 56% populace má tržní podíl smartphonů vzrůst na 74%. Počítá se, že by podobná čísla měla platit i pro Evropu. Z pohledu provozovatelů e-shopů tak růst tržního podílu smartphonů a zejména intenzita jejich používání znamená do budoucna nutnost vytvoření mobilní verze. [14]

Společnost Effetix.com, která spravuje kolem 1600 účtů na Google Analytics, očekává do roku 2015 nárůst mobilních přístupů na weby ze současných 4 % až na 15% podíl. [15]

"Dle našich indicií se zákazníci v kamenných obchodech nově chovají tak, že si cenu zboží ověřují v prodejně právě přes své chytré telefony. Pomocí mobilních verzí jim tuto práci zjednodušíme." Uvedl Jan Vetyška, výkonný ředitel APEK. [16]

Obrázek 8 - Přístupová zařízení v roce 2012 [15]

Je tedy zřejmé, že pro systémy provozující internetové obchody bude do budoucna nutností implementovat i mobilní verze, jež jsou založeny na responzivním web designu. Mobilní verze by měly mít optimalizovány datové toky, protože se klienti zpravidla připojují z pomalejších mobilních sítí. Nabídka mobilní verze může dnes být i konkurenční výhodou.

1.5.1 Responzivní web design

Jedná se o způsob stylování HTML dokumentu, který zaručí optimalizaci zobrazení stránky na různých zařízeních (mobily, notebooky, netbooky, tablety atd.). Je zde využito především vlastnosti Media Queries[1], zahrnuté ve specifikaci CSS3, která umožňuje rozpoznat zařízení, na kterém je stránka prohlížena, a následně mu přizpůsobit obsah. [17]

U e-shopů je nutné zobrazení pouze nejdůležitějších informací a jednoduchost uživatelského rozhraní. To by mělo být upraveno dle specifických požadavků na styl ovládání na dotykovém displeji telefonu. Kvůli tomu by měl být přizpůsoben vzhled a rozmístění navigačních prvků pro co nejsnadnější ovládání jedním prstem.

1.6 Platební metody

Nedílnou součástí nákupního procesu je výběr požadované platební metody zákazníkem a následné zaplacení objednávky. Český trh v současnosti nabízí následující způsoby, jak při on-line nákupu za zboží zaplatit a ty jsou obchody nabízeny prostřednictvím více než

[1] Adaptabilní vykreslování webových stránek podle různých činitelů jako rozlišení obrazovky nebo velikost obrazovky.

třiceti konkrétních služeb. Platební metody dělíme dle prostředí, ve kterém probíhají, tedy na off-line a on-line.

1.6.1 Off-line metody

Spolehlivý a zaběhlý způsob platby vhodný především pro prodej hmotného zboží. V Česku jsou stále využívanější než metody on-line plateb a jejich výhodou je méně náročná implementace do systému internetového obchodu. Je zde ale výrazná prodleva o poskytnutí informací o průběhu platby.

Dobírka

Platba při osobním odběru je dle studie APEK z roku 2012 stále zdaleka nejrozšířenější a nejjednodušší platební metodu, kterou Češi využívají při nákupu zboží přes internet. Pro obchodníky je ale tato možnost poměrně nevýhodná, protože peníze dostanou s několika denním zpožděním až po převzetí zboží zákazníkem. [18]

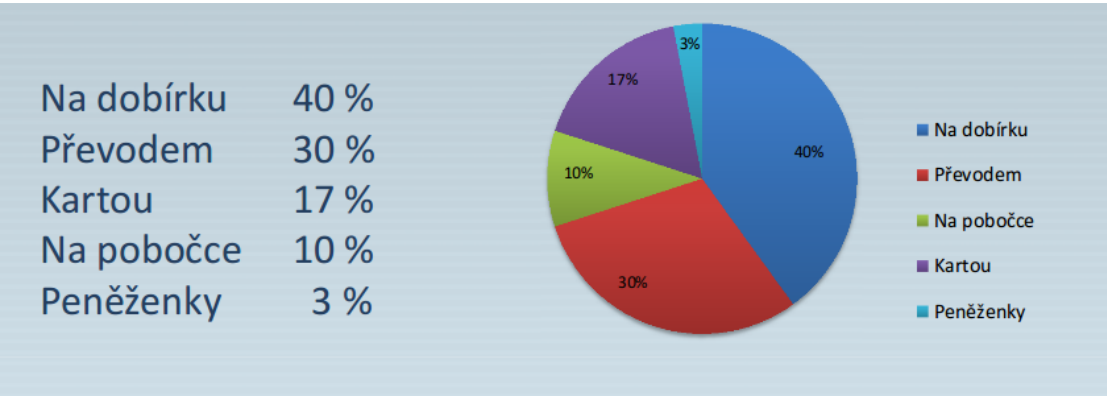
Jako platbu dobírkou označujeme platbu v hotovosti při převzetí balíku na poště nebo od specializovaného dopravce. Do této kategorie spadá i bezhotovostní platba dopravci prostřednictvím POS terminálu nebo také platba kartou na pobočce při osobním odběru.

Platba v hotovosti

Jedná se o platbu zboží v hotovosti při osobním odběru na prodejně nebo specializovaném odběrném místě.

„Také stále populární mezi zákazníky. Pro obchodníky na jednu stranu znamená zaplacení zboží ihned při prodeji, na druhou stranu ale také dodatečné náklady na provoz výdejny a správu hotovosti," uvádí APEK. [19]

Obrázek 9 - Nejčastější způsoby platby Čechů v roce 2012 [20]



Převodem na účet

Klasický transakční proces, který vyžaduje pouze založení bankovního účtu. Není zde nijak omezen rozsah plateb a transakce jsou zpoplatněny standardními sazebníky.

Nevýhodou je, že spadá do kategorie off-line plateb a připsání na účet obchodníka může trvat i dva dny.

Dle výše uvedené studie se jedná o druhou nejvyužívanější platební metodu, a proto je vhodné systém provozující internetový obchod napojit přímo na bankovní účet obchodníka. E-shop si poté automaticky páruje platby s objednávkami např. dle variabilního symbolu a obchodník se stará pouze o expedici. V současnosti nabízí API pro export dat z internetového bankovnictví např. Fio banka nebo Raiffeisenbank.

Nákup na splátky

Metoda, u které obchodník neplatí žádné poplatky, pouze musí vyčkat na schválení úvěru zákazníkovi úvěrovou společností. Poté je mu vyplacena provize za sjednaný úvěr. Splátkový prodej nabízí společnosti:

- Cetelem,
- HomeCredit,
- Essox.

1.6.2 On-line metody

Metoda vhodná při prodeji zboží a služeb, kdy je vyžadována okamžitá odezva. Výhodou je provedení transakce v řádu několika vteřin. Nevýhodou je naopak složitější implementace a vyšší náklady na provoz.

Platební karta

Platby platebními kartami umožňují v Česku obě největší kartové asociace Visa a MasterCard pomocí systému 3D Secure[2]. Jedná se o zabezpečený způsob, kdy zákazník neposkytuje informace o platební kartě přímo obchodníkovi, ale platební bráně banky. Systém poskytují banky:

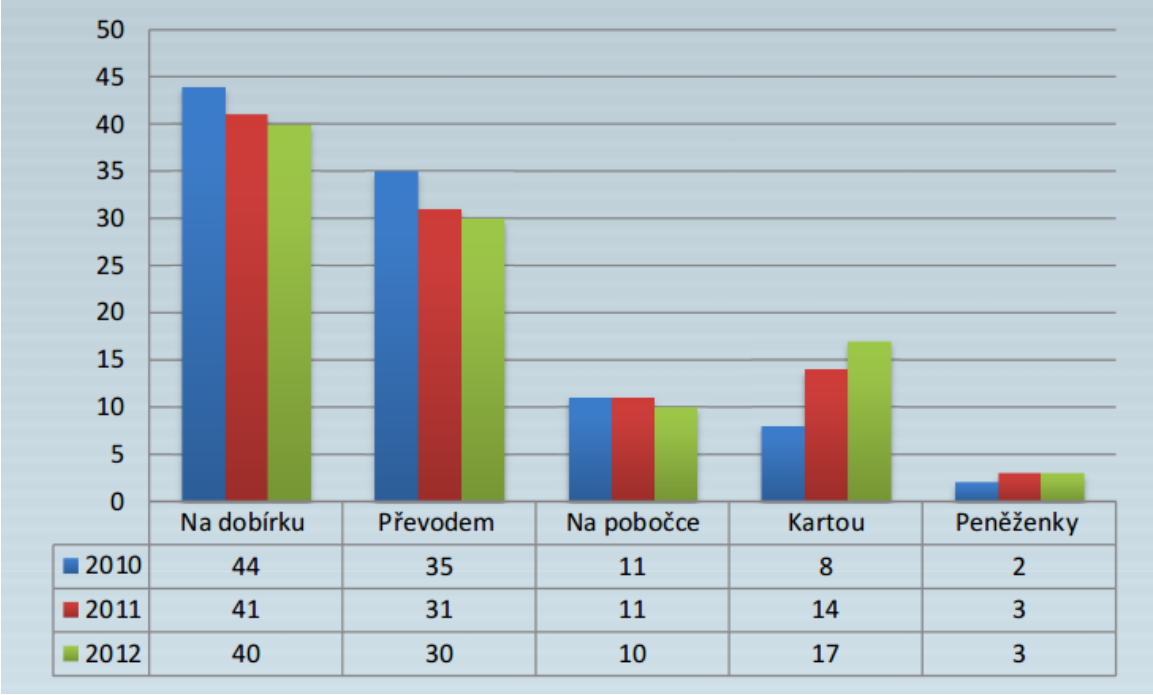
- Česká spořitelna,
- ČSOB,
- Komerční banka,
- Raiffeisenbank,
- UniCredit Bank.

[2] Global Payments Europe - http://www.globalpaymentsinc.com/Europe/Czech/index.html

Pro obchodníky jsou zde poměrně nepříjemné náklady ve formě procenta z každé transakce, případně implementační a měsíční poplatek. Na obrázku číslo 10 je patrná vzrůstající obliba platebních karet, zejména ve spojitosti se současným trendem, kterým je nákup na slevových portálech. [18]

„Platba je zatížena transakčním poplatkem v řádu jednotek procent, v případě českých internetových obchodníků obvykle mezi 1 až 3 procenty. Některé banky účtují další poplatky za aktivaci obchodníka nebo měsíční poplatky," dodává ve studii APEK. Podle studie vnímá více než 75 % obchodníků platbu kartou za nejnákladnější. [19]

Obrázek 10 -Srovnání oblíbenosti platebních metod v letech 2010 až 2012. [20]



	Na dobírku	Převodem	Na pobočce	Kartou	Peněženky
2010	44	35	11	8	2
2011	41	31	11	14	3
2012	40	30	10	17	3

Platební brána

Základem je platební brána napojená na internetové bankovnictví, kde na rozdíl od klasického bankovního převodu dostává obchodník informaci o stavu transakce okamžitě pomocí komunikačního rozhraní. V případě, že má obchodník účet u banky, která bránu poskytuje, jsou mu peníze připsány do několika hodin. Opět je každá transakce zatížena poplatky v jednotkách procent. Vlastní platební brány nabízí:

- Česká spořitelna,
- ČSOB,
- Komerční banka,
- Poštovní spořitelna,

- mBank,
- Raiffeisenbank.

Elektronická peněženka

Tyto peněženky fungují na principu předplaceného účtu nebo karty, kam si zákazník nejprve zašle finanční obnos a poté ho využívá k placení on-line. Mezi nejznámější ve světě patří PayPal, PayPay nebo MoneyBookers. V Česku provozuje ČSOB službu PaySec, která je poměrně rozšířená a její obliba stále roste.

Platební agregátor

Agregátory jsou speciální platební brány propojené s internetovým bankovnictvím, které sjednocují více platebních metod do jedné. Velkou výhodou je, že provozovatel implementuje v podstatě pouze jednu platební bránu, která mu umožní využívat téměř všechny dostupné on-line metody. V Česku službu nabízí PayU, která v současnosti agreguje 14 platebních metod[3]. Opět zde obchodník platí procenta z provedené transakce.

M-platba

Platby probíhají prostřednictvím webového či wapového rozhraní mobilního telefonu či internetu. Službu nabízejí všichni operátoři a mají stanoveny různé limity (T-Mobile 700 Kč, Telefonica 02 1.500 Kč a Vodafone 600 Kč). Společným projektem operátorů je služba Plať mobilem, pomocí které lze platit např. jízdenky MHD, parkovné, internetovou inzerci nebo sázky.

Premium Rate SMS

Jedná se o platby pomocí mobilního telefonu, kdy zákazník platbu provede zasláním speciální SMS zprávy. Službu nabízejí všichni operátoři nebo tzv. agregátoři a je limitována horní hranicí 600 Kč. Poplatky za transakce jsou poměrně vysoké a obchodník může ve výsledku obdržet pouze 30% z celkové částky.

Mobilní aplikace

Nejnovějším způsobem plateb na internetu jsou brány, které uživatel ovládá pomocí svého chytrého telefonu. Po nainstalování speciální aplikace a její aktivace (souhlas s inkasem, propojení s platební kartou apod.) stačí při platbě zadat pouze své telefonní číslo. Aplikace zpracuje požadavek ze strany obchodníka a zákazník pouze schválí provedení transakce zadáním PIN. V současnosti lze v Česku využít tyto dvě aplikace:

- Mobito,
- MasterCard Mobile.

[3] http://www.payu.cz/platebni-metody-pro-e-shopy

1.6.3 Shrnutí

Na základě studie APEK je pro systém internetového obchodu životně důležitá podpora nejrozšířenějších off-line plateb, kterými jsou dobírka, platba převodem a v hotovosti. Splátkový prodej může být zajímavý bonus pro obchodníka ve formě provizí, a proto by implementace alespoň jedné úvěrové společnosti mohla být přínosem.

V segmentu on-line plateb je nutností implementace platebních karet pomocí systému 3D Secure. Vhodným kandidátem je také agregátor plateb PayU, jehož implementací rozšíříme systém o podporu 14 platebních metod. Velký potenciál do budoucna a náskok před konkurencí by mohla zajistit platba pomocí mobilní aplikace Mobito.

Systém by měl být navržen modulárně, s ohledem na budoucí rozšíření, protože požadavky na platební metody postupně se mění.

1.7 Způsoby dopravy

Stejně důležitou součástí nákupního procesu, jakou je volba platební metody, je způsob doručení zboží. Jedná se v podstatě o pevně závislé prvky, protože obvykle každý dopravce nebo výdejní místo nabízí jiné možnosti platby. Systém by tedy měl v první řadě umožňovat sestrojit matici, pomocí které určíme aktivní platební metody pro jednotlivé zásoby doručení a opačně.

Obrázek 11- Nejvyužívanější dopravci v ČR v roce 2012 [21]

1.7.1 Dopravci

Na trhu existuje velké množství dopravních společností a systém by při jejich výběru neměl obchodníka nijak limitovat. Někteří dopravci ale limitují dopravní služby pouze na omezenou množinu PSČ, kdy při jejich využití musí internetový obchod implementovat validaci dostupnosti na zadané adrese zákazníkem (Česká pošta – Balík na poštu, PPL večerní doručení apod.).

„Výsledky studie ukazují, že i přes velký výběr přepravců, je stále nejvýznamnějším přepravcem Česká pošta, kterou následují soukromí přepravci PPL, TopTrans a DPD," uvádí APEK ve zprávě pro média. [21]

Hmotnost a rozměry

Každý sortiment má jiné požadavky na přepravu a systém internetového obchodu by je měl plně reflektovat pomocí atributů produktu. Administrace by měla umožňovat editaci váhových a velikostních limitů jednotlivých způsobů dopravy. Tím lze systémově docílit výběru vhodného dopravce, případně výpočtu ceny zásilky dle obsahu nákupního košíku zákazníka.

1.7.2 Odběrná místa

Ve městech se vzrůstající popularitě těší odběrná místa, kde si zákazník zboží vyzvedne a zaplatí podobně jako na poště, ale bez čekání a obvykle za minimální poplatek. Velké internetové obchody (Mall.cz, Alza.cz, CZC.cz apod.) mají již vybudovanou síť vlastních odběrných míst, zpravidla v každém krajském městě. Ostatní prodejci mohou využít následující otevřené sítě, které se postupně rozšiřují.

- **Heureka Point** – Síť výdejních míst Uloženka.cz, kterou přímo nabízí portál Heureka.cz. V současné době má 9 poboček.

- **PPL Partner** - Síť nezávislých provozoven, které umožňují příjem a výdej zásilek dopravce PPL. Příjemce se nemusí vázat na určitý čas doručení, pouze na otevírací dobu partnera. V současnosti obsahuje řádově desítky poboček a do konce roku 2013 jich má být cca 200.

- **Kouzelná almara** - Síť výdejních automatů. Prozatím jich je v provozu patnáct v nejnavštěvovanějších nákupních centrech v Praze. Zákazník si při nákupu vybere, do kterého z automatů služby si přeje zboží nechat uložit. Kouzelná almara slibuje uložení zboží do vybraného automatu už do 24 hodin od objednání v e-shopu. Jakmile bude zboží připraveno k vyzvednutí, obdrží zákazník prostřednictvím SMS unikátní PIN, kterým otevře příslušnou přihrádku. Kouzelná almara kromě doručení zboží je schopna i inkasovat platbu, protože součástí každého automatu je i platební terminál. [22]

1.7.3 Shrnutí

Konkurenceschopný systém by měl obchodníkovi umožnit využití služeb nejpoužívanějších dopravců (Česká pošta a PPL), včetně validace dostupnosti dle PSČ a napojení na jejich systémy umožňující hromadné online podání zásilek. Obchodníkovi tím odpadne režie spojená s vyplňováním údajů a podacích lístků. Pro konkurenční výhodu je vhodné implementovat podporu stále se rozšiřujících sítí odběrných míst Heuréka Point a PPL partner.

1.8 Rozšiřitelnost aplikace

Konkurenceschopný systém pohánějící internetové obchody by v žádném případě neměl být uzavřeným ekosystémem. Už z podstaty, že slouží k obchodování na internetu, kde se vše neustále rychle mění a vyvíjí. Dalším faktorem je, že každý obchodník má specifické požadavky, systém si pořizuje pro maximální usnadnění práce a zefektivnění svého podnikání. To jsou důvody, proč by samotný systém měl být vytvořen modulárně na živé platformě pro možnost budoucích úprav a napojení na systémy třetích stran.

1.8.1 Propojení s ekonomickými a informačními systémy

Integrace s účetním, ekonomickým (EIS) anebo manažerským (ERP) systémem posouvá práci se správou obchodu do zcela jiné dimenze. Takzvané mosty mezi informačním systémem a systémem e-shopu slouží k automatizované komunikaci a obousměrnému přenosu dat jako jsou objednávky, zboží, zákazníci a faktury. Úkolem je synchronizace dat na obou stranách a využití funkcí ERP/EIS pro maximální zefektivnění práce a snížení nákladů na správu obchodu. Integrace se realizuje dle možností konkrétního softwaru, požadavků dodavatele ERP pomocí webových služeb nebo přenosu XML/CSV souborů. Mezi nejpoužívanější systémy v ČR patří např.

- Pohoda,
- Altus Vario,
- Helios (Red, Orange, Green),
- SAP (All in one, Business one),
- Money S3,
- Duel,
- a další.

Výhody integrace s ERP

- Správa dat z jediného místa/systému.
- Zvýšení efektivity práce, snížení nákladů.
- Automatické projevy změn produktových karet v e-shopu.
- Automatické zprostředkovávání objednávek.

1.8.2 Napojení na dodavatele

Podobně jako napojení na ERP přináší propojení na velkoobchodního dodavatele způsob, jak minimalizovat práci s katalogem produktů a tím ušetřit čas a náklady. Obvykle se jedná o automatizovanou aktualizaci cen nebo skladových zásob z ceníku dodavatele. Aktualizace probíhá implicitně na pozadí, obvykle jednou denně, záleží na druhu sortimentu a požadavcích dodavatele.

Propojení se dá využít i pro jednorázový dávkový import do katalogu produktů, kterým se administrátorům ulehčí práce při plnění katalogu. Mohou se poté soustředit pouze na informační strukturu a SEO. Pro tyto účely se v praxi hojně využívají XML soubory určené pro export dat do cenových srovnávačů. Konkurenční výhodou systému je umožnit jednoduché napojení na tyto soubory přímo z administrace obchodu.

1.9 Servis a podpora

Z důvodu požadavků na rozšiřování, integraci a individuální úpravy roste potřeba komunikovat s autorem nebo implementátorem systému. Pokud má být internetový obchod konkurenceschopný, měl by se o jeho chod a další vývoj starat tým odborníků (marketingový specialisté, vývojáři atd.). Pro snadnou komunikaci mezi obchodníkem, dodavatelem systému a jeho administrátory by měl být zaveden jednotný systém podpory, tzv. HelpDesk nebo ticketovací systém.

2 Možnosti pořízení internetového obchodu

Existuje několik základních způsobů, jak systém provozující internetový obchod pořídit, a to lze buď zcela zdarma, nebo za úplatu. Každý z nich s sebou nese určitá negativa i pozitiva a hodí se pro jiný rozměr nebo fázi podnikání.

2.1 Open source systémy

Jednou z možností, jak lze internetový obchod získat téměř zdarma (vyžaduje webhosting a vlastní doménu), je instalací otevřeného systému. K této variantě se uchylují především začínající podnikatelé, kteří si chtějí obchod nejprve vyzkoušet bez velkých počátečních investic. Nízká cena obvykle bývá kompenzována časem stráveným instalací a nutnými úpravami, pro které je vyžadována větší počítačová gramotnost, znalost anglického jazyka a někdy i znalosti programování.

Open source e-shopů s českou lokalizací existuje celá řada, mezi nejznámější patří:

- PrestaShop,
- osCommerce,
- VirtueMart (rozšíření redakčního systému Joomla!),
- Quick.Cart,
- Ubercart (rozšíření pro redakční systém Drupal). [23]

Nevýhodou je omezený výběr rozšíření tzv. pluginů, které nemusí přesně odpovídat potřebám obchodování. Obchodník se poté musí smířit s unifikovaným vhledem, nastavením a neexistencí odpovědnosti autora systému. Dále se tyto systémy vyznačují generováním nekvalitního HTML kódu, což může vést k horším výsledkům při optimalizaci pro vyhledávače. Známá je také špatná kvalita zdrojových kódů, kdy může být problém s udržitelností při instalaci velkého počtu rozšíření. Otevřenost kódu s sebou nese také bezpečnostní rizika, ale na druhou stranu může být výhodou pro možnost libovolných úprav, kdy majitel není svázán komerční licencí.

2.2 Pronájem formou SaaS

Placené systémy formou SaaS (software jako služba) poskytují obchodníkovi největší bezpečí a komfort. Obchodník většinou platí formou měsíčního nájmu a jejich cena se velmi různí. Může se jednat o šablonové řešení zdarma nebo o profesionální řešení splňující nejnáročnější požadavky za desítky tisíc měsíčně. U těchto systémů může obchodník narazit na omezenou funkčnost, za jejíž rozšíření si bude muset připlatit.

2.2.1 Software as a Service

Jedná se o model nasazení softwaru, kdy je aplikace hostovaná provozovatelem služby a zákazník vůbec nemusí řešit hardware potřebný k chodu aplikace. Za službu se platí zpravidla periodicky, obvykle měsíčně a k aplikaci přistupuje přes webové rozhraní s integrací moderních Web 2.0 služeb. Spolu se systémem si zákazník kupuje i kompletní servis, zahrnující aktualizace, zálohování nebo technickou podporu.

Model SaaS je přínosný pro zákazníka i dodavatele, protože společně sdílejí rizika i odměny. Jejich vztah je upraven v tzv. SLA (Service Level Agreement) smlouvě, kde jsou jednoznačně specifikovány parametry služby (systému) jako např. dostupnost, četnost upgrade, okamžitý servis, ceny atd. [24]

Přínosy modelu SaaS pro zákazníka

- Počáteční náklady jsou výrazně nižší.
- Náklady jsou předem známé a bez překvapivých navýšení.
- Součástí služby je průběžné vylepšování softwaru a podpora.
- Zákazník se může svobodně rozhodnout přejít k jinému dodavateli, pokud služba nesplňuje očekávané přínosy.
- Redukovat dodatečné náklady a zaměřit se na core-business.
- Zákazník ví, že nebude kupovat „shelfware“ – software, který bude pouze ležet na polici.

Přínosy modelu SaaS pro dodavatele

- Lepší přizpůsobení jednotlivým zákazníkům.
- Může lépe strukturovat upgrade softwaru a technickou podporu.
- Obchodní model se odvíjí od poskytované služby a jejího přínosu pro zákazníka. Dodavatel má tak lepší šanci, že si udrží zákazníky.
- Může uplatnit alternativní licenční model, který vede k přírůstkovému podnikání. [25]

Nevýhody a limitující faktory

- Úplná závislost na připojení k síti. V off-line režimu nelze aplikaci obsluhovat.
- Potenciální bezpečnostní a operační riziko, protože data nejsou pod interní kontrolou.
- Možné riziko až příliš velké závislosti na poskytovateli.

- Nižší míra customizace, protože aplikaci lze přizpůsobovat pouze omezeně, především konfigurací editovatelných parametrů v administračním rozhraní.

2.3 Komerční řešení

Hotové komerční řešení má své výhody, protože se obvykle jedná o vyzkoušený a odladěný produkt. Pořizovací cena může být poměrně vysoká, ale menší než u kompletního řešení na klíč. Spuštění do ostrého provozu bývá rychlé, v řádu týdnů, obvykle dle požadovaných úprav na míru, jako originální grafický design, napojení na dodavatele, ERP systémy apod.

Tyto systémy obvykle bývají postaveny na moderních technologiích a neustálým vývojem reflektují nejnovější trendy na webu. Dodavatelé mívají odpovídající zázemí a tým odborníků. Standardně poskytují systém technické podpory, a proto požadavky na libovolné úpravy bývají řešeny pružně a efektivně. Zakoupením hotového řešení od renomovaného dodavatele, obchodník obvykle získá i možnost profesionálního poradenství v oblasti internetového marketingu a reklamy.

2.4 Řešení na klíč

Programování systému od nuly na zelené louce je nejdražší variantou pořízení internetového obchodu. Toto řešení je určeno pro velké společnosti nebo již zaběhlé obchody s velkou návštěvností, které potřebují zefektivnit provoz a snížit systémové požadavky.

3 Srovnání konkurenčních systémů

V této kapitole bude provedeno vymezení kritérií konkurenčních systémů, které jsou nabízeny formou SaaS. Následně budou vybráni tři kandidáti, u kterých bude provedeno porovnání s implementovaným systémem Weseus III.

3.1.1 Kritéria výběru porovnávaných systémů

Nabídka na trhu je opravdu veliká, proto je nutné vymezit základní požadavky na systém, které omezí výběr vhodných kandidátů. Protože je cílem práce vytvoření systému, na kterém lze postavit konkurence schopný internetový obchod, tak jsou hlavní kritéria výběru následující:

1. Primární zaměření systému pro obchodování na českém trhu. Stoprocentní česká lokalizace systému, podpora lokálních marketingových služeb, platebních metod, dopravců a komunikace s dodavatelem v českém jazyce.

2. Profesionální systém, který není primárně určen pro tvorbu webových stránek zdarma. Umožňuje provoz na vlastní doméně, není limitován datovým tokem a zatížen reklamou provozovatele.

3. Možnost implementace originálního grafického designu vytvořeného na základě firemní identity a grafických manuálů obchodníka.

4. Odladěný a funkční systém, který může kvalitu podložit referencemi.

3.1.2 Vybrané konkurenční systémy

Pro výběr konkurenčních systémů byla použita výše uvedená kritéria. Samotný výběr kandidátů probíhal na základě dostupných informací na oficiálních webech provozovatelů. Do porovnání byly vybrány následující systémy:

- Simplia[4],
- Shoptet[5],
- FastCentrik[6].

[4] Simplia s.r.o. - http://www.simplia.cz/

[5] Cybergenics s.r.o. - http://www.shoptet.cz/

[6] Allegro Group CZ, s.r.o. - http://www.fastcentrik.cz/

3.1.3 Porovnání se systémem Weseus III

Každý z konkurenčních systémů je nabízen v několika cenových variantách, které se liší výší limitů, počtem dostupných rozšíření nebo úrovní technické podpory. Pokročilé funkce popsané v předchozích kapitolách bývají u všech poskytovatelů shodně nabízeny až v nejvyšších tarifech, proto má smysl porovnávat pouze je.

Protože je systém Weseus III stále ve vývoji, budou uvedeny dosud implementované funkce a předpokládané ceny. Výsledná cena v ostrém provozu bude určena především konfigurací VPS (Virtuální privátní server), dle platného ceníku[7].

Tabulka 2- Porovnání cen a základních vlastností tarifů konkurenčních systémů.

	Simplia	**Shoptet**	**FastCentrik**	**Weseus III**
Varianta	VIP	Entereprise	Gold	VPS Lite
Cena / měsíc	Od 2 590 Kč	1 990 Kč	2 290 Kč	Od 1200 Kč
Počet produktů	Do 20 000	Do 20 000	Do 15 000	Dle prostoru na disku
Počet uživatelů	50	Neomezeně	Neomezeně	Neomezeně
Prostor na disku	15 GB	Dle počtu položek	5 GB	30 GB
Datový tok	Neomezeno	Neomezeno	30 GB	Neomezeno
Zkušební verze	Ne	30 dní	14 dní	30 dní
Technická podpora	E-mail, telefon, MSN	E-mail, telefon	24/7 přes webové rozhraní	E-mail, telefon
CZ doména v ceně	Ano	Ne	Neuvedeno	Ne
Zálohování dat	Neuvedeno	Neuvedeno	Ano	50 Kč / 1 GB / měsíc
Osobní konzultant	Ano	Ne	Ano	Ano
Platba provize z obratu	Ne	Ne	Ne	Ne
Aktualizace	Za poplatek	Neuvedeno	Ano	Za poplatek
Unikátní grafický design	14 900 Kč	od 11 920 Kč	14 800 Kč	od 12 000 Kč

[7] http://www.vas-hosting.cz/virtualni-server.html

Srovnání cen

Cena za pronájem se u všech konkurenčních systémů pohybuje ve stejné hladině a všechny jsou shodně omezeny počtem položek katalogu produktů. Idea nasazení systému Weseus III je přesně opačná, tedy klienta nijak neomezovat. V případě nedostatečného výkonu nebo diskového prostoru, lze systém přesunout na vlastní virtuální server, kde je možné konfiguraci libovolně měnit dle potřeb zákazníka.

Tabulka 3- Porovnání funkcionalit konkurenčních systémů.

	Simplia	Shoptet	FastCentrik	Weseus III
Katalog produktů	Ano	Ano	Ano	Ano
Pokročilé vyhledávání	Ano	Ano	Ano (AJAX)	Ano (AJAX)
Cros-sell moduly	Ano	Ano	Ano	Ano
SEO optimalizace	Ano	Ano	Ano	Ano
Export pro cenové srovnávače	Ano	Ano	Ano	Ano
E-mailing	Ano	Ano	50 Kč/měsíc	Ano
Google analytics	Ano	Ano	Ano	Ano
Napojení na ERP	Pohoda, Money S3	Pohoda, adSystem	Pohoda, S3 Money, Vario	Pohoda, Helios Red
Napojení na dodavatele	Ano	Ne	Ano	Ano

Srovnání jednotlivých funkcí

Protože do srovnání byly vybrány opravdu kvalitní řešení, tak většinu požadavků všechny z nich dokážou splnit bez nutnosti úprav na míru. Systém Weseus III nabízí veškeré požadavky na moderní katalog produktů již v současné verzi. Z porovnávaných systémů nabízí srovnatelné AJAX filtrování a vyhledávání produktů pouze FastCentrik.

Všechna řešení nabízejí standartně napojení na srovnávače cen. Samozdřejmostí je analytický nástroj Google analytics a možnost editace faktorů sloužících pro optimalizaci pro vyhledávače. Součástí všech je i modul pro E-mail marketing. Pouze u systému FastCentrik je za minimální poplatek.

Podporované platební metody

Nejrozšířenější je podpora platebních bran PayU a PayPal. Předpokládanou platbu kartou pomocí brány 3D secure nabízí pouze Simplia a Weseus III, stejně jako párování

transakcí s internetovým bankovnictvím. Systém FastCentrik vyniká podporou všech na trhu dostupných splátkových společností.

Podpora dopravců

V podpoře dopravců a odběrných míst může systému Weseus III konkurovat pouze Shoptet, který nabízí shodně Heureka point, Balík na poštu a navíc odběrná místa sítě Zasilkovna.cz. Weseus III integruje naopak síť PPL partner a validaci PSČ pro večerní doručení.

Tabulka 4 -Porovnání podpory platebních metod a dopraních služeb.

	Simplia	**Shoptet**	**FastCentrik**	**Weseus III**
Platby	PayPal, Platební karty, párování s IB	PayPal, PayU	PayPal, PayU	PayU, Platební karty, párování s IB
Splátový prodej	Ne	Iplatba	Cetelem, Home kredit, Essox	Ne
Váhové limity	Neuvedeno	Ne	Ano	Ano
Online podání	Podací archy pro ČP a PPL	Export pro PPL	Export do aplikace PoštMistr	Online podání ČP, PPL
Odběrná místa	Neuvedeno	Heuréka Point, Zasilkovna.cz, Balík na Poštu	Balík na Poštu	Heuréka Point, PPL partner, Balík na Poštu

4 Vývoj a implementace aplikace

V této kapitole bude popsán postup vývoje systému internetového obchodu. Systém bude implementován jako nová verze redakčního systému Weseus doplněná o modul internetového obchodu. Vývoj bude rozdělen do několika základních fází, které postupně povedou k úspěšné realizaci a nasazení aplikace v ostrém provozu. Jednotlivé fáze vývoje aplikace jsou:

- analýza současného stavu systému,
- analýza požadavků,
- výběr vhodných technologií a nástrojů,
- návrh případů užití,
- návrh datových tříd a architektury systému,
- tvorba datového modelu,
- programování systému,
- testování,
- nasazení v praxi.

4.1 Analýza současného stavu systému

Redakční systém Weseus II je v praxi úspěšně nasazen na mnoha projektech a jeho vývojem se zabýval Ing. Karel Dytrych ve své diplomové práci[8]. Je napsán objektově a modulárně, bohužel v čistém PHP pomocí vlastních knihoven. Ve srovnání s moderními PHP Frameworky byly u použitých knihoven zjištěny tyto nedostatky:

- napsány bez využití jmenných prostorů na verzi PHP 5.2,
- chybějící podpora cachování, AJAXu a efektivní databázové vrstvy,
- implicitně neřeší bezpečnost,
- nelze využít komponentový přístup,
- neexistence dokumentace a aktivní komunity.

[8] Zdroj [26]

Generátor administrace

Velikým přínosem je naopak vytvořený generátor administračního rozhraní. Tento generátor umožňuje na základě modelu databáze vygenerovat plně funkční kostru administrace. Vygenerovaná aplikace je postavena na třívrstvém modelu a obsahuje PHP soubory pro obsluhu datového modelu i byznys logiky. Prezentační vrstva je tvořena dynamickými HTML šablonami a pro jednotný design jsou použity jednotné CSS styly.

Generátor byl vytvořen pro ulehčení tvorby nových modulů, kdy odpadne poměrně rutinní tvorba editačních formulářů pro vkládání a editaci nových záznamů. Programátor může soubory po vygenerování libovolně upravovat, měnit uspořádání jednotlivých prvků a ušetřený čas věnovat vývoji veřejné části aplikace.

4.2 Analýza požadavků

Sběr požadavků na systém je stěžejní částí celé analýzy. Jeho podceněním mohou nastat závažné komplikace a výsledný produkt nemusí splňovat očekávání zadavatele. Požadavky se dělí do dvou základních kategorií na funkční a nefunkční. Pro sběr požadavků se používají tři základní způsoby.

1. Konzultace se zadavatelem pomocí otevřené diskuze, kdy je důležité, aby analytik kladl cílené otázky. Poté se tato metoda stává velmi efektivní a vede i k odhalení skrytých požadavků.

2. Vyplnění dotazníků skupinou zainteresovaných osob. Získané informace jsou vhodné především pro ověření nebo zpřesnění informací získaných konzultacemi.

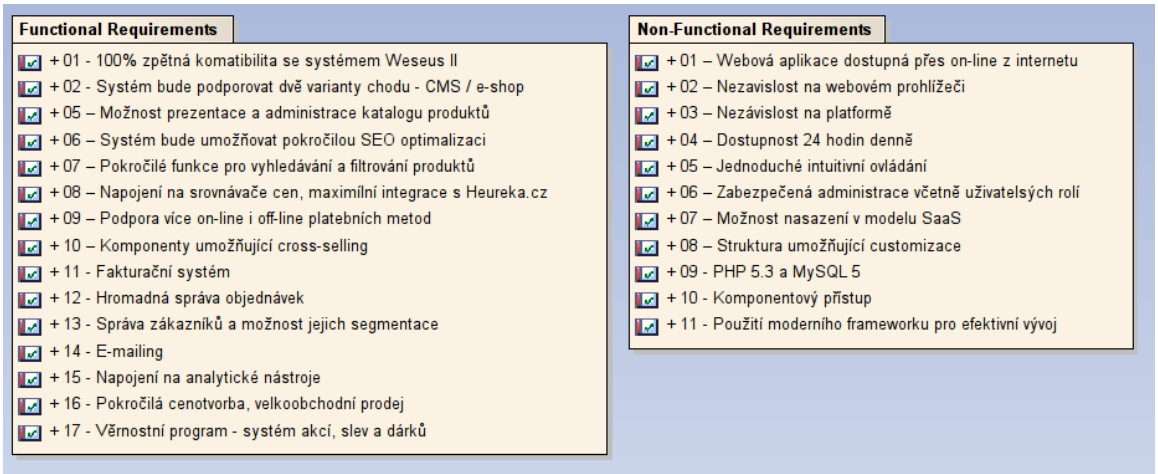
3. Použití dílny požadavků, která probíhá formou skupinové diskuze zainteresovaných osob. Všechny vyslovené požadavky se zapisují a zpřesňují v několika iteracích.

Při analýze systému bylo využito konzultací se zadavatelem, které vedly k získání převážně nefunkčních požadavků a uceleného pohledu na systém. Dále bylo využito principu dílny požadavků, které se účastnili potencionální zájemci o licenci systému. Poslední fází analýzy požadavků bylo porovnání konkurenčních systémů na trhu, popsané v teoretické části. Získané požadavky byly následně zpracovány v programu Enterprise Architect.

Charakteristika systému

Základním požadavkem je vytvoření internetového obchodu, který bude schopen nasazení v byznys modelu SaaS. Systém musí zachovat kompletní funkcionalitu CMS Weseus II a modul e-shopu musí jít jednoduše deaktivovat. Důvodem je požadavek zpětné kompatibility a možnosti upgrade již běžících instancí.

Obrázek 12- Diagram základních požadavků systému



4.3 Technologie pro návrh a implementaci systému

V této podkapitole bude proveden výběr a popis jednotlivých technologií pro analýzu, návrh a implementaci systému. Z nefunkčních požadavků vyplynulo zachování zavedené platformy označované jako LAMP, která bude popsána níže. Výběr ostatních technologií není nijak limitován. Při výběru bude upřednostněn software se svobodnou licencí, nebo nástroje používané v průběhu studia.

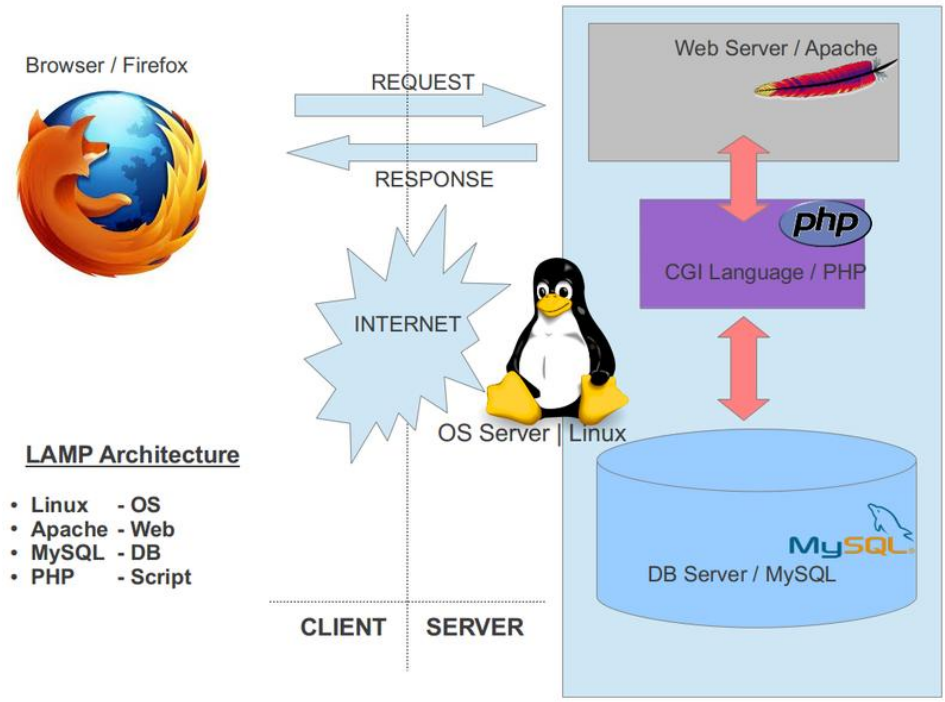
4.3.1 Platforma LAMP

Jedná se o zkratku symbolizující využití kombinace svobodného softwaru, která je složena z počátečních písmen těchto technologií:

- operačního systému Linux,
- webového serveru Apache,
- databázového systému MySQL,
- skriptovacího programovacího jazyka PHP.

Jednotlivé technologie na sobě nejsou nijak závislé a dají se nahradit alternativami (např. PHP lze nahradit jazyky Pearl nebo Python). Platforma je rozšířená pro implementaci dynamických webových stránek a využívá síťový model klient-server. Na obrázku číslo 13 je graficky znázorněna komunikace mezi jednotlivými technologiemi. Díky svému svobodnému charakteru je běžně nabízena hostingovými centry za poměrně příznivé ceny.

Obrázek 13- Schéma architektury platformy LAMP. Zdroj [27].



Skriptovací jazyk PHP

Jako programovací jazyk byl zvolen PHP. Důvodem je jeho rozšířenost při tvorbě dynamických webových aplikací a požadovaná zpětná kompatibilita s původním systémem. Pro implementaci bude použit ve verzi 5.3, která již obsahuje funkcionality běžné u vyspělých programovacích jazyků, jako např. jmenné prostory, anonymní funkce, nový garbage collector apod..

Hlavním nedostatkem tohoto jazyka je to, že od počátku nebyl navržen jako objektově orientovaný a ani později doplněná podpora OOP nedosahuje kvalit čistě objektových jazyků, jakými jsou např. Java nebo C#. Tento základní nedostatek lze vyřešit použitím třídy Nette\Object[9] jako základního předka všech objektů aplikace, která je součástí Nette frameworku[10] blíže představeného v kapitole 4.3.2. .

Relační databáze MySQL

MySQL je multiplatformní relační databáze. Patří mezi nejvíce rozšířené díky své svobodné licenci, rozšířenosti webhostingových služeb a podpoře jazyka PHP. Umožňuje použití několika úložných enginů, které se od sebe liší způsobem ukládání dat do souborů. Při vývoji aplikace je použit engine InnoDB, kvůli podpoře cizích klíčů a transakcí. Pro manipulaci, správu a organizování uložených dat využívá obecný

[9] http://doc.nette.org/cs/php-language-enhancements

[10] http://nette.org/cs/

dotazovací jazyk SQL, který je podobně jako u jiných relačních databází doplněn o specifická rozšíření.

4.3.2 Nette framework

Jedním z požadavků na systém bylo použití moderního PHP frameworku, který zefektivní vývoj aplikace, vyřeší nedostatky původních knihoven a umožní komponentový přístup. Nette framework byl zvolen, protože splňoval všechna tato kritéria a autor měl s jeho použitím více než tříleté praktické zkušenosti. K vývoji byla použita poslední stabilní verze 2.0.10.

Nette je ryze český open-source framework, kolem kterého vznikla velmi početná komunita vývojářů, jež ho neustále vyvíjí a píší pro něj různá rozšíření. Jeho původním autorem je David Grudl, který stále řídí vydávání nových verzí a v současnosti se snaží o jeho rozšíření do zahraničí.

Mezi největší přednosti Nette patří[11]:

- Promyšlený a čistý objektový návrh využívající nových vlastností PHP 5.3 a 5.4, komponent a událostmi řízeného modelování.
- Implicitně eliminuje výskyt bezpečnostních děr a jejich zneužití, jako je např. XSS, CSRF, session hijacking, session fixation atd.
- Podpora moderních technologií HTML5, AJAX nebo SEO.
- Efektivní databázová vrstva Nette\Database nezávislá na konkrétním typu databáze. Umožňuje také využití libovolné knihovny např. Dibi, NotORM nebo ORM frameworku Doctrine 2.
- Od verze 2.0 využívá návrhový vzor Dependency Injection (DI), který slouží pro snížení závislostí mezi jednotlivými částmi systému.
- Komponentový přístup, pomocí kterého lze dosáhnout maximální znovupoužitelnosti kódu. Možnost jednoduše využívat rozšíření napsané komunitou.[12]
- Ladící a testovací nástroje, které pomůžou rychle odhalit, logovat a opravit chyby.
- Kvalitní dokumentace, návody a nejaktivnější komunita v ČR.

[11] http://nette.org/cs/#toc-features

[12] http://addons.nette.org/cs/

4.3.3 Analytické a vývojové nástroje

Jazyk UML

K vizuálnímu návrhu systému byl zvolen grafický jazyk UML (Unified Modeling Language), který podporuje objektově orientovaný přístup k analýze, návrhu a popisu programových systémů.

Enterprise Architect

UML diagramy, byly modelovány pomocí nástroje Enterprise Architect[13] od společnosti Sparx system. Nástroj byl konkrétně použit při sběru požadavků, tvorbě Use-Case diagramů a modelování návrhových tříd. Zvolen byl z důvodu vlastních znalostí a zkušeností získaných v několika předmětech studia.

MySQL Workbench

Jedná se o aplikaci přímo od tvůrců databáze MySQL a je určená k její komplexní správě. Umožňuje připojení přímo na databázový server a následnou obousměrnou synchronizaci s modelem databáze. Tento nástroj byl použit při návrhu modelu databáze, tzv. ER diagramu. V rámci tvorby modelu byl MySQL Workbench použit k vytvoření databázových tabulek, jejich atributů a datových typů. Dále k definici primárních a cizích klíčů, integritních omezení a indexů při optimalizaci SQL dotazů.

Netbeans IDE

Pro efektivní vývoj aplikací je vhodné použití vývojového prostředí tzv. IDE (Integrated Development Environment), které dokáže programátorovi značně ulehčit práci našeptáváním kódu, funkcemi pro refaktoring kódu apod.. Netbeans IDE je univerzálním prostředím se svobodnou licencí, které lze nakonfigurovat dle vlastních představ instalací zásuvných modulů.

Při vývoji systému byla použita tato konfigurace:

- Netbeans IDE pro PHP ve verzi 7.1.1,
- modul pro Nette framework,
- modul pro konfigurační soubory Neon,
- modul verzovacího systému Git.

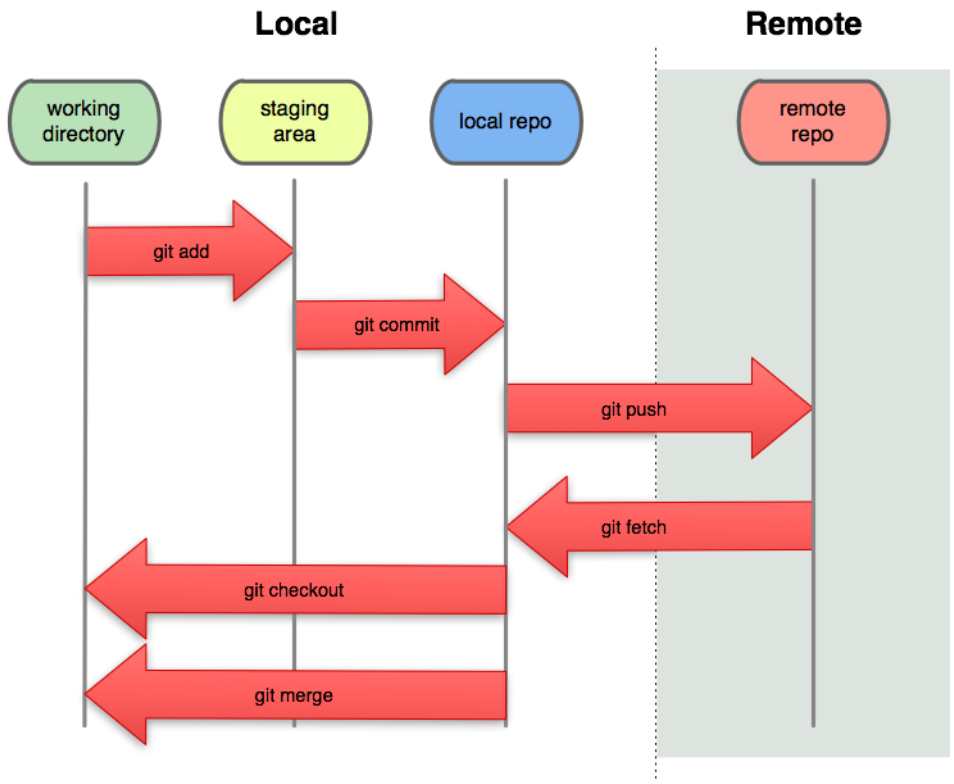
4.3.4 Verzovací systém Git

Jedná se o distribuovaný systém správy verzí pod svobodnou licencí. Obrovskou výhodou je, že vývoj probíhá lokálně a lze tedy plnohodnotně pracovat off-line. Při práci v týmu může současně každý z programátorů pracovat na libovolné části systému a po dokončení úprav nahrát úpravy do centrálního repozitáře, který je poté distribuuje

[13] http://www.sparxsystems.com/

ostatním. Použití Gitu přinese výhody i při vývoji jednotlivcem, protože zaznamenává kompletní historii kódu ve dvou verzích lokální „soukromé “ a centrální „veřejné“. Dále umožňuje vést vývoje ve více větvích a následně je slučovat.

Obrázek 14- Grafické znázornění příkazů verzovacího systému Git. Zdroj [28].



Při vývoji tohoto systému, byl Git použit pro vývoj jedním programátorem a jeho síly bylo využito především pro zpřehlednění vývoje a vedení historie kódu v repozitáři. Pro potřeby testování byla testovaná verze na testovacím produkčním serveru reprezentována pomocí publikace obsahu centrálního repozitáře. Tím bylo docíleno možnosti paralelního testování a lokálního vývoje. Odpadlo tím hlavně ruční kopírování souboru pomocí FTP klienta, které je zdlouhavé, protože se vždy musí zaktualizovat všechny upravené soubory.

Další výhodou držení projektu v repozitáři je možnost využití dependency managera, konkrétně nástroje Composer,[14] který bude detailněji představen níže.

4.3.5 Správce závislostí Composer

Composer je nástroj, který umožňuje pokročilou správu závislostí PHP knihoven použitých v aplikaci. Konfigurace se provádí v souboru *composer.json*, kde je možné deklarovat libovolně složité závislosti jednotlivých knihoven. Knihovny jsou

[14] http://getcomposer.org/

reprezentovány tzv. balíčky, které si nástroj sám stahuje z git, svn nebo mercurial repozitáře. Je možné využít veřejný centrální[15] repozitář, kam lze nahrávat i vlastní balíčky nebo privátní z vlastního serveru. Po deklaraci závislostí lze pomocí příkazové řádky jednoduše vytvořit novou instalaci aplikace nebo pouze zaktualizovat knihovny na novou verzi.

Příklad deklarace závislosti knihovny Nette frameworku s definicí vlastního repozitáře vedoucí na aktuální vývojovou větev.

```
  {
    "repositories": [
        {
            "type": "vcs",
            "url": "git://github.com/nette/nette.git"
        }
    ],
    "require": {
        "nette/nette": "dev-master",
    }
}
```

Příklad příkazu instalace aplikace Nette sandbox, která slouží jako funkční základ pro vývoj nové aplikace.

```
$ composer create-project nette/sandbox nazev-projektu
```

Při nasazení internetového obchodu v modelu SaaS narážíme na neustálou potřebu vývoje nových funkcí a zpětných aktualizací. Composer v kompinaci s Git repozitáři poskytuje vhodné řešení problému vývoje a aktualizace již spuštěných instancí aplikace. Systém je nutné rozdělit do samostatných knihoven a řádně deklarovat jednotlivé závislosti.

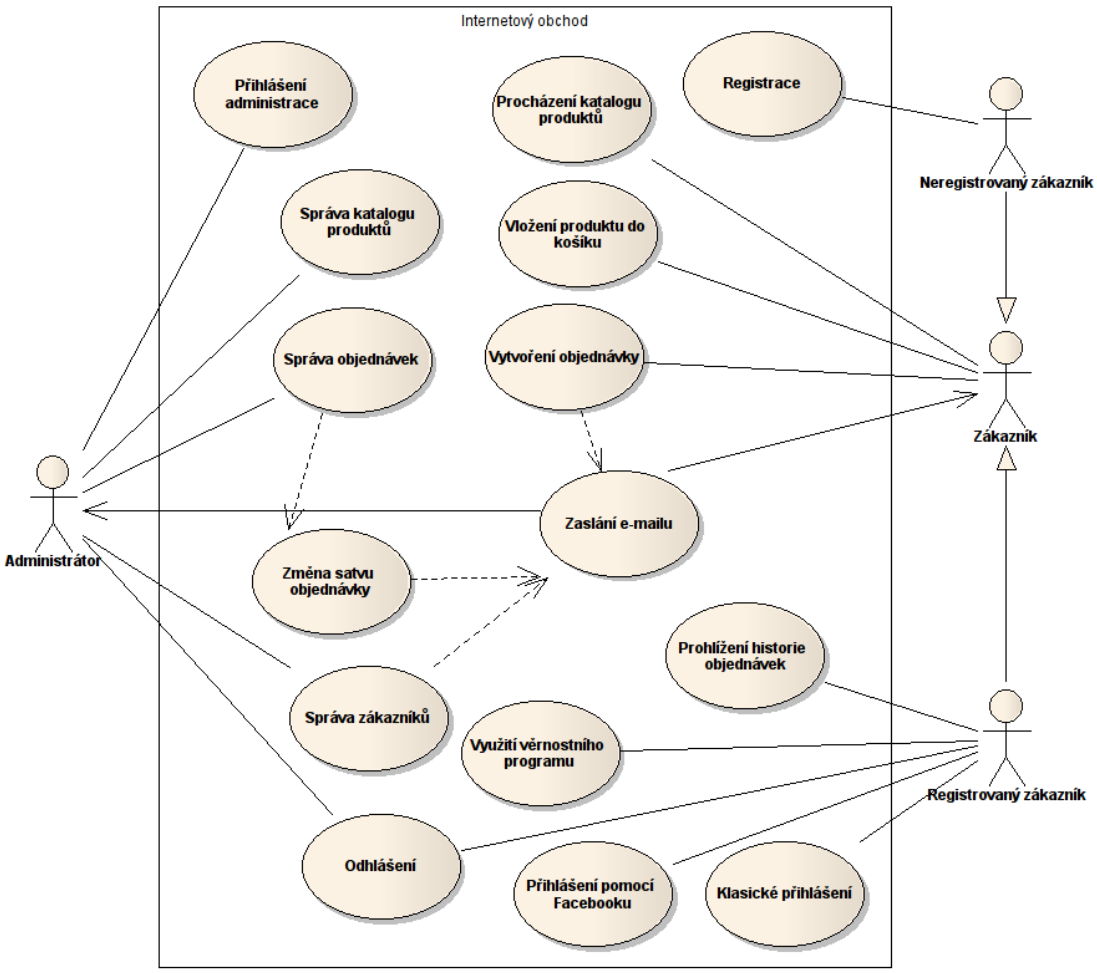
[15] https://packagist.org/

4.4 Návrh případů užití

Use-Case diagramy neboli případy užití znázorňují interakci mezi uživatelem a systémem. Diagramy případu užití jsou určeny k definici chování systému, obsahují soubor scénářů pro použití systému a znázorňují:

- aktéry systému,
- hranice modelovaného systému,
- konkrétní scénáře případů užití.

Obrázek 15 - Use Case diagram internetového obchodu.



4.4.1 Aktéři

Prvním krokem při návrhu případů užití je definice aktérů. Jedná se o externí entity, které používají navrhovaný systém nebo se dotýkají jeho hranice. Může se jednat o konkrétního uživatele, roli nebo jiný systém. Při návrhu byli na základě uživatelů určeni následující aktéři:

- administrátor systému,
- registrovaný zákazník,
- neregistrovaný zákazník.

Administrátor má přístup do zabezpečené administrační části a na základě přidělených oprávnění může provádět správu internetového obchodu. Administrační část je logicky rozdělena na moduly umožňující správu:

- katalogu produktů a obsahu stránek,
- objednávek a faktur,
- zákazníků,
- základního nastavení systému.

Aktér neregistrovaný zákazník vznikl z požadavku umožnění nákupu bez registrace, kdy při objednávce vyplní doručovací údaje a systém mu na pozadí nevytváří žádný uživatelský účet. Má možnost provést registraci a získat výhody registrovaného uživatele jako je např. věrnostní program, přehled historie a stavu provedených objednávek, rychlého přihlášení pomocí účtu na sociální síti Facebook apod..

4.5 Návrh datových tříd a architektury systému

Návrhový model tříd je reprezentován diagramem tříd, který poskytuje statický pohled na implementační třídy systému. Návrhový model by měl přesně specifikovat, jakým způsobem má navrhovaný systém vykonávat jednotlivé funkcionality. Třídy by měly být specifikovány tak, že na základě modelu by mělo být možné je přímo implementovat.

Měly by obsahovat definici:

- atributů (včetně jejich názvů, datových typů a oprávnění),
- operací (včetně všech parametrů, viditelnosti a návratových typů),
- vazeb mezi třídami.

V případě, že jsou třídy detailně specifikované, lze pomocí CASE nástrojů vygenerovat přímo zdrojové kódy obsahující funkční kostru modelovaných tříd. V praxi je někdy efektivnější přenechat návrh detailů až na samotnou implementaci programátorem. Lze tak dosáhnout řádově vyšší efektivity a odpadne přenášení případných změn mezi zdrojovým kódem a návrhovým modelem.

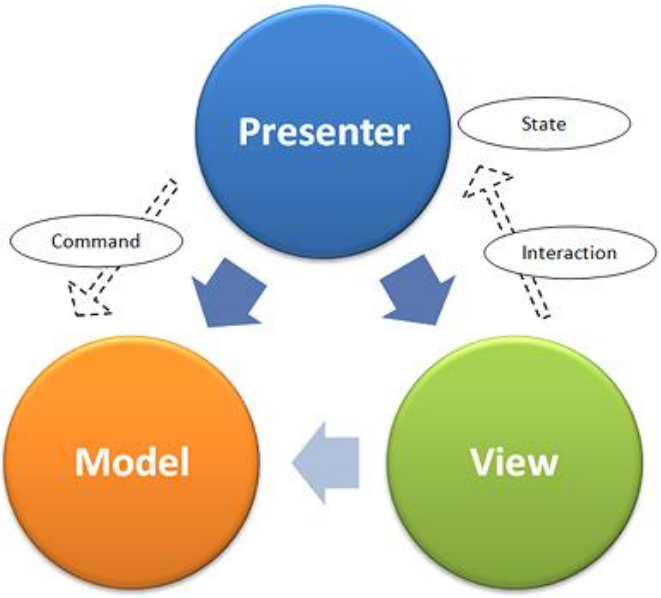
Návrh datových tříd internetového obchodu byl maximálně směrován dle architektury a standardů použitého Nette frameworku. Framework sám o sobě představuje aplikační základ s doporučenou strukturou datových tříd. Jeho architektura je založena na třívrstvém modelu aplikace MVP (Model-View-Presenter), návrhovém vzoru Dependency Injection a komponentovém přístupu.

Vytvořený návrhový model tříd vystihuje pouze základní informace a konkrétní detaily byly zpracovány až v průběhu implementace. Veškeré diagramy jsou obsaženy na přiloženém CD.

4.5.1 Architektura MVP

Jedná se o softwarovou architekturu oddělující uživatelské rozhraní, aplikační logiku adatový model do tří vrstev. Tím je dosaženo zpřehlednění struktury systému, která usnadňuje vývoj a testování jednotlivých částí samostatně. Je odvozena od architektury MVC a založena na konceptu ovládacích prvků, tzv. widgetů.

Obrázek 16- Schéma architektury Model-View-Presenter. Zdroj[29].



Vrstvy jsou reprezentovány následujícími částmi.

1. **Model** obsluhuje práci s datovým modelem a obsahuje základní business logiku aplikace.

2. **View** se stará o prezentaci informací a interakci s uživatelem. Oproti architektuře MVC navíc zpracovává uživatelský vstup.

3. **Presenter** obsluhuje aplikační a presentační logiku. Manipuluje s modelem a zajišťuje aktualizaci view, buď přímo, nebo pomocí systému notifikací.

4.5.2 Návrhový vzor Dependency Injection

Návrhový vzor Dependency Injection (DI) slouží pro snížení závislostí mezi jednotlivými částmi systému. DI je zjednodušeně řečeno předávání závislostí, kdy je třídám odebrána zodpovědnost za získávání objektů, které vyžadují ke své činnosti. Implementovat lze předáváním potřebných objektů přímo v konstruktoru třídy, tzv. setterem[16] nebo parametrem volané metody. Použitím DI se vyhneme nežádoucím globálním proměnným, statickým metodám nebo návrhovému vzoru Singleton.

Nette DI container

Základem aplikace postavené na Nette frameworku je systémový DI kontejner, který poskytuje všechny potřebné služby a parametry. Tvoří ho třída Nette\DI\Container, která pružně zajišťuje, že se instance služeb vytváří pouze jednou. Kontajner běžně neobsahuje pouze služby frameworku (připojení k databázi, sessions, cache apod.), ale také služby vlastních nebo přidaných knihoven.

Nette Configurator

K definici služeb využívá Nette vlastní konfigurační jazyk Neon[17]. Programátor tedy nakonfiguruje služby pomocí souborů s příponou .neon a framework se poté pomocí třídy Nette\Config\Configurator postará o vygenerování PHP kódu kontejneru, který uloží do cache. Při konfiguraci lze využívat dědičnosti služeb, factories nebo auto-writingu popsaného níže.

Auto-writing

Automaticky předává požadované služby konstruktoru nebo jiným metodám definované služby. Podmínkou je, že služba musí být definovaná výše a musí být v kontejneru právě jedna. Programátor se tedy vůbec nemusí starat o vypisování závislostí do konfiguračního

[16] Metoda určená pro nastavení konkrétního atributu třídy. Např. $this->setId(1) nastaví atribut Id na hodnotu 1.

[17] Jednoduchý konfigurační jazyk podobný YAML nebo JSON. http://ne-on.org/

souboru. Stačí, aby kód konstruktoru řádně okomentoval, protože se auto-writing řídí podle *type hintů* a anotací *@return*.

Factories

Factories neboli tzv. továrničky, se definují podobně jako služby s tím rozdílem, že jim lze předávat parametry. Proto se při jejich zavolání v aplikaci vytváří vždy nová instance.

Příklad zápisu konfiguračního souboru neon.

```
services:

    database:

        class: Nette\Database\Connection(%dsn%, %user%, %password%)

        setup:

            - setCacheStorage(@cacheStorage)
```

Předchozí konfigurace vede k vygenerování tohoto PHP kódu.

```
function createServiceDatabase()

{

    $service = new Nette\Database\Connection(...);

    $service->setCacheStorage($this->cacheStorage);

    return $service;

}
```

4.5.3 Komponentový přístup

Nette framework byl pro tvorbu systému zvolen i díky této funkcionalitě, která umožňuje maximální znovupoužitelnost kódu a snadné zakomponování již vytvořených knihoven komunitou.

Komponenta představuje vykreslitelný objekt neboli widget, který má svojí vnitřní logiku a lze ho použít i mimo aplikaci. Typicky se jedná například o menu, výpis posledních novinek, ankety, formuláře apod.. Pomocí Nette jde komponenty vytvářet poměrně snadno, jako samostatné třídy, jež jsou stejně jako všechny presentery potomkem Nette\Application\UI\Control. Každé komponentě lze přiřadit šablonu a tím lze ovlivnit její vzhled.

4.6 Návrh databáze

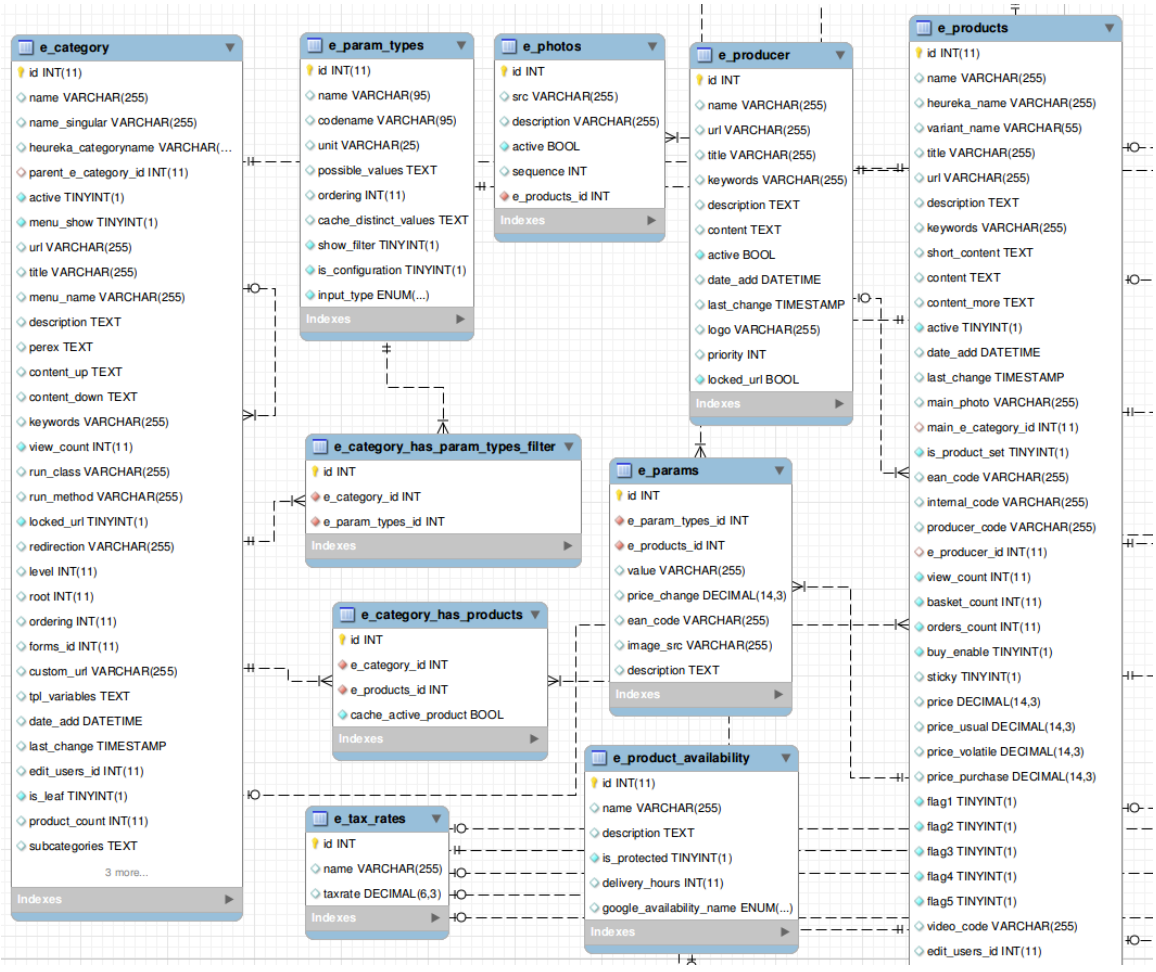
Dalším důležitým krokem tvorby systému je návrh datového modelu. Na kvalitě návrhu databáze závisí úspěšnost celého projektu. Databáze musí zajistit správné ukládání

a konzistenci persistentních dat, které budou ukládány do relační databáze. Pro zobrazení modelu se používá ER (Entity-relationship) diagram. Před samotným návrhem je nejdříve nutné přesně specifikovat, jaká data bude aplikace uchovávat a jakým způsobem s nimi bude manipulovat. Návrh datového modelu poté probíhá na ve dvou fázích:

1. Návrh konceptuálního datového modelu, jehož úkolem je popsat datovou základnu pomocí entit. Nevztahuje se k žádnému konkrétnímu databázovému systému. Skládá se z entit, vazeb mezi nimi, kardinalit a klíčů.

2. Druhou fází je návrh fyzického modelu, kdy jsou doplněny elementy konkrétního databázového systému, jako jsou cizí klíče, indexy, pohledy nebo procedury.

Obrázek 17- Výřez ER diagramu zobrazující produkt a dynamické parametry.



Při návrhu databáze systému Weseus III byl kladen důraz na dodržení 3. normální formy, díky které dosáhneme konzistence a integrity dat. Vzhledem k tomu že, systém měl zachovat data a funkcionalitu redakčního systému, byla původní databáze zachována a doplněna o tabulky internetového obchodu.

Pro jasné oddělení tabulek modulu e-shopu byla zavedena konvence jejich pojmenování s prefixem ve tvaru *e_anglicky_nazev*. Na obrázku číslo 17 je zobrazen výřez diagramu popisující základní entity katalogu produktů, kterými jsou produkt, jeho dynamické parametry, kategorie produktů a výrobce.

Protože výsledný diagram obsahuje řádově desítky tabulek a nelze ho celý zobrazit ani v příloze, bude obsažen v elektronické podobě na přiloženém CD.

4.7 Implementace systému

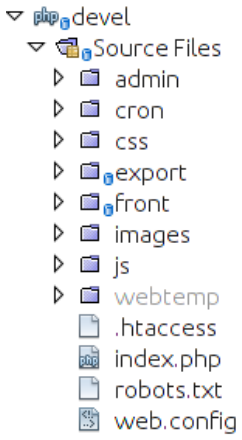
Pro implementaci systému byly použity technologie představené v kapitole číslo 4.3 a pro uchovávání dat je využíván datový model, který byl navržen v předchozí kapitole. V této kapitole bude popsán postup vývoje a struktura systému.

4.7.1 Adresářová struktura

Systém Weseus III se skládá ze dvou hlavních částí, veřejné a zabezpečené. Ve struktuře jsou reprezentovány adresáři *front* a *admin* a jejich vývoj bude odděleně popsán v následujících podkapitolách.

Další důležitou částí systému jsou skripty uložené v adresáři *cron*, které jsou periodicky volané plánovačem úloh[18]. Starají se např. o XML exporty pro externí systémy, uložené v adresáři *export*, aktualizace katalogu produktů, ukládání cenové historie apod..

Obrázek 18- Základní adresářová struktura aplikace.



V adresářích *js*, *images* a *css* jsou uloženy zdrojové soubory použitých javascriptových knihoven, obrázků a kaskádových stylů, které tvoří grafickou šablonu e-shopu.
V produkčním režimu jsou tyto soubory spojeny do jednoho a minifikovány pomocí komponenty WebLoader[19], která pro ukládání souborů využívá adresář *webtemp*. Tento

[18] http://interval.cz/clanky/jak-na-demona-cron/

[19] http://addons.nette.org/cs/webloader

adresář slouží i pro ukládání dočasných miniatur všech prezentovaných obrázků. Systém zobrazuje pouze na míru vygenerované kopie obrázků, které lze jednoduše doplnit ovodoznak, a originální soubory jsou bezpečně uloženy v administrační části.

4.7.2 Front-end

V požadavcích na Front-end neboli veřejnou část systému bylo zachování původních funkcí CMS Weseus II. Jelikož modul e-shopu svým rozsahem převyšoval samotné CMS, byla z důvodu efektivity vývoje a další udržitelnosti kódu zvolena varianta jeho kompletního přepsání pomocí Nette frameworku. Implementace frontendu tedy probíhala v těchto fázích:

- Přepsání jádra CMS,
- Testování a odladění nové verze CMS (včetně nasazení v ostrém provozu),
- Implementace modulu internetového obchodu.

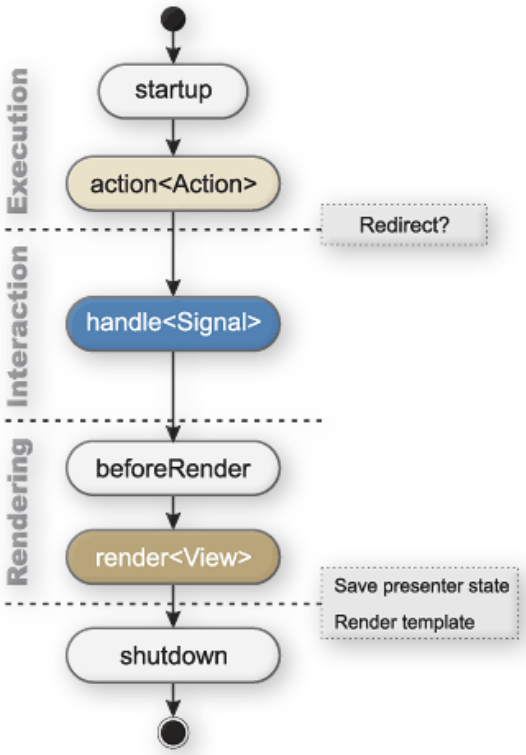
Při programování této části aplikace byla využita architektura představená v kapitole 4.5, založená na modelu MVP, komponentách a Dependency Injection. Pro rozdělení aplikace do logických celků bylo využito následujících jmenných prostorů:

- FrontModule,
    - FrontModule \EshopModule,
- Models,
    - Models\EshopModels,
- Weseus,
    - Weseus\Core,
    - Weseus\Security,
    - Weseus\Routers,
    - Weseus\Helpers,
    - Weseus\Mail,
    - Weseus\Forms,
    - Weseus\ Components,
    - Weseus\ Cron.

Presentery

Presentery jsou třídy, které na základě požadavku od uživatele volají příslušnou aplikační logiku a předají view výsledky určené pro zobrazení. Pro zpracování požadavků obsahují speciální metody nazvané *action* a *signal*. Action jsou metody, které slouží k vykreslení určité stránky (Latte šablony[20]) a signály jsou subrequesty určené zejména pro AJAX požadavky. Protože presentery neumí zpracovávat HTTP požadavky přímo, používá Nette framework pro jejich obousměrný překlad na URL adresy speciální třídy, tzv. Routery[21]. Třídy obsluhující routování aplikace jsou sjednoceny ve jmenném prostoru *Weseus\Routers* a konfigurace jejich pravidel se nachází ve spouštěcím souboru *bootstrap.php*. Na obrázku číslo 19 je zobrazen životní cyklus presenteru.

Obrázek 19- Životní cyklus presenteru Nette frameworku. Zdroj [30].



Moduly

Při vývoji bylo využito podpory modulární struktury, kdy lze presentery rozdělit do podadresářů a jim příslušných jmenných prostorů. Routovací systém nejprve vyhodnocuje volání správného modulu, poté presenteru a jeho akce. Systém je tvořen

[20] http://doc.nette.org/cs/templating

[21] http://doc.nette.org/cs/routing

moduly FrontModule a EshopModule reprezentující presentery a šablony CMS a modulu e-shopu.

Šablonovací systém Latte

Pro tvorbu dynamických HTML šablon bylo použito systému Latte, který je součástí Nette frameworku. Pomocí speciální syntaxe umožňuje jednoduše a bezpečně prezentovat data. Automaticky výpis ošetřuje funkcí *htmlSpecialChars* a zabezpečuje před útoky jako XSS[22]. Syntaxe se skládá z maker[23], pomocí kterých lze tvořit podmínky, cykly nebo využívat např. dědičnost šablon a helperů, které slouží pro modifikaci dat odkazovanou PHP funkcí. Latte obsahuje celou řadu základních helperů[24] pro práci s řetězci, formátování data apod.. Umožňuje také tvorbu vlastních maker a helperů.

Vlastních helperů (namespace *Weseus\Helpers*) bylo použito například při výpočtu ceny produktu u konkrétní cenové skupiny pro registrované velkoobchodní partnery. Následující zápis vypíše konkrétní cenu včetně znaku aktuálně nastavené měny.

```
{$product|price|currency}
```

Komponenty

Systém je tvořen desítkami znovupoužitelných komponent, které jsou implementovány samostatnými třídami se jmenným prostrem *Weseus\ Components*. Každá komponenta se před voláním v šabloně musí nejprve zaregistrovat v příslušném presenteru. Pro registraci komponent, které je třeba volat v libovolné části aplikace, byl zvolen společný předek všech presenterů *WeseusPresenter.*

Příklad registrace komponenty obsluhující vyhledávání s našeptáváním.

```
/**
 * Továrnička pro tvorbu komponenty SearchInput.
 * Vykresluje šablonu SearchInput.latte.
 * @return \SearchInputControl
 */
protected function createComponentSearch() {
    $search = new SearchInputControl();
```

[22] Cross-site scripting (XSS) je metoda narušení WWW stránek využitím bezpečnostních chyb ve skriptech (především neošetřené vstupy).

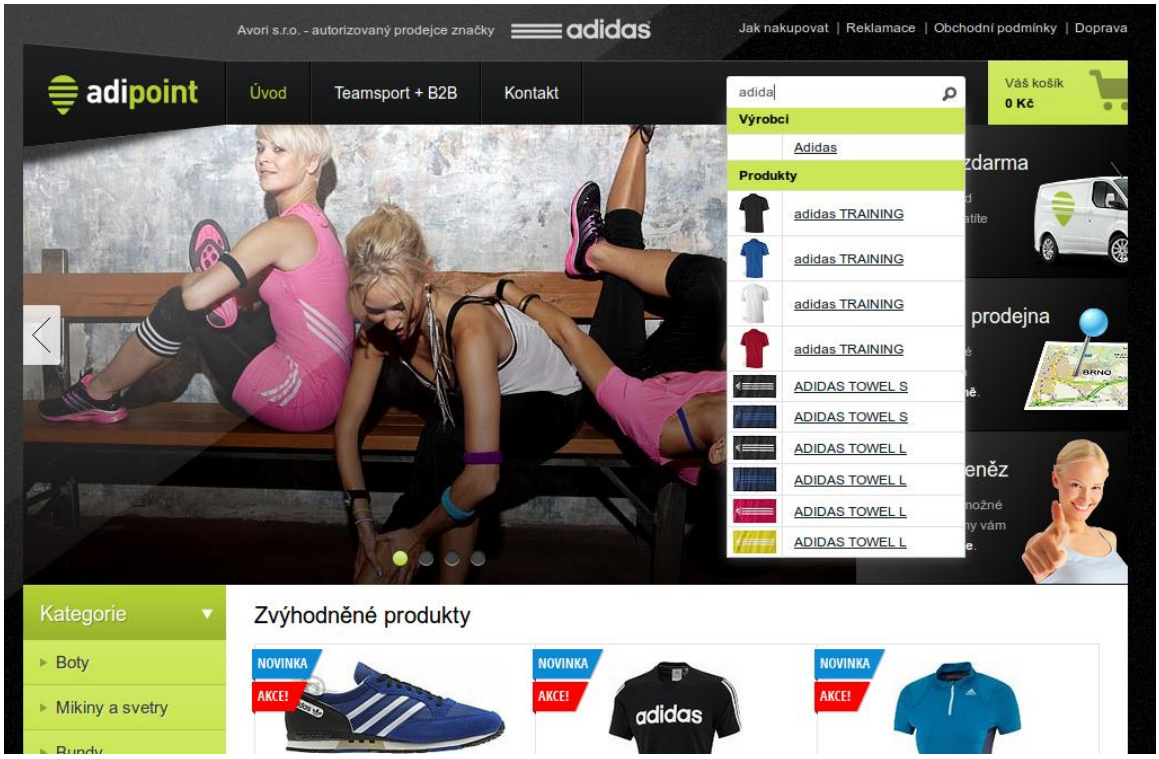
[23] Výchozí Latte makra http://doc.nette.org/cs/default-macros

[24] Výchozí helpery šablon http://doc.nette.org/cs/default-helpers

```
        $search->setModel($this->context->searchModel);

        $search->setIsEshop($this->context->parameters['eshop']);

        return $search;

    }
```

V šabloně lze zavolat pomocí makra `{control search}`, které komponentu vykreslí. Použití v praxi[25] je znázorněno na obrázku číslo 20.

Obrázek 20 -Příklad komponenty SearchInput.



4.7.3 Back-end

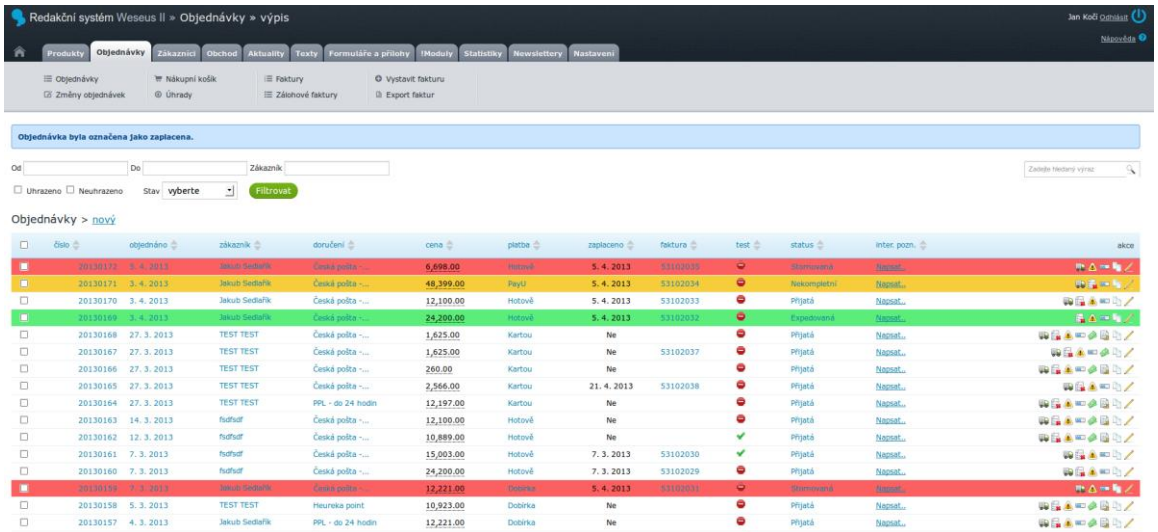
Pro vývoj administrační části byl použit generátor administrace popsaný v kapitole číslo 4.1. Nalezené nedostatky původních knihoven byly odstraněny integrací Nette frameworku. V plné síle bylo využito jeho ladících nástrojů, které usnadnili odhalování a reportování chyb. Dále byl do základní třídy administrace integrován kompilátor DI kontejneru, který umožnil volat z libovolného místa veškeré funkcionality a knihovny vytvořené v části front-endu.

[25] http://www.adipoint.cz/

Ukázka funkce integrující DI kontejner do administrace.

```
/**
 * Vrátí DI container Nette aplikace.
 * @return Nette\DI\Container
 */
public function Nette() {
    static $container;
    if ($container)
        return $container;
    $configurator = new Nette\Config\Configurator;
    $configurator->setTempDirectory(TEMP_DIR);
    $configurator->createRobotLoader()
            ->addDirectory(LIBS_DIR)
            ->addDirectory(APP_DIR)
            ->register();
    $configurator->addConfig(APP_DIR . '/config/config.neon');
    return $container = $configurator->createContainer();
}
```

Obrázek 21- Uživatelské rozhraní administrace, konkrétně seznam objednávek.



Vygenerované soubory byly postupně upravovány tak, aby výsledné uživatelské rozhraní bylo co nejpřívětivější a splňovalo požadavky moderní Web 2.0 aplikace.

Volbou tohoto postupu byla implementace výrazně urychlena a i do budoucna představuje možnost urychlení vývoje nových modulů. Na obrázku číslo 21 je zobrazena ukázka uživatelského rozhranní.

4.8 Testování

Před nasazením systému v produkčním režimu je nutné řádně otestovat veškeré funkcionality a zobrazení na všech podporovaných platformách.

4.8.1 Uživatelské testování

Pro účely testování systému bylo zvoleno uživatelské testování. Během implementace byl systém testován průběžně po každé dílčí fázi vývoje. Samotné testování probíhalo ve dvou fázích:

1. testování na lokální kopii systému,
2. testování na produkčním serveru.

Lokální testování probíhalo souběžně s programováním, kdy byla každá nová funkcionalita pečlivě otestována s více možnými scénáři. Po vyhodnocení správné funkčnosti byl proveden *commit* pomocí verzovacího systému Git. Následné změny byly příkazem *push* přeneseny do centrálního repozitáře, který zároveň představoval funkční testovací instanci. Na produkčním serveru probíhalo testování stejným způsobem jako na lokální kopii. Tímto způsobem byly odstraněny všechny závažné chyby a nedostatky kódu.

4.8.2 Testování použitelnosti

Přestože systém slouží pro nasazení v modelu SaaS, má zpravidla každá instance vlastní grafický layout. Po implementaci grafického designu je před ostrým spuštěním konkrétní instance nutné znovu otestovat všechny funkcionality. Důvodem je možná nekompatibilita nebo narušení skriptů běžících na straně klienta.

Dále musí projít vizuální kontrolou ve všech podporovaných internetových prohlížečích:

- Internet Explorer,
- Firefox,
- Opera,
- Google Chrome,
- Safari.

Pro testování byly využity nejnovější verze prohlížečů. Vyjimkou je Internet Explorer, u kterého musí být díky rozšířenosti jeho starších verzí použito programu IE tester, který usnadní kontrolu zobrazení ve všech novějších verzích od řady 7.

Pro testování zobrazení na mobilních zařízeních bylo nutné použít mobilní telefony a tablety se třemi nejrozšířenějšími platformami:

- Google Android[26],
- Windows Phone 7.8[27] nebo novější Windows 8[28],
- Apple iOS[29].

4.9 Nasazení

Systém je primárně určen pro nasazení v bussines modelu SaaS. V kontextu internetového obchodu se jedná o co nejsnazší poskytnutí vlastní kopie zákazníkovi, který si může ihned vyzkoušet práci s katalogem produktů a ostatní funkce systému. Z tohoto pohledu je systém připraven, protože byl otestován na produkčním serveru (viz. kapitola 4.8).

Pro potřeby nasazení systému jako služby byla vytvořena jednoduchá webová aplikace, která je veřejně dostupná z internetu[30] a umožňuje obchodníkovi vytvoření vlastní zkušební kopie systému. Systém je nasazen na vlastním virtuálním serveru se a pomocí shell skriptu uvedeném v příloze B provádí vytváření nových instalací.

Požadavky na zákazníka

Zájemci o zkušební verzi internetového obchodu stačí pouze vlastnit libovolné zařízení s připojením k internetu a běžný internetový prohlížeč, viz 4.8.2 . Po načtení instalační webové aplikace vyplní validní e-mailovou adresu a potvrdí odesláním formuláře. Poté aplikace na pozadí vytvoří novou kopii systému, včetně testovacích dat a na zadanou adresu zašle přístupové údaje. Zákazník tím získá zdarma plně funkční verzi systému omezenou na zkušební dobu třiceti dnů.

[26] http://www.google.com/mobile/android/

[27] http://www.windowsphone.com/cs-cz

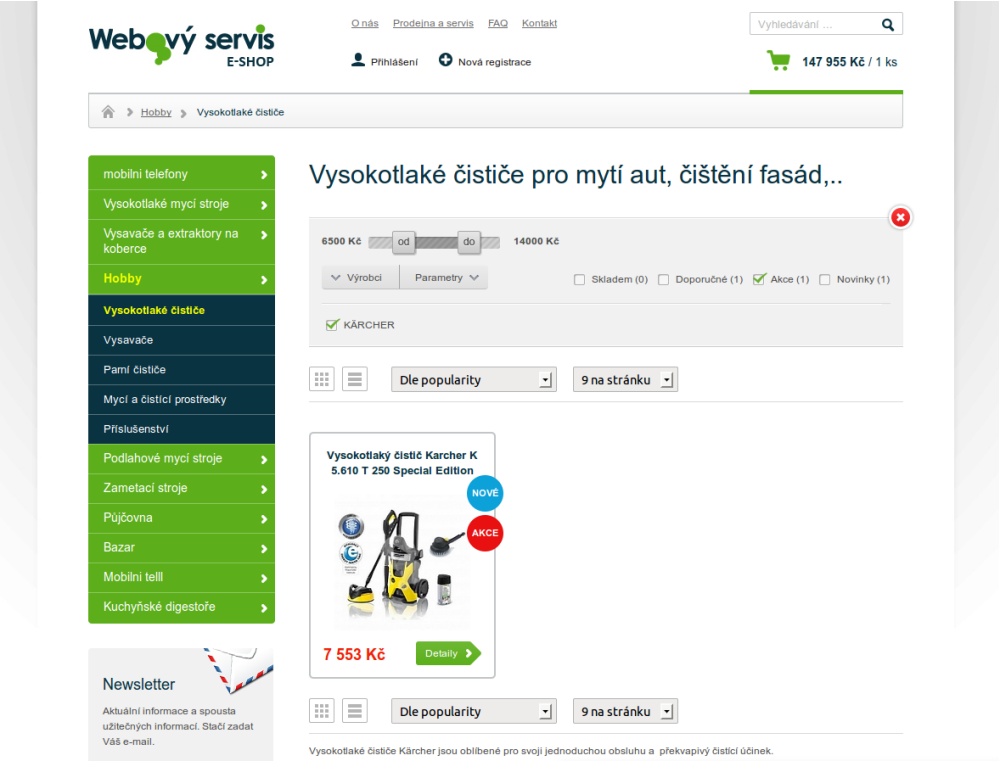
[28] http://windows.microsoft.com/cs-cz/windows-8/meet

[29] http://www.apple.com/cz/ios/

[30] http://demo.webovy-servis.com

Na obrázku číslo 22 je zobrazen snímek vytvořené testovací kopie systému, která obsahuje základní grafickou šablonu.

Obrázek 22 - Ukázka filtrování produktů na výpisu kategorie.

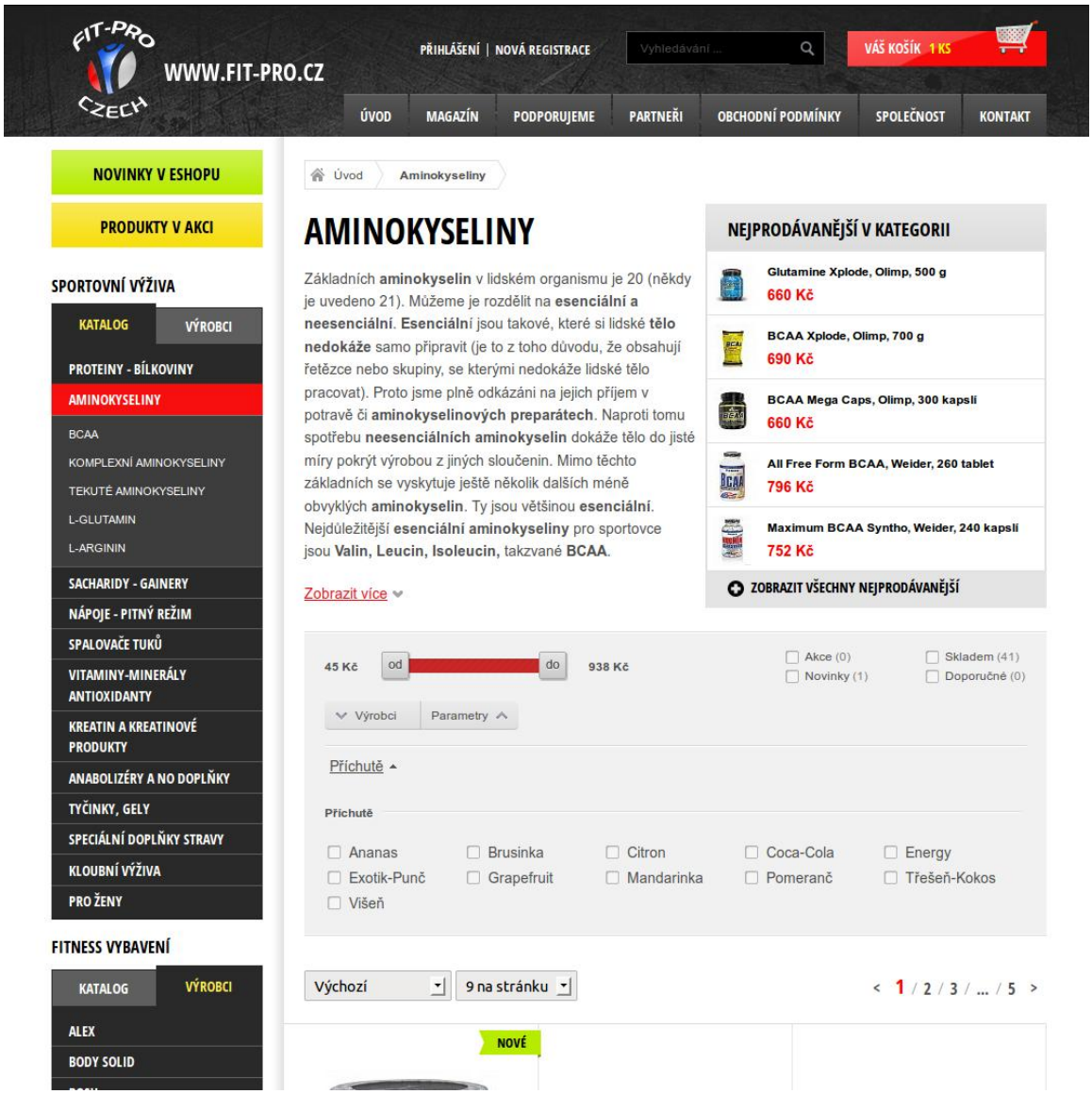


Zákazník si tedy může ve zkušební době systém vyzkoušet a porovnat s konkurencí. Pro rychlé zorientování v administraci slouží příručka obsažená na přiloženém CD. Pokud se zákazník rozhodne pro koupi systému, zbývá pouze vytvořit a implementovat unikátní grafickou šablonu. Dále bývají obvykle realizovány úpravy na míru, jako je napojení na ERP systémy nebo dodavatele. Veškeré customizace lze libovolně implementovat, protože má každá instance vlastní zdrojové kódy uchované ve vlastním Git repozitáři.

Poslední fází je publikování otestované aplikace na vlastní doméně a zahájení marketingových aktivit. Zákazník se tak vůbec nemusí starat o správu hardware, údržbu systému a může se plně věnovat správě přijatých objednávek. Na obrázku 23 je zobrazena ukázka nasazení systému v praxi[31] včetně unikátního grafického designu, parametrického filtrování a výsledku optimalizace textů pro vyhledávače.

[31] FIT-PRO CZECH s.r.o. - http://www.fit-pro.cz

Obrázek 23 - Ukázka nasazení systému v praxi. Zachycuje implementaci unikátní grafické šablony.



Závěr

Cílem této diplomové práce bylo vytvoření konkurenceschopného internetového obchodu, který bude možné nasadit v byznys modelu SaaS. V teoretické části byly nejprve vymezeny požadavky na kvalitní internetový obchod a definovány možnosti jeho pořízení. Dále byl popsán samotný model SaaS a byly uvedeny klady i zápory jeho použití. Pro získání uceleného pohledu na konkurenční systémy byly nejprve dle předem stanovených kritérií vybrány tři nejlepší systémy na trhu. Následně bylo provedeno jejich porovnání se systémem Weseus III a vymezeny standardy. Informace získané v teoretické části byly využity při analýze požadavků na vyvíjený systém.

Samotný vývoj systému probíhal v několika fázích a časově nejnáročnější byla zřejmě analytická část práce. Investovaný čas do analýzy se ale bezesporu vrátil při implementaci, kdy práci urychlila dobrá volba technologií, zvláště Nette frameworku, navržená struktura datových tříd a modelu databáze. Po otestování systému a nasazení na základní grafický design byly vytvořeny skripty sloužící pro nasazení systému jako služby.

Výsledný systém obsahuje všechny základní funkcionality katalogu produktů, umožňuje optimalizaci pro vyhledávače a podporu všech marketingových nástrojů navržených v teoretické části. Systém je optimalizovaný pro zobrazení na mobilních zařízeních a je napojen na platební brány 3D secure a PayU. Podporuje integraci s dopravci Česká pošta, PPL partner a Heuréka Point. V současné verzi je dokončena implementace propojení s ERP systémy Pohoda a Helios Red.

První internetový obchod byl uveden do ostrého provozu v lednu roku 2013 a v současné době společnost Webový servis spouští již patnáctou instalaci. Systém tedy lze považovat za komerčně úspěšný, a co do funkcionalit snese srovnání s nejlepšími řešeními na trhu. Důkazem je hodnocení systému portálem Heuréka.cz, kdy aplikace obdržela osm z devíti bodů reprezentujících integrované služby.

Cíle diplomové práce tedy byly naplněny, ale vývoj systému bude pokračovat dále. Dalším milníkem je integrace účetního softwaru S3 Money a Duel nebo mobilních plateb Mobito a Mastercard Mobile.

Literatura

[1] CHLEBOVSKÝ, Vít. CRM: řízení vztahů se zákazníky. Vyd. 1. Brno: Computer Press, 2005, vi, 190 s. ISBN 80-251-0798-1.

[2] Eshopy B2C i B2B. ALLEGRO GROUP CZ, s.r.o. ShopCentrik - profesionální e-shopy na míru [online]. 2012 [cit. 2013-04-06]. Dostupné z: http://www.shopcentrik.cz/vyhody-reseni-shopcentrik/internetove-obchody-pro-segmenty-b2c-i-b2b.aspx

[3] HERINK, Ondřej. Jak změřit a využít uživatelský prožitek (User Experience) ke zvýšení obratu e-shopu?. GFK CZECH, s.r.o. GfK Czech [online]. 2010 [cit. 2013-04-06]. Dostupné z: http://www.gfk.cz/public_relations/newsletter/newsletter2010/newsletter_articles/006709/index.cz.html

[4] VÁLKA, Ondřej. Co je UX design. Notebook of a freelance interaction designer [online]. 2011, 11.4.2011 [cit. 2013-04-06]. Dostupné z: http://valka.info/notes/page/2/

[5] BUSTOS, Linda. Improving Conversion After the Add-To-Cart. Get Elastic [online]. 2011, 24.1.2011 [cit. 2013-04-06]. Dostupné z: http://www.getelastic.com/improving-conversion-after-the-add-to-cart/

[6] Uživatelské testování použitelnosti. H1.CZ S. R. O. Jednička v internetovém poradenství a výkonnostním marketingu [online]. 2013 [cit. 2013-04-06]. Dostupné z: http://www.h1.cz/pouzitelnost

[7] KUBÍČEK, Michal. Velký průvodce SEO: jak dosáhnout nejlepších pozic ve vyhledávačích. Vyd. 1. Brno: Computer Press, 2008, 318 s. ISBN 978-80-251-2195-5.

[8] SEDLÁK, Mirek a Petra MIKULÁŠKOVÁ. Jak vytvořit úspěšný a výdělečný internetový obchod. 1. vyd. Brno: Computer Press, 2012, 336 s. ISBN 978-80-251-3727-7.

[9] Ohrozí srovnávač Google Nákupy postavení Zboží.cz a Heureka.cz?. MORÁVEK, Daniel. Podnikatel.cz [online]. 2012, 3. 8. 2012 [cit. 2013-04-06]. Dostupné z: http://www.podnikatel.cz/clanky/ohrozi-srovnavac-google-nakupy-postaveni-zbozi-cz-a-heureka-cz/

[10] Jak vybírají Češi e-shopy? Nejčastěji podle srovnávačů cen. Lidovky.cz [online]. 2013, 4.3.2013 [cit. 2013-04-06]. Dostupné z: http://www.lidovky.cz/cesi-pri-online-nakupech-stale-vic-vyuzivaji-srovnavace-cen-pld-/media.aspx?c=A130303_164159_ln-media_sk

[11] BAUCKMANN, David. Pohled do budoucnosti srovnávačů cen. TyInternety.cz [online]. 2012, 17. 09. 2012 [cit. 2013-04-06]. Dostupné z: http://www.tyinternety.cz/2012/09/17/clanek/pohled-do-budoucnosti-srovnavacu-cen/

[12] RŮŽIČKA, Antonín. Proč a jak nasadit e-commerce v Google Analytics. Sun marketing [online]. 2012, 7.2.2012 [cit. 2013-04-06]. Dostupné z: http://www.sunitka.cz/c/419-proc-a-jak-nasadit-e-commerce-v-google-analytics

[13] KAUSHIK, Avinash. Webová analytika 2.0: kompletní průvodce analýzami návštěvnosti. Vyd. 1. Brno: Computer Press, 2011, 456 s. ISBN 978-80-251-2964-7.

[14] Tři čtvrtě uživatelů chytrého telefonu chtějí nový: to je ráj. Netshopper [online]. 2013, 06.03.2013 [cit. 2013-04-06]. Dostupné z: http://www.netshopper.cz/tri-ctvrte-uzivatelu-chytreho-telefonu-chteji-novy-to-je-raj.aspx

[15] Jaká přístupová zařízení používali návštěvníci na komerčních webech v roce 2012?. EFFECTIX.COM S.R.O. Doba webová: GU FIREM [online]. 2013, 17.2.2013 [cit. 2013-04-06]. Dostupné z: http://www.doba-webova.com/cs/mobilni-web

[16] E-shopy jdou do mobilních verzí. První spustila Alza, následovat jí bude Kasa. Podnikatel.cz [online]. 2012, 27.3.2012 [cit. 2013-04-06]. Dostupné z:

http://www.podnikatel.cz/clanky/e-shopy-jdou-do-mobilnich-verzi-prvni-spustila-alza-nasledovat-ji-bude-kasa/

[17] Responzivní web design. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2013-04-06]. Dostupné z: http://cs.wikipedia.org/wiki/Responzivn%C3%AD_web_design

[18] Revoluce se nekoná – dobírka stále vládne českým e-shopům. Netshopper.cz [online]. 2011, 22.04.2011 [cit. 2013-04-06]. Dostupné z: http://www.netshopper.cz/revoluce-se-nekona-dobirka-stale-vladne-ceskym-e-shopum.aspx

[19] MÁČALOVÁ, Pavlína. Podnikáte na internetu? Podívejte se, jaké platební metody jsou pro vás nejefektivnější. ProByznys.info [online]. 2012, 8. 11. 2012 [cit. 2013-04-06]. Dostupné z: http://probyznysinfo.ihned.cz/obchod-a-marketing-obchod/c1-58291050-podnikate-na-internetu-podivejte-se-jake-platebni-metody-jsou-pro-vas-nejefektivnejsi

[20] VETYŠKA, Jan. Platební metody na českém internetu 2012. SOCR ČR [online]. 26.6.2012, 2012 [cit. 2013-04-06]. Dostupné z: http://www.socr.cz/assets/zpravodajstvi/tiskove-zpravy/platebni-metody-na-ceskem-internetu-2012-final.pdf

[21] MACICH, Jiří. APEK: Mezi dopravci vede Česká pošta, zákazníky irituje fingovaný pokus o doručení. ChannelWorld [online]. 2012, 1.6.2012 [cit. 2013-04-07]. Dostupné z: http://channelworld.cz/podpora/apek-mezi-dopravci-vede-ceska-posta-zakazniky-irituje-fingovany-pokus-o-doruceni-6474

[22] VYLEŤAL, Martin. Kouzelná almara spustila své balíkomaty. Velké e-shopy zatím nejeví zájem. Lupa.cz [online]. 2013, 5.4.2013 [cit. 2013-04-07]. Dostupné z: http://www.lupa.cz/clanky/kouzelna-almara-spustila-sve-balikomaty-velke-e-shopy-zatim-nejevi-zajem/

[23] E-shopy: Přehled open source systémů. Open source na českém hostingu [online]. 2012 [cit. 2013-04-08]. Dostupné z: http://opensource.cesky-hosting.cz/e-shopy/

[24] Software & Information Industry Association. Software as a Service: Strategic Backgrounder, 02 / 2001. URL: http://www.siia.net/estore/ssb-01.pdf

[25] Software as a Service. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2013-04-08]. Dostupné z: http://cs.wikipedia.org/wiki/Software_as_a_Service

[26] Návrh a implementace redakčního systému pro komerční využití. Pardubice, 2010. Diplomová práce. Univerzita Pardubice.

[27] LAMP (software bundle). In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2013-04-13]. Dostupné z: http://en.wikipedia.org/wiki/LAMP_%28software_bundle%29

[28] CHACON, Scott. Vše je lokální. Proč je Git lepší než X? [online]. 2013 [cit. 2013-04-13]. Dostupné z: http://whygitisbetterthanx.gitfu.cz/#everything-is-local

[29] GRUDL, David. Nette Framework: MVC & MVP. Zdroják.cz [online]. 2009, 24.3.2009 [cit. 2013-04-14]. Dostupné z: http://www.zdrojak.cz/clanky/nette-framework-mvc--mvp/

[30] MVC aplikace & presentery. Nette framework: Dokumentace [online]. 2011 [cit. 2013-04-21]. Dostupné z: http://doc.nette.org/cs/presenters#toc-model-view-controller-mvc

Seznam příloh

Příloha A – struktura přiloženého CD s popisem obsahu

Příloha B – shell skript pro nasazení v modelu SaaS

Příloha C – uživatelský manuál

Příloha A – struktura přiloženého CD s popisem obsahu

- KociJ_NavrhImplementace_JH_2013.pdf – text diplomové práce ve formátu PDF
- analyza
    - analyza.eap – obsahuje zpracované požadavky, Use-Case diagram a návrhové třídy;
- manual - Weseus_III_manual.docx – uživatelská příručka
- aplikace – veškeré zdrojové kódy aplikace;
- databaze
    - db_create.sql – skript pro vygenerování základní databázové struktury;
    - er_diagram_workbench.mwb – ER diagram databáze ve formátu MySQL Workbench;
- deploy
    - deploy-git.sh – shell skript pro nasazeni v modelu SaaS;

Příloha B - shell skript pro nasazení v modelu SaaS

```
#!/bin/bash
function verbose() {
      echo "Je nutne zadat prave 1 parametr jako nazev repozitare"
}

if [ -z "${1}" ]; then
      verbose
      exit -1
fi

repo="${1}"
PWD="/home/git"

COPY_FROM="/www/hosting/domena.ltd/adresar/"
HOSTING_PATH="/www/hosting/domena.ltd "
GIT_SERVER="git@muj.server.cz"

OLD_DBNAME="sandbox_db"
OLD_USER=" sandbox_user"
OLD_PASS=" sandbox_pass"
OLD_NB_NAME=" sandbox_netbeans_project_name"

NEW_USER="deploy_user"
NEW_PASS="deploy_pass"

DIR_WEBTEMP=$COPY_FROM"webtemp"
DIR_LOG=$COPY_FROM"front/log"
DIR_TEMP=$COPY_FROM"front/temp"
DIR_CACHE=$DIR_TEMP"/cache"
DIR_SESSIONS=$DIR_TEMP"/sessions"
DIR_TEMPLATES=$COPY_FROM"admin/templates_c"
DIR_LANG=$COPY_FROM"front/app/lang"

for i in "$repo"; do
      #git working tree
      mkdir "$PWD/${i}.git"
      cd "$PWD/${i}.git"
      git --bare init
      cp -ra hooks/post-receive.sample hooks/post-receive
      echo "GIT_WORK_TREE=$HOSTING_PATH/${i} git checkout -f" >>
hooks/post-receive
      chmod a+x hooks/post-receive
      cd -
      mkdir -p "$HOSTING_PATH/${i}"

      #promazani temp slozek
      find $DIR_WEBTEMP -mindepth 1 -delete
      find $DIR_LOG -mindepth 1 -delete
      find $DIR_TEMP -mindepth 1 -delete
      find $DIR_TEMPLATES -mindepth 1 -delete
      mkdir -p $DIR_SESSIONS

      #kopie souboru
```

```
      cp -ra "$COPY_FROM." "$HOSTING_PATH/${i}"

      #nasatveni configu
      CONFIG_FILE="$HOSTING_PATH/${i}/front/app/config/config.neon"
      sed -i "s/dbname: $OLD_DBNAME/dbname: ${i}/g" $CONFIG_FILE
      sed -i "s/user: $OLD_USER/user: $NEW_USER/g" $CONFIG_FILE
      sed -i "s/password: wsd123eshop/password: $NEW_PASS/g"
$CONFIG_FILE
      echo -e '\nNasatven config.neon';

      #nasatveni netbeans project
      sed -i "s/<name>$OLD_NB_NAME/<name>${i}/g"
$HOSTING_PATH/${i}/"nbproject/project.xml"
      echo -e '\nNasatven Netbeans projekt';

      #nastaveni vlastnika www-data
      chown -R www-data:www-data "$PWD/${i}.git" "$HOSTING_PATH/${i}"

      cd /tmp

      #git clone a push origin master
      git clone "$GIT_SERVER:${i}.git" "git-${i}"
      cp -ra "$HOSTING_PATH/${i}/." "/tmp/git-${i}"
      cd -
      cd "/tmp/git-${i}"
      touch testfile
      git add --all
      git commit -a -m "prvni commit celeho webu"
      git push origin master

      rm -rf "/tmp/git-${i}"
      cd -
      echo -e "\n\nVytvoren repozitar ${i}.git"
      echo "Tento repozitar bude automaticky opublikovan do adresare
$HOSTING_PATH/${i}"
      echo "Klon na lokalni disk se provede prikazem: git clone
$GIT_SERVER:${i}.git vas-zvoleny-lokalni-adresar"

      #vytvoreni db
      mysql -u$NEW_USER -p$NEW_PASS -h localhost -e "CREATE DATABASE
${i} CHARACTER SET utf8 COLLATE utf8_czech_ci;"
      echo -e "\n\nVytvorena db ${i}"

      #kopie struktury a dat db
      mysqldump -h localhost -u$NEW_USER -p$NEW_PASS $OLD_DBNAME | mysql
-h localhost -u $NEW_USER -p$NEW_PASS ${i}
      echo -e "\n\nVytvorena kopie db ${i}"

done
```