

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

WEBOVÝ SYSTÉM NÁKUPNÍHO PORADENSTVÍ

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

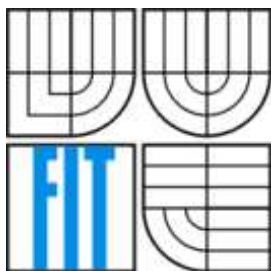
AUTOR PRÁCE
AUTHOR

Miroslav Babják

BRNO 2012



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

WEBOVÝ SYSTÉM NÁKUPNÍHO PORADENSTVÍ

WEB CUSTOMER GUIDANCE SYSTEM

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

Miroslav Babják

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. Radek Burget, Ph.D.

BRNO 2012

Abstrakt

Předmětem této práce je vytvořit internetový obchod komunikující s ERP systémem Money S5. Součástí tohoto obchodu je i systém nákupního poradenství, který má zákazníkům maximálně usnadnit výběr vhodných produktů na základě zvolených parametrů. Systém nákupního poradenství je postaven na PHP frameworku Nette s využitím architektury Model-View-Presenter. Pro uložení dat jsou využity technologie MySQL a Microsoft SQL Server 2008. Pro implementaci uživatelského rozhraní je využit značkovací jazyk XHTML, CSS, JavaScript a JQuery.

Abstract

Subject of this work is create internet shop communicating with ERP system Money S5. Part of this shop is customer guiding system, which has facilitate customer selection of suitable products based on selected parameters. Guiding system is build on PHP framework Nette using Model-View-Presenter architecture. For data storage is using MySQL and Microsoft SQL Server technology. Implementation of user interface using XHTML markup language, CSS, JavaScript and JQuery.

Klíčová slova

PHP, Nette, framework, ERP systém, Money S5, MySQL, MS-SQL, e-shop, XHTML, CSS, JQuery, nákupní poradenství, webová aplikace

Keywords

PHP, Nette, framework, ERP system, Money S5, MySQL, MS-SQL, e.shop, XHTML, CSS, JQuery, shopping consultancy, web application

Citace

Babják Miroslav: Webový systém nákupního poradenství, bakalářská práce, Brno, FIT VUT v Brně, 2012

Webový systém nákupního poradenství

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Radka Burgeta Ph.D.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Miroslav Babják

13.5.2012

Poděkování

Rád bych poděkoval svému vedoucímu Ing. Radku Burgetovi Ph.D. za pomoc a odborné vedení při vypracování této bakalářské práce. Dále bych chtěl poděkovat lidem z firem Pseudom s.r.o. a Hamendeggs s.r.o., kteří mi umožnili vypracování této práce a také za jejich podporu a pomoc.

© Miroslav Babják, 2012

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

Obsah.....	1
1 Úvod.....	3
2 Použité technologie.....	4
2.1 Frameworky.....	4
2.1.1 Model – View – Controller (MVC).....	5
2.2 Přehled PHP frameworků.....	6
2.3 Skriptovací jazyk PHP.....	7
2.4 Relační databáze MS-SQL.....	7
2.5 Relační databáze MySQL.....	8
2.6 XHTML.....	8
2.7 Cascading Style Sheet (CSS).....	8
2.8 JavaScript a JQuery framework.....	8
3 ERP systémy a Money S5.....	10
3.1 Enterprise Resource Planning (ERP).....	10
3.2 Money S5.....	11
3.3 Propojení s webovými aplikacemi.....	11
3.3.1 Money S5 eshop.....	12
3.3.2 Zrcadlení databázových tabulek.....	12
3.3.3 Přímý přístup do databáze.....	12
4 Návrh architektury systému.....	13
4.1 Diagram případů užití.....	14
4.2 Entity – Relationship diagram.....	16
4.2.1 E-R diagram systému.....	16
4.2.2 E-R diagram grafického konfiguratoru.....	17
4.3 Návrh databáze.....	18
4.3.1 Pohled sdružující informace o produktech.....	19
4.3.2 Pohled sdružující parametry produktů.....	19
5 Implementace systému.....	21
5.1 Vytvoření databáze grafického konfiguratoru.....	21
5.2 Propojení s ERP systémem Money S5.....	22
5.2.1 SQL dotaz pro výběr produktů.....	22
5.2.2 Registrace uživatelů.....	23
5.2.3 Vkládání objednávek.....	24
5.3 Výběr automobilu.....	26

5.4	Zobrazení automobilu	27
5.5	Změna barvy automobilu	27
5.6	Implementace horizontálního posuvníku	28
5.7	Změna velikosti disku	29
6	Testování v reálném nasazení	31
7	Závěr	32

1 Úvod

Tato práce popisuje tvorbu aplikace pro prodej litých disků a pneumatik na automobily. Zadavatelem této práce je firma Pseudom s.r.o. Tato firma se pohybuje v tomto odvětví již 5 let a provozuje internetový obchod jak na Slovensku, tak v České republice. Nyní chtělo svůj repertoár rozšířit o další internetový obchod, který však navíc bude obsahovat systém nákupního poradenství. Tímto se tedy zrodil projekt disky.sk, který by se měl stát evropským portálem pro prodej hliníkových disků a pneumatik. Jeho hlavní předností by měl být systém nákupního poradenství, který návštěvníkům pomůže s výběrem disků na jejich automobily a nabídne jim špičkové uživatelské rozhraní. Navíc pro co nejjednodušší správu objednávek a financí, chtěl zákazník tento internetový obchod integrovat na svůj firemní systém Money S5. Toto byl klíčový požadavek, který zásadním způsobem ovlivnil vývoj celé aplikace.

Aplikace sází na využívání systému nákupního poradenství, jelikož jen málo portálů tento systém nabízí. Tento systém by měl uživateli navrhnout vhodné disky na jeho automobil a usnadnit mu výběr mezi nimi. Celý systém je založen na co nejrealističtějším zobrazení automobilu a disků. Kdy návštěvník si pouze vybere konkrétní automobil a hned se mu zobrazí všechny disky na něj pasující, aniž by musel znát technické údaje vozidla či disku. Navíc si může vybrat barvu automobilu ze široké škály barev, aby si automobil co nejvíce přizpůsobil svému. Každý disk v nabídce, může být zobrazen přímo na automobilu, což návštěvníkovi velmi usnadní výběr vhodného modelu disku pro jeho automobil. Tento systém se samozřejmě již na některých portálech vyskytuje, avšak náš systém sází na špičkové zobrazení automobilů a tedy co nejrealističtější pohled. Navíc na Slovensku nikdo takový systém nevlastní a portály si jej většinou pronajímají od zahraničních partnerů za obrovské peníze. Je tedy možné uvažovat o možnosti prodeje tohoto systému nákupního poradenství těmto subjektům.

V tomto dokumentu se nejdříve seznámíme s technologiemi, pomocí kterých je tato aplikace vyvíjena. Dále se podíváme na ERP systémem Money S5 a jeho možnosti integrace webových řešení. Základním kamenem každé aplikace je návrh, na který se zaměříme v další kapitole. Poslední kapitola se zabývá implementací a popisuje, jakým způsobem byly programovány důležité součásti aplikace. Na konci dokumentu najdeme závěrečné shrnutí a návrhy pro vylepšení systému.

2 Použité technologie

Zadavatel projektu nás v použitých technologiích nijak neomezil. Výběr nejvíce ovlivnila skutečnost, že aplikace musela komunikovat s ERP systémem Money S51. Systém Money S5 je postaven na technologiích firmy Microsoft, tím pádem vznikla otázka, jestli použít i pro vývoj aplikace technologii této firmy pro zajištění snazší komunikace a kompatibility. Po zvážení všech pro a proti jsme nakonec zvolili volně dostupné technologie. Konkrétně skriptovací jazyk PHP, který však v základní instalaci neobsahuje modul pro komunikaci s MS-SQL databází, který pro naše účely je potřeba. Vznikl tím tak problém, kde hostovat výslednou aplikaci. Jelikož jsme měli dost specifické požadavky, tak byl výběr hostingové společnosti dost komplikovaný. Naštěstí jsme měli k dispozici virtuální server, kde jsme si mohli instalaci PHP přizpůsobit podle našich požadavků.

Jelikož má být aplikace dostupná z internetu, tak technologie na straně klienta jsou více méně jednoznačně dané. Pro zobrazení uživatelského rozhraní je využita kombinace XHTML a CSS doplněna skriptovacím jazykem JavaScript.

Na straně serveru i klienta jsou využity dostupné frameworky, pro usnadnění a zpřehlednění programování aplikace. Na straně serveru je použit framework Nette s nástavbou Dibi pro snazší komunikaci s databází. Na straně klienta je použit nejrozšířenější javascriptový framework jQuery.

Využívání frameworků výrazně usnadňuje programování základních paradigmat a umožňuje tak programátorovi zaměřit se na funkčnost vlastní aplikace.

2.1 Frameworky

Jazyk PHP je ve světě tvorby webových aplikací jeden z nejrozšířenějších jazyků, konkurenci mu tvoří pouze jazyky Ruby a Python. Jazyk je lehce naučitelný a jeho instalace na webový server je velmi jednoduchá. S příchodem PHP verze 5 a jejího přepracovaného objektového přístupu a množstvím nových funkcí a knihoven oproti verzi 4 se začali vytvářet první frameworky, které mají za úkol programátorovi maximálně usnadnit vytváření základních funkcí webové aplikace.

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

Framework si můžeme představit jako jakousi abstraktní vrstvu nad jazykem samotným. Nejedná se o externí knihovny jazyka PHP, nýbrž se jedná o aplikaci psanou v jazyce PHP, která umožňuje používat frameworkem vytvořené proměnné, třídy a metody. Tyto třídy a metody slouží pro usnadnění vytváření běžných součástí většiny PHP aplikací. Jako příklad můžeme uvést spojení a komunikace s databází, vytváření a validace formulářů, využití AJAXu apod.

Z počátku je psaní aplikace pomocí frameworku zdlouhavé a pracné. Je totiž potřeba se s frameworkem seznámit, zjistit jaké možnosti nám nabízí, zapamatovat si použité třídy a metody apod. Avšak po překonání počátečních útrap nám bude odměnou značné urychlení vývoje webových aplikací, jejich bezpečnost a snadné rozšíření o další funkce. [4]

¹ Pojem ERP systém bude vysvětlen v kapitole 3

Výhody využití PHP frameworku:

1. Rychlejší vývoj aplikace
2. Zkrácení zdrojového kódu
3. Snadné doplnění dalších funkcí
4. Umožňuje vývoj MVC aplikací
5. Snadná tvorba Friendly URI

2.1.1 Model – View – Controller (MVC)

Jedná se o softwarová architekturu, která vznikla z potřeby oddělit od sebe získávání dat z databáze a jejich následné zpracování a zobrazení. Tyto jednotlivé činnosti jsou rozděleny do jednotlivých „modulů“, které se starají o jejich vykonání a výsledné data si posílají navzájem mezi sebou. Těmi to kroky se celá aplikace velmi zpřehledňuje a usnadňuje budoucí vývoj celé aplikace.

Model

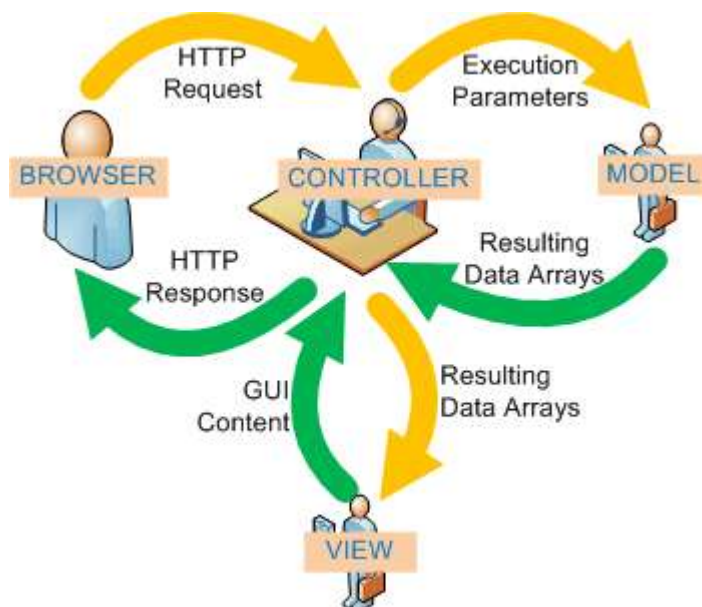
Model tvoří funkční základ celé aplikace a obsahuje její logiku. Jakékoli akce s daty (získání dat, aktualizace dat, mazání apod.) zajišťuje právě model. Pro komunikaci s modelem je ustanoveno pevně dané rozhraní, přes které Controller posílá požadavky na data. Model neví o existenci View nebo Controller.

View

View je vrstva starající se o zobrazení uživatelského rozhraní aplikace. Obvykle používá šablonovací systém a ví, které komponenty se mají zobrazit.

Controller

Controller je řadič, starající se o zpracování požadavků uživatele na jejichž základě volá příslušnou aplikační logiku aplikace, tedy model. Zpracuje data získána z modelu a poté deleguje zobrazení těchto dat na vstupu View.



Obrázek 2.1: MVC model aplikace [7]

2.2 Přehled PHP frameworků

Jak již bylo zmíněno, existuje spousta dostupných PHP frameworků. Zde zmíníme pouze ty nejznámější a nejpoužívanější, probereme jejich základní vlastnosti a přínos pro programátora.

Zend framework

Tento framework se začal vyvíjet od roku 2005. V současné době je k dispozici verze 1.11. Zend byl od začátku stavěn pomocí modulárních balíčků. Použité třídy mají jasně strukturované pojmenování a dané balíčky mohou být použity samostatně. Třídy jsou vytvářeny striktně podle návrhových vzorů a je kladen velký důraz na to, aby si programátor mohl třídy sám rozšiřovat. Daň za tyto možnosti a pravidla je velikost zdrojového kódu frameworku. Je známo, že Zend nevyniká svou rychlostí, pokud tedy není nainstalován jeden z balíčků řešící tento problém. Zend framework je zaměřen více na formu než na obsah, tím pádem se některé užitečné knihovny, které jsou již v jiných frameworkcích nějakou dobu, přidali až v posledních verzích. Jedná se však o jeden z největších frameworků, který má do budoucna velké možnosti a obrovskou uživatelskou základnu. [\[5\]](#)

Symphony

Tento framework se snaží o co největší úsporu kódu psaného programátorem. Poměrně velká část zdrojového kódu byla rozdělena do konfiguračních souborů, které jsou psány ve velmi úsporném a čitelném syntaktickém zápisu zvaném YAML. Jedná se o deklarativní formu programování, která není moc obvyklá. Z toho důvodu je osvojení tohoto frameworku nejsložitější, avšak ve výsledku velmi produktivní. Symphony umí také automaticky generovat administraci vaší aplikace a obsahuje velké množství pluginů pro usnadnění práce programátora. [\[5\]](#)

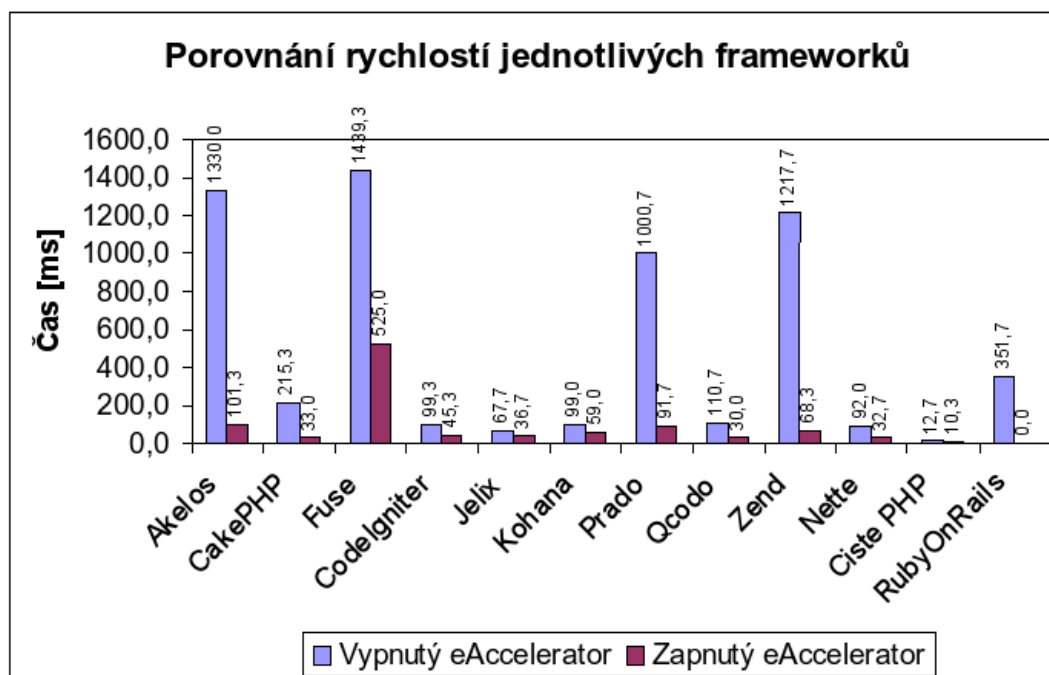
Cake PHP

Jedná se o velmi jednoduchý a rozšířený framework, který se inspiroval v Ruby on Rails. V současné době je k dispozici verze 2.1.1. Výhodou tohoto frameworku je krátký a přímočarý kód, jeho databázová vrstva (active record) a promyšlené view helpery. Jedná se o velmi vhodný nástroj pro psaní CMS systémů a menších e-shopů. Umožňuje také integraci se Zend frameworkem kde kombinuje svou jednoduchost a přímočarost se silou Zendu. Na internetu lze také nalézt spoustu českých psaných návodů. [\[5\]](#)

Nette

Jedná se o framework vyvíjený českým autorem Davidem Grudlem, který vznikl v roce 2008. Jedná se o MVC framework psaný v PHP 5 s plným využitím OOP. Tento framework je zaměřen z velké části na použití komponent, čímž umožňuje už jednou napsaný kód využít na více místech. Dále Nette umožňuje psaní odkazů nikoli pouze klasickou URI adresou, ale také adresou na konkrétní funkci, která se po kliku na prvek vykoná. Nabízí tedy programování řízené událostmi. Další výhodou je vytváření a směrování URL adres, které dává programátorovi značnou volnost v tom, jak budou dané URL vypadat. Další ze skvělých možností Nette je tzv. laděnka, která umožňuje velmi jednoduše a efektivně debugovat napsané aplikace.

Ikdyž se jedná o dost mladý framework, má velkou uživatelskou základnu. Hlavní výhodou je, že veškeré materiály, dokumentace a návody, jsou psané v českém jazyce. [\[5\]](#) [\[6\]](#)



Obrázek 2.2: Porovnání rychlosti jednotlivých frameworků [8]

2.3 Skriptovací jazyk PHP

PHP (Hypertext Preprocessor) je platformou nezávislý skriptovací jazyk používaný hlavně pro tvorbu dynamických webových aplikací. Procesor jazyka PHP běží na HTTP serveru, kde je integrován buď pomocí přídatného modulu, nebo pomocí standardního rozhraní CGI. Při požadavku klienta na PHP soubor, je tento soubor nejprve zpracován PHP procesorem, který vykoná vložený kód a vytvoří HTML dokument, který je následně poslán klientovi jako odpověď. Klient tedy dostává jako odpověď pouze stránku s HTML výsledným kódem. Syntaxe jazyka PHP vychází z několika programovacích jazyků, jako příklad můžeme uvést jazyk C a Perl. [1] Pro jazyk PHP existuje spousta rozšiřujících knihoven pro práci s různými druhy databází, zpracování obrázků, XML souborů apod.

V současnosti je jazyk PHP nejvíce využíván v kombinaci s databázovým serverem MySQL a webovým serverem Apache.

Aplikace je psána v PHP verzi 5.2, i když je v současné době k dispozici PHP verze 5.4.0. Verze 5.2 je použita z důvodu instalace modulu pro komunikaci s MS-SQL databází, který není kompatibilní s vyšší verzí PHP.

2.4 Relační databáze MS-SQL

Microsoft SQL server je relační databázový a analytický systém určený zejména pro podnikovou sféru. Využívá se hlavně pro e-obchody, datové sklady a podnikové řešení firmy Microsoft. SQL server nativně obsahuje podporu pro jazyk XML. Server byl vyvinut pro zpracování velkého objemu transakcí, skladování dat a běh aplikací usnadňující rozhodování. Pro dotazování používá jazyk Transact-SQL, což je verze jazyka SQL upravené firmou Microsoft. SQL server také podporuje replikaci, která umožňuje vytvořit kopie databáze, navíc SQL server hlídá, aby data byla stále synchronizována.

Jedná se tedy hlavně o podnikové řešení a není primárně určeno pro použití ve webových aplikacích.

2.5 Relační databáze MySQL

MySQL patří k nejrozšířenějším relačním databázovým serverům založených na dotazovacím jazyku SQL. To hlavně proto, že je distribuován pod opensource licenci a jeho výkon je v porovnání s konkurenčními databázovými servery velmi dobrý. V minulosti však MySQL nenabízel takové funkce a možnosti jako konkurenční systémy, avšak to se změnilo s příchodem verze 5, která přinesla podporu uložených procedur a funkcí a vytváření pohledů a triggerů.[\[3\]](#)

V aplikaci je použita MySQL databáze ve verzi 5.0, která plně dostačuje pro potřeby aplikace.

2.6 XHTML

XHTML(Extensible HyperText Markup language) je značkovací jazyk pro tvorbu webových prezentací vyvinut konsorciem W3C. Tento jazyk vznikl přeformulováním staršího jazyka HTML, tak aby odpovídal standardu XML. XHTML oproti HTML zpřísnil syntaxi, vypustil některé značky a značně se změnila hlavička dokumentu, kterou XML vyžaduje. V současné době je XHTML dostupné ve verzi 1.1, avšak stále se používá jeho starší verze 1.0, která se rozlišuje na Strict a Transitional. [\[9\]](#)

Aplikace je napsána ve značkovacím jazyce XHTML 1.0 ve variantě Transitional, jazyk XHTML obecně odstranil značky pro vizuální úpravu dokumentu a je tedy nutné, pro formátování dokumentu, použít jazyk CSS.

2.7 Cascading Style Sheet (CSS)

Jedná se o jazyk, které v současné době představuje základní prostředek pro definici vzhledu webových stránek. Jazyk CSS využívají značkovací jazyky, které nemají prostředky pro definici vzhledu webové stránky, tedy jazyky HTML, XHTML a XML. Jedná se o deklarativní jazyk, jehož dokument je tvořen takzvanými pravidly. Pro výběr množiny HTML prvků, kterým se mají přiřadit specifikované vlastnosti, se využívá selektorů. V současnosti nejnovější verze tohoto jazyka je CSS 3, avšak zatím nemá plnou podporu všech internetových prohlížečů, proto se stále z velké části využívá CSS 2.

2.8 JavaScript a JQuery framework

JavaScript je objektově orientovaný skriptovací jazyk, který je součástí HTML kódu webové stránky a je interpretován na straně klienta. Umožňuje vytvořit interaktivní stránky, aniž by je musel zpracovávat webový server. Spolupracuje se všemi internetovými prohlížeči, když každý prohlížeč tento jazyk interpretuje trochu jiným způsobem. Je tedy nutné být při programování opatrní a kontrolovat funkčnost kódu ve všech prohlížečích. Z počátku se JavaScript využíval hlavně na validaci formulářů před jejich odesláním na webový server. Dnes je funkce JavaScriptu poněkud jiná, spolupracuje se všemi aspekty internetových prohlížečů a webových stránek.

Nad jazykem JavaScript bylo vytvořeno množství frameworků, pravděpodobně nejznámější z nich je framework JQuery. JQuery umožňuje jednoduchý výběr elementů DOM² na základě selektorů, které známe z jazyka CSS. Zároveň umožňuje pracovat s událostmi, nabízí pokročilé funkce pro práci s poli a také základní funkce pro animaci prvků.

V současné době je JQuery k dispozici ve verzi 1.7.2. V této aplikaci je použit framework JQuery v jeho nejnovější verzi.

² Objektový Model Dokument - <http://www.tvorba-webu.cz/dom/>

3 ERP systémy a Money S5

Tato kapitola se zabývá ERP systémy, konkrétně systémem Money S5 a jeho možnostmi pro propojení s webovými aplikacemi. Jelikož zadavatel už nějakou dobu používá pro řízení administrace firmy ERP systém Money S5 požadoval, aby aplikace s tímto řešením spolupracovala. Chtěl tedy zajistit to, aby měl správu obchodu v jedné aplikaci zároveň se systémem vytváření faktur, účetnictvím apod. Tato skutečnost vyžadovala provést analýzu systému Money S5 a to konkrétně jeho možnosti pro propojení s webovými aplikacemi. Je několik možných řešení tohoto propojení, konkrétní možnosti si popíšeme níže.

3.1 Enterprise Resource Planning (ERP)

Jedná se o komplexní informační systémy organizací zastřešující činnosti související s výrobou, financemi, účetnictvím, řízením lidských zdrojů apod. Oproti minulosti, kdy firmy řešili každou výše zmíněnou činnost samostatným systémem, dnes firmy spíše volí možnost jednoho velkého systému zastřešujícího všechny tyto odvětví, volí tedy ERP systém, který integruje veškerá data a procesy organizace do jednoho celku. Při nasazení ERP systému do organizace neznamena pouze instalace příslušného softwaru, ale firmy zabývající se implementací těchto systémů provedou kompletní analýzu procesů a postupů organizace. Nasazení ERP systému tedy většinou znamená změnu zaběhlých postupů, která není vždy lehká, avšak přínos ERP systému je obrovský. [\[10\]](#)

Přínosy ERP systémů:

1. zefektivnění a zrychlení ekonomických (podnikových) procesů
2. centralizaci a vyčištění dat, snížení chybovosti
3. optimalizace pracovního toku dokumentů (workflow)
4. dlouhodobé úspory v investicích do informačních systémů a hardware
5. zvýšení bezpečnosti
6. rychlejší výstupy (efektivnější reporting) pro vedení firmy (zaměstnanci nemusí připravovat podklady)
7. podpora pro vedení účetnictví podle mezinárodních standardů



Obrázek 3.1: Model ERP systému [12]

3.2 Money S5

Money S5 je ERP systém vyvinutý českou společností Cigler Software. Tento systém je tvořen moduly, které plní různé funkce a dohromady vytváří celiství program pokrývající všechny aspekty firemních požadavků. Celý systém je postaven na technologiích od firmy Microsoft. Struktura systému je hvězdicová a jejím středem stojí jeden databázový server, na kterém jsou uložena veškerá data. Tyto informace jsou dostupné přes lokální síť i přes internet. Uživatel se tedy může do systému přihlásit odkudkoli. Umožňuje také jednoduše vytvářet uživatelské role s různými přístupovými právy. Jako databázový server byl zvolen MS SQL Server od firmy Microsoft, pro jeho rychlost a bezpečnost. Tento systém pokrývá širokou škálu firemní agendy od jednoduché správy financí a účetnictví až po řízení vztahů se zákazníky a manažerské vyhodnocování. Systém má velmi přívětivé a intuitivní uživatelské rozhraní založené na uživatelském rozhraní MS Office. Systém je vytvořen v 5 jazykových mutacích a obsahuje českou a slovenskou legislativu. [12]

3.3 Propojení s webovými aplikacemi

V dnešní době obrovského rozvoje internetu, kde stále narůstá počet lidí nakupujících v tomto prostředí, bylo nutné zapracovat nějaké řešení jak umožnit správu internetového obchodu a režii s tím spojenou pomocí ERP systému. ERP systém může starat o skladové zásoby, zpracovávání objednávek, automatický nákup zboží apod. Vznikla tedy potřeba dostat data z ERP systému do webového prostředí. Níže si popíšeme 3 možnosti tohoto propojení se systémem Money S5. Každé z těchto 3 řešení má svoje výhody a nevýhody, které je potřeba zvážit. Naše aplikace využívá spojení dvou těchto metod, které bude blíže popsáno v kapitole 5.2.

3.3.1 Money S5 eshop

Toto řešení je výsledkem spolupráce firem Cígler Software a MK Solutions. Jedná se o plně integrovaný internetový obchod do informačního systému Money S5. Umožňuje kompletní správu a vedení internetového obchodu. Tento modul si vytváří vlastní databázi (na stejném databázovém serveru kde běží Money S5), kterou automaticky synchronizuje s databází Money S5. Dále obsahuje redakční systém pro správu obchodu na webu, který je jednoduchý a intuitivní.

Jedná se komplexní řešení, které obsahuje všechny funkce pro provoz internetového obchodu s možností přizpůsobení se na míru zákazníkovi. Nevýhodou tohoto řešení je určitě cena, která se pohybuje od 100 000 Kč. Dále celková neprůhlednost celého obchodu a nemožnost jakýchkoli úprav bez zásahu firmy, která toto řešení poskytl. [13]

3.3.2 Zrcadlení databázových tabulek

Toto řešení není zdaleka tak sofistikované a není jakkoli integrovatelné do systému Money S5 a také neumožňuje správu obchodu z tohoto systému. Pro implementaci tohoto řešení je potřeba dobře znát strukturu databáze Money S5 a dále znalost konfigurace MS SQL Serveru. Pokud chceme data zrcadlit na databázový server na jiné platformě, např. MySQL, tak je nutné nainstalovat ODBC³, které umožní spojení i s jinými databázovými platformami. Princip tohoto řešení spočívá v tom zrcadlit tabulky potřebné pro chod internetového obchodu na vlastní databázový server, ze kterého pak bude obchod data číst. Toto zrcadlení lze uskutečnit za pomoci linkovaného serveru⁴ na databázový server Money S5 a triggerů pro synchronizaci databází. Tedy při jakékoli změně v tabulkách, které chceme zrcadlit, se pomocí triggerů tato operace provede i na druhém serveru.

Jedná se o náročné řešení z hlediska znalosti struktury databáze Money S5 a znalost databázových serverů. Také nemá takové možnosti správy produktů v internetovém obchodě, ale zase se jedná o řešení za zlomek ceny komerčního řešení.

3.3.3 Přímí přístup do databáze

Toto řešení je velmi podobné jako zrcadlení databázových tabulek, jen s tím rozdílem, že místo zrcadlení tabulek přistupujeme k tabulkám přímo do databáze Money S5. Vzhledem k tomu, že databázový server musí být dostupný z internetu, tak je tedy možné k němu přistoupit přímo. Samozřejmě je potřeba znát strukturu databáze a postupovat velmi opatrně, při úpravě dat touto cestou. Nicméně můžeme potom s internetovým obchodem pracovat tak, jako by na žádném ERP systému neběžel a dotazovat jako do každé jiné databáze. Toto řešení nám zajišťuje konzistenci dat, tady vždy budeme pracovat s aktuálními daty.

Toto řešení má opět značnou finanční výhodu oproti komerčnímu řešení, avšak přináší s sebou problémy s optimalizací SQL dotazů, jelikož databázi nemůžeme žádným zásadním způsobem upravovat.

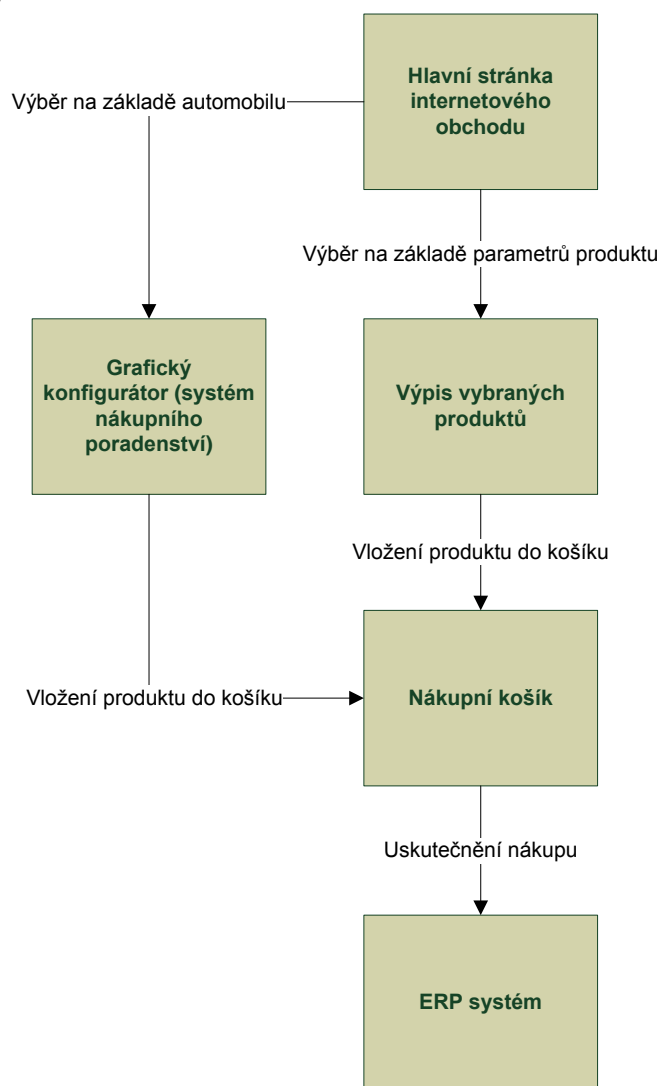
³ Open Database Connectivity – api pro přístup k databázovým systémům

⁴ Více informací o linkovaném serveru naleznete zde <http://msdn.microsoft.com/en-us/library/ms188279.aspx>

4 Návrh architektury systému

Návrh architektury systému vychází hlavně z požadavků klienta, avšak aplikace se taktéž musela přizpůsobit z důvodu integrace na ERP systém Money S5. Tato skutečnost totiž znamenala nemožnost úprav v databázové struktuře a to programování aplikace zkomplikovalo.

Mezi požadavky klienta bylo, aby aplikace získávala aktuální data z databáze Money S5, konkrétně se jednalo o seznam prodáváných produktu, jejich cenu a dostupnost. Aplikace také měla umožňovat jednoduchou změnu jazykové mutace a ukládat objednávky do Money S5, kde se zpracovávaly. Pro přiblížení funkcionality a struktury celého systému bylo vytvořeno blokové schéma, na kterém můžeme názorně vidět, jak je aplikace provázána se systémem nákupního poradenství a s ERP systémem.



Obrázek 4.1: Blokové schéma výsledného systému

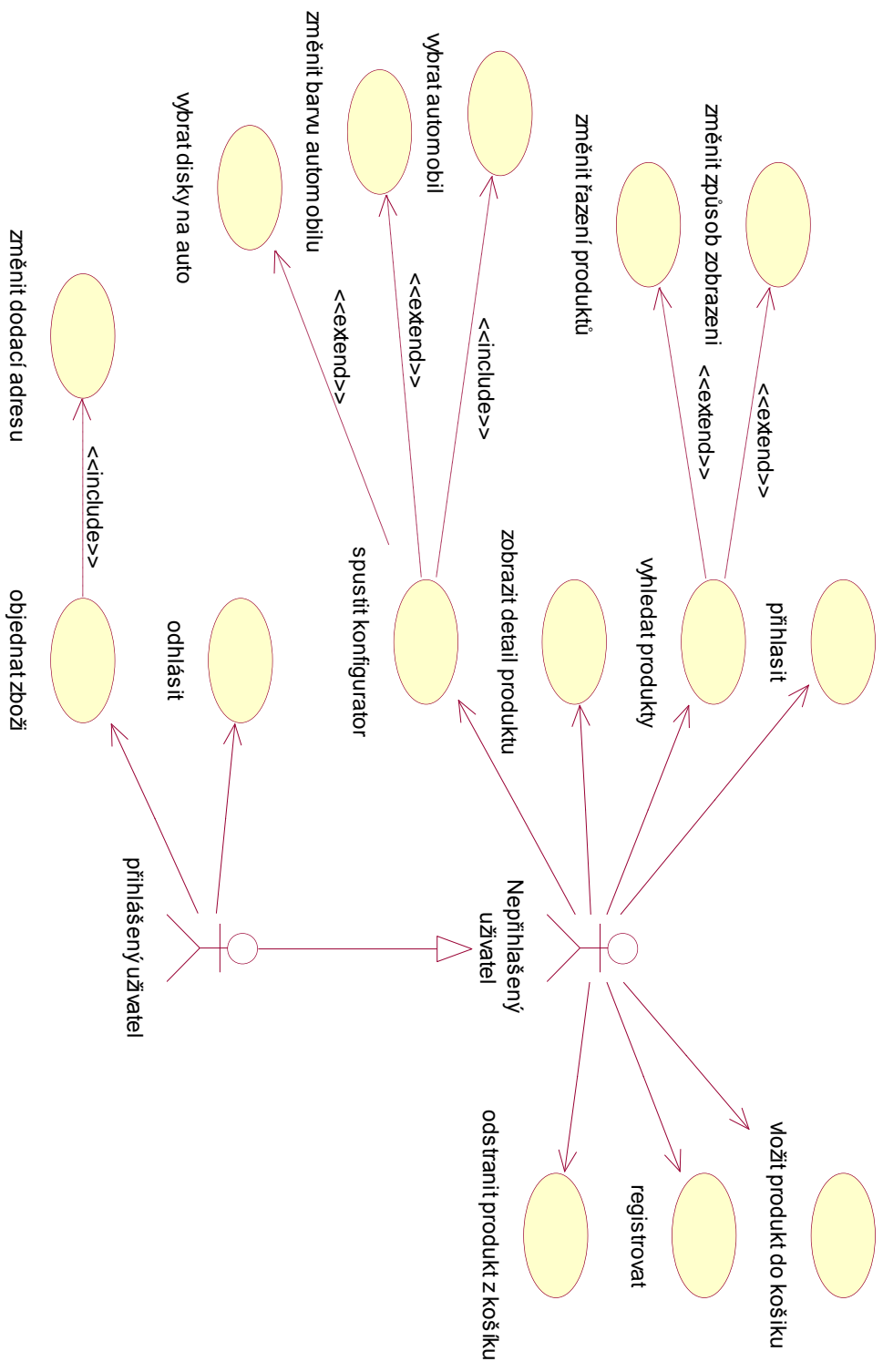
4.1 Diagram případů užití

Jedná se o jednu ze součástí modelovacího jazyka UML. Tento model nám specifikuje, kteří účastníci se v systému budou vyskytovat a jaké akce budou provádět. Specifikuje tedy akce, které lze v systému vykonávat a kteří uživatelé je mohou vykonávat. Také nám určují hranice systému a interakce mezi účastníky.

Tento model byl vytvářen na základě konzultací se zadavatelem a se společností Cígler Software a zobrazuje všechny akce, které lze v systému provádět.

Aktéři tohoto systému jsou pouze dva a to přihlášený a nepřihlášený uživatel.

1. **Nepřihlášený uživatel** – Jedná se o návštěvníka systému, který nemá své přihlašovací jméno a heslo. Může si prohlížet obsah systému, ale nemůže provést objednávku, ta je umožněna pouze přihlášeným uživatelům
2. **Přihlášený uživatel** – Tento aktér má stejné možnosti jako nepřihlášený uživatel, navíc však může provést objednávku.



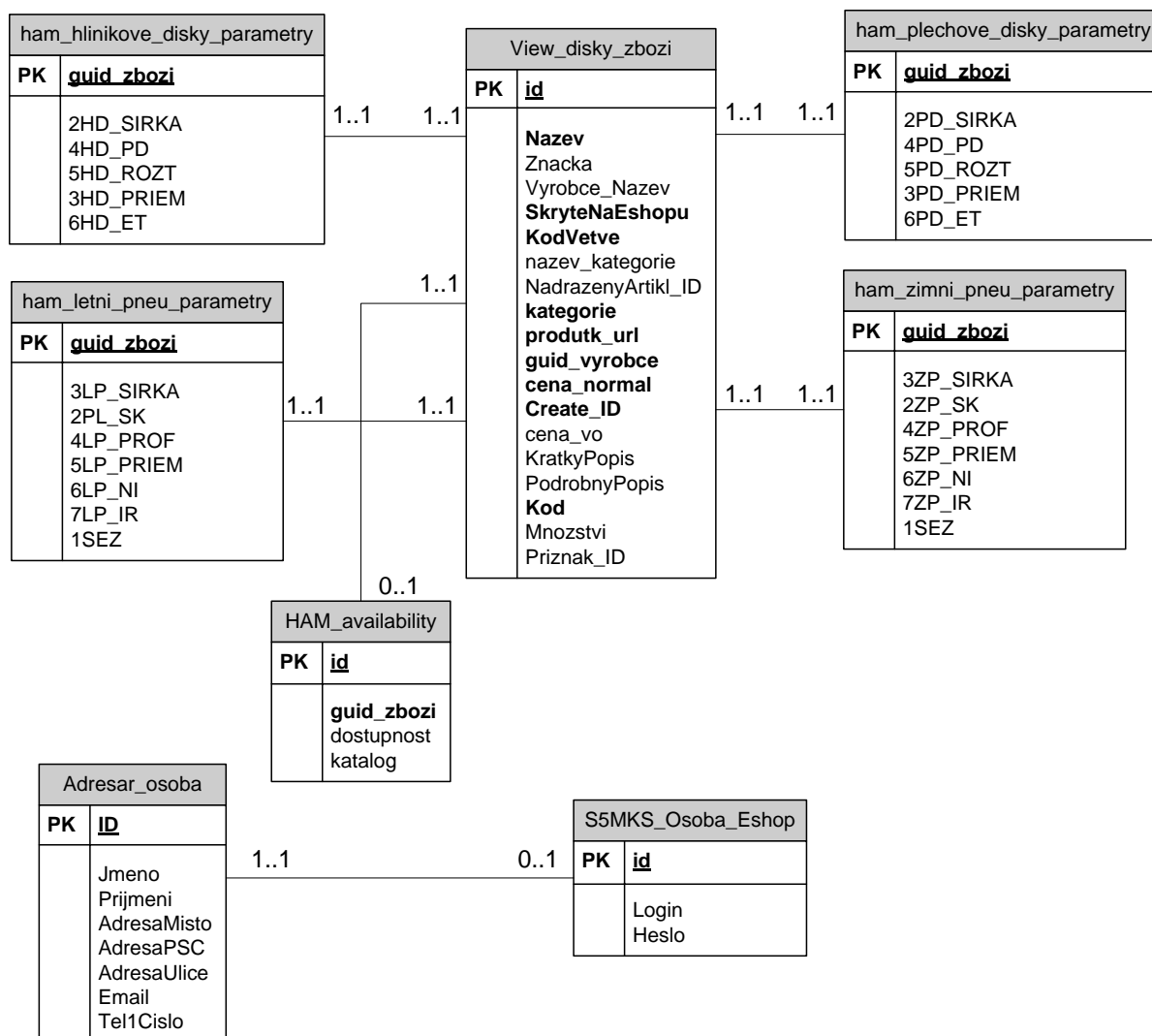
Obrázek 4.2: Diagram případů užití systému

4.2 Entity – Relationship diagram

Slouží jako datový model statických dat uložených v databázi. Na základě tohoto modelu se vytváří návrh samotné databáze. Model tvoří entity a vztahy mezi entitami. Každá entita reprezentuje tabulku v databázi. Výsledný diagram vytváří neorientovaný graf, kde uzly grafu představují entity a hrany vztahy mezi entitami. Každá entitní množina obsahuje atributy, včetně primárního klíče. Cizí klíče nejsou v entitních množinách vyznačeny.

4.2.1 E-R diagram systému

Jak již bylo několikrát řečeno, systém vychází již z existující databáze, tudíž E-R diagram nebyl vytvářen pro potřeby návrhu databáze samotné, ale pro zřehlednění a určení participujících entit a jejich vztahů. Nejedná se o model celé databáze, ale pouze o tu část databáze, kterou systém přímo využívá.



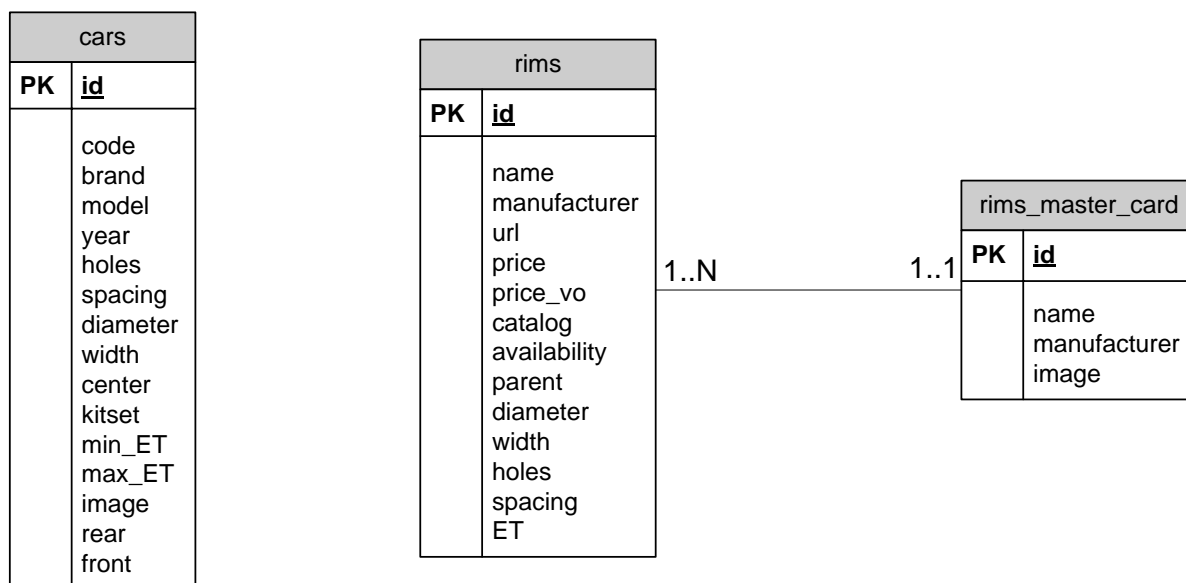
Obrázek 4.3: E-R diagram systému

Popis jednotlivých entit:

- view_disky_zbozi** – tato entitní množina představuje všechny produkty nacházející se v systému. Jedná se o vytvořený pohled, který sdružuje potřebná data o produktech do jedné entitní množiny, pro snazší přístup k těmto datům, tedy pro zrychlení dotazu. Obsahuje tři kandidátní klíče, **id**, **produkt_url** a **Kod**. Jako primární klíč byl z těchto kandidátních klíčů vybrán atribut **id**, jelikož se s ním pracuje v ostatních entitních množinách. Dalšími důležitými atributy jsou **SkryteNaEshopu**, **NadrazenyArtikl_ID** a **produkt_url**. Atribut **SkryteNaEshopu** rozlišuje produkty, které se mají zobrazit v systému a které nikoli. **NadrazenyArtikl_ID** představuje cizí klíč identifikující rodiče. Pomocí atributu **produkt_url** systém definuje URL adresu konkrétního produktu a následně pomocí tohoto atributu daný produkt vyhledává.
- ham_hlinikove_disky_parametry** – tento pohled obsahuje parametry produktů, konkrétně hliníkových disků. Jediným kandidátním klíčem je atribut **guid_zbozi**, který byl zvolen primárním klíčem a zároveň se jedná o cizí klíč. Atributy **2HD_SIRKA**, **4HD_PD**, **5HD_ROZT**, **3HD_PRIEM**, **6HD_ET** představují parametry produktů.
- ham_ocelove_disky_parametry** – tento pohled obsahuje také parametry produktů, konkrétně se jedná o ocelové disky. Opět je zde pouze jeden kandidátní klíč, který je klíčem primárním i cizím. Jedná se o atribut **guid_zbozi**, ostatní atributy představují jednotlivé parametry produktu. Oproti předchozímu pohledu se liší pouze v předponě názvu parametru, která je v této entitní množině **PD** namísto **HD**.
- ham_letni_pneu_parametry**, **ham_zimni_pneu_parametry** – jedná se dva, skoro totožné pohledy, které představují parametry produktů. Konkrétně se jedná o letní a zimní pneumatiky. Každý pohled obsahuje pouze jeden kandidátní klíč, kterým je v obou případech atribut **guid_zbozi**. Tento atribut je zároveň primárním i cizím klíčem. Ostatní atributy představují samotné parametry. V jednotlivých entitních množinách se liší pouze předponou názvu atributu, který je v případě entitní množiny **ham_letni_pneu_parametry** **LP** a v entitní množině **ham_zimni_pneu_parametry** **ZP**. Navíc se v obou tabulkách objevuje atribut **1SEZ**, který představuje číselný kód období, pro které jsou pneumatiky určeny, tedy letní a zimní období.

4.2.2 E-R diagram grafického konfigurátoru

Pro spuštění grafického konfigurátoru byla stávající databáze nedostačující z hlediska struktury. Proto bylo nutné vytvořit novou databázi, která byla navržena pro specifické požadavky konfigurátoru. Základem bylo ujasnit si, jakým způsobem bude grafický konfigurátor fungovat, a podle toho navrhnout databázi. Základní otázkou bylo, jakým způsobem budou disky přiřazeny k jednotlivým automobilům. Vyskytly se dvě možnosti jak toto spojení uskutečnit. První možností bylo modelovat mezi tabulkami **cars** a **rims** vztah o kardinalitě **M:N**. To by znamenalo, že každý automobil bude mít přesně definováno, které disky na něj pasují. Od této možnosti bylo po několika schůzkách se zadavatelem nakonec upuštěno z důvodu komplikovanějšího vkládání nových disků k automobilům a také proto, že nebyly k dispozici zdroje určující tuto relaci u všech výrobců disků. Přistoupilo se tedy k druhé variantě, která vybírala disky na základě parametrů automobilu. Toto řešení nám zajišťuje jednoduché rozšíření disků pro automobily a také odpadlo zjišťování relací mezi automobilem a diskem u výrobců disků. Byl vytvořen E-R diagram, který splňuje specifikované požadavky.



Obrázek 4.4: E-R diagram grafického konfigurátoru

Popis jednotlivých entit:

cars – tato entitní množina obsahuje informace o automobilech dostupných na konfigurátoru a tvoří základ celé aplikace. Obsahuje jeden kandidátní klíč a tím je atribut **id**, který je zároveň primárním klíčem. Ostatní atributy popisují konkrétní automobil a jeho vlastnosti. Atributy brand, model, year identifikují automobil podle značky, modelu a roku výroby. Holes, spacing, diameter, width, center, min_ET a max_ET jsou atributy definující vlastnosti automobilu jako je počet děr pro disk, rozteč šroubů, průměr, šířka disku, rozměr středového otvoru, minimální a maximální ET disku. Dále bylo potřeba ke každému automobilu přiřadit jeho obrázek a definovat pozice disků v tomto obrázku. K tomu slouží atributy image pro určení cesty k obrázku auta, rear pro určení pozice zadního disku a front pro určení pozice předního disku.

Rims – obsahuje všechny disky, které jsou v konfigurátoru dostupné. Tvoří jej atributy name a manufacturer, které definují název disku a název jeho výrobce. Atributy diameter, width, holes, spacing a ET definují vlastnosti disku, tedy jeho průměr, šířku, počet děr, rozteč šroubů a ET. Dalšími parametry jsou price, price_vo a availability, tyto parametry poskytují informace o ceně a dostupnosti disku. Atribut url určuje URL adresu disku. Atribut parent je cizím klíčem, který odkazuje do tabulky **rims_master_card**.

rims_master_card – v této entitní množině se nachází informace o modelech disků. Atribut name určuje jméno modelu, manufacturer zase výrobce modelu a atribut image obsahuje cestu k obrázku modelu disku. Cizím klíčem je atribut **id**, který je v této tabulce jediným kandidátním klíčem.

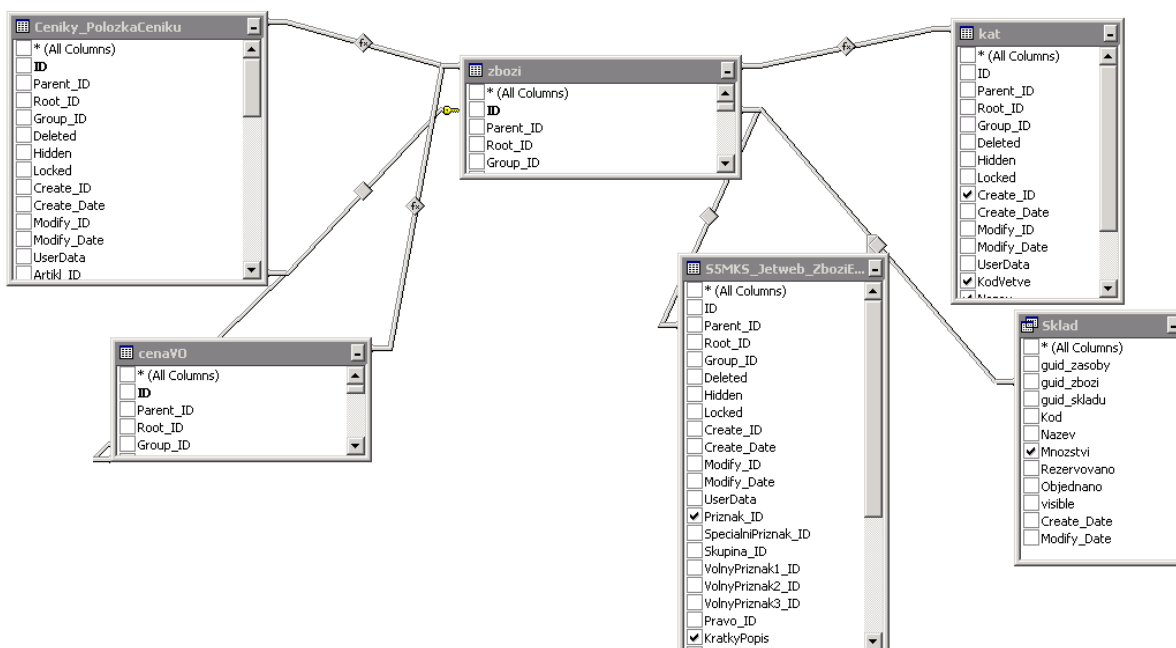
4.3 Návrh databáze

V případě celého systému návrh databáze nebyl možný, databáze již byla vytvořena a systém se jí musel přizpůsobit. Vzhledem ke struktuře databáze a jejich tabulek by výsledné dotazy na výběr produktů obsáhly asi 7 tabulek, z nichž každá obsahuje průměrně 10 000 záznamů. To by značným způsobem aplikaci zpomalovalo. Bylo tedy nutné najít nějaké řešení, které by zmenšilo počet tabulek, přes které je nutné se dotazovat. Bylo vytvořeno několik pohledů, které sdružují informace z několika

tabulek do jednoho celku. Jedná se o dva typy pohledů, jeden sdružuje informace o produktech, jako je jejich název, cena apod. Druhý typ obsahuje parametry jednotlivých produktů.

4.3.1 Pohled sdružující informace o produktech

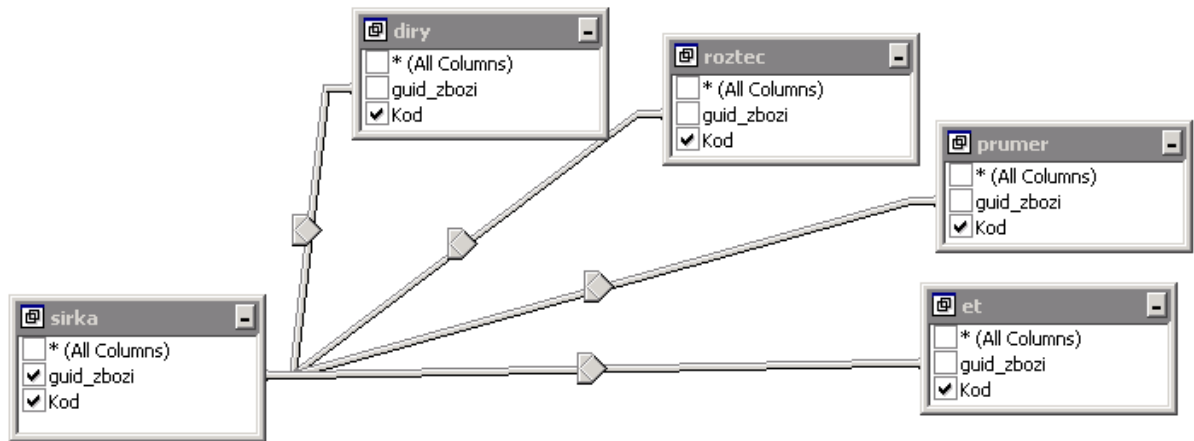
Jelikož tvůrci databáze systému Money S5 ze začátku nepočítali s integrací internetového obchodu do jejich systému, jsou informace členěny podle požadavků systému Money S5, které ale nevyhovují z hlediska optimalizace dotazů. Potřebné informace k jednotlivým produktům bylo potřeba sjednotit do jednoho pohledu. Byl tedy vytvořen pohled **view_disky_zbozi**. Tento pohled sdružuje všechny potřebné informace o produktech v na jednom místě a zmenšuje tedy množství tabulek, které je nutné procházet.



Obrázek 4.5: Struktura pohledu **view_disky_zbozi**

4.3.2 Pohled sdružující parametry produktů

Každý produkt uložený v databázi má asi 5 parametrů. Názvy těchto parametrů a jejich hodnoty jsou uloženy v jiné tabulce. Tedy kardinalita jejich vztahu M:N. Tento vztah je reprezentován tabulkou, která spojuje záznamy z obou tabulek. Jedna z požadavek, byla umožnit vyhledávání produktů pomocí všech jeho parametrů. To sice bylo možné, ale bylo to vysoce neefektivní. Bylo nutné se do tabulky s parametry dotazovat tolikrát, kolik parametrů bylo vybráno. V nejhorsím případě by se dotazoval 6krát do stejné tabulky. Bylo tedy nutné tento problém nějak vyřešit, tedy získat všechny parametry produktu dotazem do jedné tabulky. To znamenalo změnit strukturu, kde původně jeden záznam tabulky obsahoval jeden parametr produktu. Navíc názvy jednotlivých parametrů byly odlišné pro různé typy produktů. Produkty se dělí na letní a zimní pneumatiky a hliníkové a ocelové disky. Bylo tedy nutné vytvořit čtyři pohledy, pro každý typ jeden. Struktura těchto pohledů je totožná, liší se pouze předponou v názvu parametru.



Obrázek 4.6: Struktura pohledu ham_hlikove_disky_parametry

5 Implementace systému

5.1 Vytvoření databáze grafického konfigurátoru

Pro vytvoření databáze grafického konfigurátoru bylo potřeba získat data o jednotlivých automobilech a tyto data zpracovat podle navržené struktury databáze. Každý výrobce disků poskytuje velkoodběratelům soubor, ve kterém jsou uvedeny všechny modely disků, které výrobce nabízí k jednotlivým automobilům. Zároveň jsou v tomto souboru uvedeny parametry těchto disků. Z těchto údajů můžeme vyčíst, jaké parametry má daný automobil. Problém je, že každý výrobce disků používá trochu odlišnou strukturu značení automobilů a jejich modelů. Kdy z důvodu úspory velikosti souborů sdružují názvy modelů nebo typů modelů do jednoho řádku.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	Car brand	Car group	Type	from	to	PCC	CB	Wheel d.	O	Branc	Design	Article	Klase	KW	Str.		
2	ALFA ROMEO	145/146	930	1994-07	2001-01	4	98	58	5.5x14	28	DEZENT	F	TFH1S28	Z.F8	66-95	175-95	
3	ALFA ROMEO	145/146	930	1994-07	2001-01	4	98	58	5.5x14	38	DEZENT	F	TFH28	Z.F24	66-95	175-95	
4	ALFA ROMEO	145/146	930	1994-07	2001-01	4	98	58	5.5x14	32	DEZENT	L	TLH1HA32	Z.F8	66-95	175-95	
5	ALFA ROMEO	145/146	930	1994-07	2001-01	4	98	58	5.5x14	35	DEZENT	L	TLH2HA35	Z.F24	66-95	175-95	
6	ALFA ROMEO	145/146	930	1994-07	2001-01	4	98	58	5.5x14	35	DEZENT	L DARK	TLH2HA35	Z.F24	66-95	175-95	
7	ALFA ROMEO	145/146	930	1994-07	2001-01	4	98	58	5.5x14	32	DEZENT	L SI	TLH1SA32	Z.F8	66-95	175-95	
8	ALFA ROMEO	145/146	930	1994-07	2001-01	4	98	58	5.5x14	35	DEZENT	L SI	TLH2SA35	Z.F24	66-95	175-95	
9	ALFA ROMEO	145/146	930	1994-07	2001-01	4	98	58	5.5x14	32	DEZENT	RE	TREH1SA32	Z.F8	66-95	175-95	
10	ALFA ROMEO	145/146	930	1994-07	2001-01	4	98	58	5.5x14	35	DEZENT	RE	TREH2SA35	Z.F24	66-95	175-95	
11	ALFA ROMEO	145/146	930	1994-07	2001-01	4	98	58	5.5x14	35	DEZENT	RE DARK	TREH2KA35	Z.F24	66-95	175-95	

Obrázek 5.1: Ukázka souboru s informacemi o automobilech a discích výrobce ALCAR

	A	B	C	D	E	F	G	H	I	J	L	M	N	O	P	Q
1	značka	type	from	code	Výrobca	Design	Kat.číslo	katalóg M	VOC bez C	Šírka	Priemer	Počet dier	Rozteč	ET	STRED	
2	Alfa Rome	145	1994-07	930	Brook	B21	21.164098	152	88,67	7	16	4	98	35	58,1	
3	Alfa Rome	146	1994-12	930	Brook	B21	21.164098	152	88,67	7	16	4	98	35	58,1	
4	Alfa Rome	147	2001-01	937	Brook	B21	21.164098	152	88,67	7	16	4	98	35	58,1	
5	Alfa Rome MITO		2008-09	955	Brook	B21	21.164098	152	88,67	7	16	4	98	35	58,1	
6	CITROËN	NEMO	2008-05	A	Brook	B21	21.164098	152	88,67	7	16	4	98	35	58,1	
7	CITROËN	NEMO Var	2008-05	225L	Brook	B21	21.164098	152	88,67	7	16	4	98	35	58,1	
8	Fiat	500	2007-08	312	Brook	B21	21.164098	152	88,67	7	16	4	98	35	58,1	
9	Fiat	500 Cabrio	2010-01	312	Brook	B21	21.164098	152	88,67	7	16	4	98	35	58,1	
10	Fiat	BRAVA	1995-10	182	Brook	B21	21.164098	152	88,67	7	16	4	98	35	58,1	
11	Fiat	BRAVO	1995-10	182	Brook	B21	21.164098	152	88,67	7	16	4	98	35	58,1	

Obrázek 5.2: Ukázka souboru s informacemi o automobilech a discích výrobce BENET

Tato skutečnost spolu s nejednotným formátem zápisu jednotlivých modelů v souboru získání těchto údajů dost komplikuje. A to tím způsobem, že nebylo možné napsat obecný kód pro zpracování těchto souborů, ale bylo nutné ošetřit několik typů zápisu modelové řady auta u každé značky.

```
switch($last_info['model']) {
    case "Camry/Kombi":
    case "Passat/Variant":
    case "Polo Classic/Variant":
    case "Polo/GTI":
    case "Sharan II/4-motion i.V.":
        $tmp = explode("/", $last_info['model']);
        for($i=0;$i < (count($tmp)-1);$i++) {
            $models[] = $tmp[$i];
            $models[] = $tmp[$i]." ".$tmp[1];
        }
        break;
    case "Golf V/GTI/R32/4-motion":
        $tmp = explode("/", $last_info['model']);
        for($i=0;$i < (count($tmp)-3);$i++) {
            $models[] = $tmp[$i];
        }
}
```

```

        $models[] = $tmp[$i]." ".$tmp[1];
        $models[] = $tmp[$i]." ".$tmp[2];
        $models[] = $tmp[$i]." ".$tmp[3];
    }
    break;
}

```

Zdrojový kód 5.1: Úryvek zdrojového kódu pro zpracování souborů výrobců pro značku Volkswagen

Tímto jsme tedy získali jednotlivé modely a jejich typy. Dalším krokem bylo získat jednotlivé průměry a šířky disků a k nim jejich minimální a maximální ET pasující na konkrétní model auta. To už nebyl takový problém, bylo pouze nutné sledovat hodnoty v příslušných sloupcích a zpětně si je ukládat, jakmile se hodnota v sledovaném sloupci změnila, zapsali jsme do databáze poslední údaje před současně procházenou hodnotou. Tímto jsme tedy získali tabulku obsahující konkrétní modely automobilů a jejich parametry, na jejichž základě jsme poté mohli vyhledávat disky vyhovující těmto parametrům z tabulky disků.

5.2 Propojení s ERP systémem Money S5

V předchozích kapitolách jsme si uvedli co je ERP systém a jaké možnosti pro spojení poskytuje systém Money S5. Nyní si popíšeme, jak je toto spojení implementováno v této aplikaci.

Jelikož zadavatel již provozuje internetový obchod, který je realizován pomocí komerčního řešení JetWeb S5, jsou v systému zahrnuty prostředky pro správu tohoto obchodu a zobrazovaných produktů. Toto komerční řešení poskytlo tedy jakýsi základ, který jsme využili v naší aplikaci. Spojení naší aplikace se systémem Money S5 je tedy kombinací komerčního řešení JetWeb S5 a přímých dotazů do databáze systému. Využívá některých pohledů a možností importu dat vytvořených tímto komerčním systémem, avšak nepřebírá jeho plnou funkcionalitu. Neposkytuje takové možnosti správy internetového obchodu, ale pouze základní možnost nezobrazovat daný produkt. Avšak i tato volba je závislá na nastavení v komerčním systému. Tedy toto nastavení je totožné pro oba systémy. Ukážeme si příklad dotazu do databáze na výběr produktů a popíšeme si ho.

5.2.1 SQL dotaz pro výběr produktů

SQL dotaz zobrazený níže slouží pro výběr letních pneumatik s parametry šířka 195, profil 55, průměr 15, index nosnosti 85 a index rychlosti T. První tabulka kam se dotazujeme je **ham_letni_pneu_parametry**, v této tabulce jsou uloženy všechny parametry letních pneumatik v systému. Vybereme tedy ty, které vyhovují výběru. Tuto tabulku sloučíme s tabulkou **view_disky_zbozi** na základě atributu `guid_zbozi` z první tabulky a atributu `id` z tabulky druhé. Dále vybraný produkt nesmí mít nastaven atribut `SkrytyNaEshopu`, cena musí být větší než 0 a musí být přiřazen do nějaké kategorie. Tím jsme získaly všechny varianty pneumatik, které vyhovují výběru. Potřebujeme však i údaje rodiče tohoto produktu, proto se znovu dotazujeme do tabulky **view_disky_zbozi** tentokrát jsou tabulky spojeny na základě parametrů `NadrazenyArtikl_ID` a `id`. Vzhledem k tomu, že do databáze přistupují dvě aplikace, je nutné zamezit výlučnému přístupu, tedy zamykání tabulek při vykonávání dotazů. To je zajištěno příkazem **WITH(NOLOCK)**. Tím samozřejmě nastává možnost, že si jednotlivé transakce přepíší data v tabulkách. Avšak tato skutečnost nemůže nastat, jelikož ani jedna aplikace nezapíše data přímo do databáze, ale využívá importů, které data sekvenčně zapíše. K odstranění zamykání tabulek jsme přistoupili z toho důvodu, že některé dotazy trvají několik vteřin a tím pádem může dojít ke kolizi.

Některé atributy jsou datového typu `UNIQUEIDENTIFIER`, který je specifický pro MS SQL Server. Tyto atributy je nutné přetypovat z důvodu kompatibility s PHP. K přetypování se využívá funkce **CAST()**. I když využití této funkce zpomaluje zpracování dotazu musíme ji využívat.

```

SELECT Zbozi_eshop.ID,Zbozi_eshop.Nazev, CAST(Master.ID as
nvarchar(50)) as parentID,Master.Nazev as
parentNazev,Zbozi_eshop.cena_normal as cena,Zbozi_eshop.Mnozstvi as
mnozstvi,CAST(Zbozi_eshop.KratkyPopis as TEXT) as KratkyPopis FROM
(SELECT guid_zbozi FROM
[S5_Agenda_001_20100630084731].[dbo].[ham_letni_pneu_parametry]
WITH(NOLOCK) WHERE [3LP_SIRKA]='195 ' AND [5LP_PRIEM]='15 ' AND
[4LP_PROF]='55 ' AND [6LP_NI]='85 ' AND [7LP_IR]='T ' ) as Param1
JOIN (SELECT * FROM
[S5_Agenda_001_20100630084731].[dbo].[View_disky_zbozi] WITH(NOLOCK)
WHERE KodVetve LIKE '000000000000%' AND SkryteNaEshopu=0 AND
cena_normal > 0 AND Kategorie LIKE '%[01234569]%' ) as Zbozi_eshop ON
Zbozi_eshop.ID=Param1.guid_zbozi JOIN
[S5_Agenda_001_20100630084731].[dbo].[View_disky_zbozi] as Master
WITH(NOLOCK) ON Master.ID=Zbozi_eshop.NadrazenyArtikl_ID AND
Master.Kategorie LIKE '%[01234569]%' AND Master.SkryteNaEshopu=0
ORDER BY cena ASC

```

Zdrojový kód 5.2: Ukázka SQL dotazu do databáze

5.2.2 Registrace uživatelů

Registrace nových uživatelů probíhá za pomoci nástrojů pro import dat do systému Money S5. Tento nástroj je součástí JetWeb S5. Import dat probíhá pomocí XML šablon, které se ukládají do příslušné tabulky v databázi. Na databázovém serveru běží aplikace, která kontroluje, jestli nejsou v tabulce nějaké nové záznamy, pokud ano, tak XML šablonu importují do Money S5.

```

<?xml version="1.0" encoding="windows-1250"?>
  <S5Data>
    <FirmaList>
      <Firma ID="db9e6c37-f70a-42bb-b31b-70a28bbcbe29">
        <Group ID="6ABB5FA2-044D-4B47-AA27-C75EB1145A7F"/>
        <FyzickaOsoba>True</FyzickaOsoba>
        <Nazev>Babják Miroslav</Nazev>
        <Poznamka>Registrace: 01.04.2012, 10:13:38</Poznamka>
        <FaktStat_ID>ED67BBE0-F18E-48AB-A7E9-
F0A30097B28D</FaktStat_ID>
        <ObchStat_ID>ED67BBE0-F18E-48AB-A7E9-
F0A30097B28D</ObchStat_ID>
        <ProvStat_ID>ED67BBE0-F18E-48AB-A7E9-
F0A30097B28D</ProvStat_ID>
        <Adresy>
          <OdlisnaAdresaProvozovny>>false</OdlisnaAdresaProvozovny>
          <OdlisnaFakturacniAdresa>False</OdlisnaFakturacniAdresa>
          <ObchodniAdresa>
            <KodPsc></KodPsc>
            <Nazev> </Nazev>
            <Ulice> </Ulice>
            <Misto> </Misto>
            <NazevStatu> </NazevStatu>
          </ObchodniAdresa>
        </Adresy>
      </Firma>
    </FirmaList>
  </S5Data>

```

Zdrojový kód 5.3: Ukázka XML šablony pro registraci uživatelů

5.2.3 Vkládání objednávek

Vkládání objednávek do Money S5 probíhá stejným způsobem jako registrace uživatelů. Tedy pomocí XML šablony, které je nahrána do tabulky v databázi a poté zpracována. Šablonu můžeme rozdělit na tři pomyslné části a to na základní informace o objednávce, informace o objednavateli a informace o objednaných produktech.

5.2.3.1 Informace o objednávce

Zde se objevují údaje jako je ID objednávky v databázi, poznámky k objednávce, celková cena objednávky, kód objednávky, datum vystavení a její stav.

```
<ObjednavkaPrijata ID="3e526654-2993-e7e3-72f3-72ea48609866">
<Group ID="EAF3D595-618D-4B4A-8FDF-8C9CA0652CEE" />
<Stav EnumValueName="Rozpracovano">0</Stav>
<PlatnostOd>T00:00:00</PlatnostOd>
<PlatnostDo>T00:00:00</PlatnostDo>
<DatumVystaveni>2011-10-20T00:00:00</DatumVystaveni>
<Odkaz>PD02855</Odkaz>
<Nazev>Prijatá objednávka 20.10.2011 | PD02855</Nazev>
<Poznamka>----- ZPRÁVA ZÁKAZNÍKA -----
-- [Objednal: Babják Miroslav -, babjak@hamendeggs.com] ----
----- ZPRÁVA DODAVATELE -----</Poznamka>
<CelkovaCastka>200,40</CelkovaCastka>
<Suma>
  <Zaklad>167,00</Zaklad>
  <Celkem>200,40</Celkem>
  <Dan>33,40</Dan>
</Suma>
```

Zdrojový kód 5.4: Ukázka XML šablony pro vložení objednávky. Část s informacemi o objednávce

5.2.3.2 Informace o objednavateli

Informace o uživateli jsou základem celé objednávky. V souboru můžeme najít několik typů adres a uživatelé, které se můžou navzájem lišit. Systém odlišuje tři typy adres a to adresu objednavatele, adresu příjemce faktury a adresu příjemce zboží. U každé adresy je též obsaženo ID uživatele v databázi.

```
<Adresa>
  <Misto>Zlaté Hory</Misto>
  <PSC>79376</PSC>
  <Ulice>Ondřejovice 266</Ulice>
  <Nazev>Babják Miroslav</Nazev>
  <KontaktniOsobaNazev>Babják Miroslav</KontaktniOsobaNazev>
  <KontaktniOsoba ObjectName="Osoba" ObjectType="Object"
ID="76EB6516-DACE-506F-5726-8819D548F0D5"/>
</Adresa>
<AdresaPrijemceFaktury>
  <Stat>Česká republika</Stat>
  <Nazev>Babják Miroslav</Nazev>
  <Misto>Zlaté Hory</Misto>
  <PSC>79376</PSC>
  <Ulice>Ondřejovice 266</Ulice>
  <KontaktniOsobaNazev>Babják Miroslav</KontaktniOsobaNazev>
```

```

    <KontaktNiOsoba ObjectName="Osoba" ObjectType="Object"
ID="76EB6516-DACE-506F-5726-8819D548F0D5"/>
  <AdresaKoncovehoPrijemce>
    <Nazev>Babják Miroslav</Nazev>
    <Ulice>Ondřejovice 266</Ulice>
    <PSC>79376</PSC>
    <Misto>Zlaté Hory</Misto>
    <KontaktNiOsobaNazev>Babják Miroslav</KontaktNiOsobaNazev>
    <KontaktNiOsoba ObjectName="Osoba" ObjectType="Object"
ID="76EB6516-DACE-506F-5726-8819D548F0D5"/>
  </AdresaKoncovehoPrijemce>

```

Zdrojový kód 5.5: Ukázka XML šablony pro vložení objednávky. Sekce s informacemi o objednavateli

5.2.3.3 Informace o objednaném zboží

Kromě klasických údajů jako je název, počet kusů, cena za kus a podobně, jsou navíc potřebné informace o skladu, ze kterého se má zboží odebrat, ceníku ve kterém se nachází cena zboží a také ID zásoby ve skladu. Tyto informace slouží pro automatické odečtení položky ve skladu a přiřazení správné ceny zboží.

```

<Mnozstvi>4,00</Mnozstvi>
<Sleva>0,00</Sleva>
<JednCena>50,10</JednCena>
<CelkovaCena>200,40</CelkovaCena>
<DokladObjectName>ObjednavkaPrijata</DokladObjectName>
<TypCeny EnumValueName="SDani">1</TypCeny>
<Jednotka>ks</Jednotka>
<Poznamka></Poznamka>
<Nazev>155/65R13 73T WinterContact TS800</Nazev>
<Katalog>CO3531480000</Katalog>
<TypObsahu EnumValueName="SObsahem">1</TypObsahu>
<ObsahPolozky ObjectName="ObsahPolozkySArtiklem"
ObjectType="Object">
  <Artikl_ID>E9A40BD8-A019-434C-AD06-D2EDF2E22847</Artikl_ID>
  <DruhPolozky_ID>CF2308F5-C299-4224-A9E5-
AA351DE69DF5</DruhPolozky_ID>
  <Zasoba_ID>BF9E4620-10C8-4D9F-A45A-9433A4083145</Zasoba_ID>
  <Cenik_ID>C628DCE9-A03D-4070-B330-86AF32119637</Cenik_ID>
  <CenovaHladina_ID></CenovaHladina_ID>
  <Sklad_ID>9375A36F-9D18-4A2E-840A-8DBBEC1365CE</Sklad_ID>
  <Jednotka_ID>7A9A5A93-EBC5-4DD0-B310-
C166A0B2FA16</Jednotka_ID>
</ObsahPolozky>

```

Zdrojový kód 5.6: Ukázka XML šablony pro vložení objednávky. Sekce s informacemi o produktu

5.3 Výběr automobilu

Konkrétní automobil se vybírá na základě formulářového výběrového pole. Tyto pole jsou na sobě závislá, tedy vybraná položka z předchozího pole určuje, které položky se budou nacházet v dalším poli. Vzhledem k tomu, že konfigurátor neobsahuje obrázky všech automobilů, bylo nutné tyto automobily ve výběrových polích odlišit a to tím, že u těchto aut bude v polích červené pozadí.

Tento výběrový formulář implementuje komponenta s názvem `carForm`. Pro vytvoření formuláře je použita implicitní třída frameworku Nette s názvem `AppForm`. Tato třída obsahuje metody, které umožňují vkládat jednotlivé formulářové prvky a poté celý formulář jednoduše zobrazit. Neumožňuje však vytvořit závislá výběrová pole. Z toho důvodu bylo nutné využít externí komponentu `DependentSelectBox`⁵, která umožňuje do formuláře přidat tyto prvky.

```
$form->addDependentSelectBox("model", "", $form["brand"], array($this, "getValueModelSelect"))->setPrompt("Zvolte model");
```

Zdrojový kód 5.7: Kód pro vložení závislého výběrového pole do formuláře

Metoda `addDependentSelectBox` přijímá čtyři parametry v tomto pořadí:

1. Název komponenty
2. Popisek komponenty
3. Rodičovská komponenta
4. Název funkce pro naplnění výběrového pole daty

Z důvodu odlišení automobilů bylo tedy potřeba přidat XHTML značce `<option>` třídu, která toto zajistí. Nette umožňuje vytvářet výběrová pole z PHP pole typu klíč, hodnota. V našem případě to však nebylo možné. Bylo tedy nutné generovat prvky výběrového pole postupně a kontrolovat, zda je dostupný obrázek automobilu. Pro generování těchto prvků byla využita další implicitní třída frameworku Nette a to třída `Html`. Tato třída umožňuje jednoduše vytvářet XHTML prvky a ty poté přiřazovat do formuláře vytvářeného pomocí třídy `AppForm`.

```
foreach($result as $r){
    if($r['image'] == "" || $r['image'] ==
"xchybajuce_auto_sedan.png" || $r['image'] ==
"xchybajuce_auto_offroad.png" || $r['image'] ==
"xchybajuce_auto_hatchback.png" || $r['image'] ==
"xchybajuce_auto_combi.png"){
        $el = NHtml::el('option')->value($r['madeValue'])-
>setText($r['madeText']->class("red"));
    } else $el = NHtml::el('option')->value($r['madeValue'])-
>setText($r['madeText']);
    $arr[$r['madeValue']] = $el;
}
```

Zdrojový kód 5.8: Kód pro generování prvků výběrového pole

⁵ <http://addons.nette.org/cs/dependentsselectbox>

5.4 Zobrazení automobilu

V první řadě bylo nutné zvolit optimální formát obrázku. Obrázek by neměl být tak velký, aby na něm šly vidět případné chyby, ale zase musel být velký natolik, aby nezanikaly zobrazené disky. Nakonec byla zvolena velikost 1000x600px. Byla také přidána záře, která utvrzuje dojem 3D zobrazení.

Při této velikosti obrázku však vznikl problém při zobrazení na nižším rozlišení. Docházelo totiž k překrývání obrázku prvkem pod ním. Bylo tedy zapotřebí dynamicky měnit vertikální polohu obrázku podle rozlišení návštěvníka. K vyřešení tohoto problému jsme využili javascriptovou knihovnu JQuery, která vyřešení tohoto problému značně usnadnila. Nejdříve jsme si zjistili velikost dokumentu v prohlížeči, od této hodnoty jsme odečetly výšku všech prvků na stránce, které ji ovlivňují. Nakonec jsme od této hodnoty odečetli výšku obrázku automobilu. Tím jsme zjistili hodnotu v axelech, o kterou je potřeba posunout obrázek směrem nahoru. Aby byla zachována celistvost pozadí, museli jsme posunout i pozadí celé stránky.

```
var documentHeight = $(document).height();
var heightForCar = documentHeight - 223;
var backPosition = heightForCar - 550;
if(backPosition > 0)
    backPosition = 0;
if(heightForCar > 600)
    heightForCar = 600;
$("#carContainer").css({
    "height" : heightForCar+"px",
});
$("#carImage").css({
    "height" : heightForCar,
    "backgroundPosition" : "0 "+backPosition+"px"
});
$("body").css("backgroundPosition", "0 "+backPosition+"px");
$("#carImage img").css("top", "-"+Math.abs(backPosition)+"px");
```

Zdrojový kód 5.9: JQuery kód pro změnu pozice obrázku auta

5.5 Změna barvy automobilu

Jelikož zadavatel chtěl mít konfigurátor co nejvíce přizpůsobitelný ze strany zákazníka, tedy aby si zákazník mohl daný automobil maximálně přizpůsobit svému vozidlu. Z toho důvodu bylo potřeba zpracovat mechanismus, který umožní změnu barvy automobilu a to v co nejširší škále barev. Pro tyto účely byl vybrán nástroj Colorpicker, což je knihovna napsána za pomoci javascriptu a JQuery. Umožňuje výběr z široké palety barev. Současně také vybranou barvu zapsat v hexadecimálním tvaru, pro použití v CSS.



Obrázek 5.3: Obrázek Colorpickeru použitý v aplikaci

Samotná změna barvy automobilu probíhá za pomoci dvou divů umístěných přes sebe. Ve spodním divu dochází ke změně barvy podle výběru v Colorpickeru. Vrchní div slouží jako maska. Obrázek automobilu tvoří tedy silueta automobilu, která mu udává tvar a poté stíny a odlesky, které dodávají automobilu realističnost. Je to velmi jednoduchý princip, avšak vyžaduje pečlivý výběr fotografií, ze kterých se obrázky tvoří a samotný proces vytvoření masky je velmi náročný.



Obrázek 5.4: Maska automobilu

5.6 Implementace horizontálního posuvníku

Uživatelské rozraní je koncipováno tak, že návštěvník má vybraný automobil stále před sebou. Museli jsme tedy zvolit takový způsob zobrazení disků, který odpovídá této koncepci. Zvolili jsme tedy variantu horizontálního posuvníku, který nám umožní zobrazit všechny disky a zároveň vybraný automobil zůstává stále před očima návštěvníka. Je k dispozici spousta knihoven implementující takový posuvník, ale žádná nevyhovovala našim požadavkům. Museli jsme tedy vytvořit vlastní jednoduchý posuvník.

Posuvník využívá celou šířku okna prohlížeče, jeho velikost se tedy dynamicky mění v závislosti na rozlišení obrazovky návštěvníka. Abychom určili velikost posuvníku, bylo potřeba definovat šířku plochy, kterou bude využívat jeden prvek posuvníku. Jakmile jsme znali tento údaj, mohli jsme vypočítat, kolik prvků bude zobrazeno na stránce a podle toho určit šířku posuvníku.

```
$("#slider-container").width((rimsOnPage() * 120));
$("#slider-container ul").width($("#slider-container
ul").children().length * 120 + 120);
```

Zdrojový kód 5.10: JQuery kód pro výpočet šířky posuvníku

Jakmile jsme znali počet zobrazovaných prvků na stránce, tedy i šířku posuvníku, mohli jsme přistoupit k programování funkcí zajišťující samotný posuv. Pro posuv je implementována funkce, která jako parametr přijímá textový řetězec určující směr posunutí. Funkce je přednastavená, aby posouvala vždy o 5 prvků. Nejdříve zkontroluje kolik prvků je ještě potřeba zobrazit, pokud je jich více jak 5, tak posune posuvník o 600px (5 * velikost prvku). Pokud je k zobrazení méně jak 5 prvků posune posuvník o počet prvků krát velikost prvku. Je-li počet prvků k zobrazení roven nule, vrátí se posuvník opět na začátek.

```
if(direction == "right"){
    var rimsToShow = allRims - rimsOnPage() - (Math.abs(left) /
120);
    if(rimsToShow > 5){
        var move = 600;
    } else var move = rimsToShow * 120;
    if(rimsToShow == 0){
        newLeft = 0;
        $("#prev-arrow").addClass("hidden");
    } else {
        newLeft = left - move;
        $("#prev-arrow").removeClass("hidden")
    }
}
$("#slider-container ul").animate({"left": newLeft},600);
```

Zdrojový kód 5.11: JQuery kód pro posuv posuvníku doprava

5.7 Změna velikosti disku

Zadavatel vyžadoval co největší autentičnost zobrazení, s tím souvisí i změna velikost disků na zobrazovaném automobilu. Průměr disku totiž také dost zásadním způsobem ovlivňuje výběr zákazníka. Bylo tedy nutné měnit velikost disků, podle vybraného průměru. Pro implementaci změny velikosti a pozice disku byly využity prostředky javascriptového frameworku JQuery.

V první fázi bylo nutné vypočítat velikosti disků pro jednotlivé rozměry. O výpočet se stará funkce `setDiameterSize()`, která pro každý možný průměr vytvoří objekt obsahující informace o velikosti a pozici disku. Tyto objekty ukládá do globální proměnné pro snadný přístup z funkce implementující změnu velikosti. Rozměry pro jednotlivé průměry disků jsou měněny na základě koeficientu, kterým se násobí rozdíl mezi současným a novým průměrem. Pokud se změní velikost je také nutné změnit pozici disku. Velikost posunutí je určena jako rozdíl maximální velikosti disku a velikosti pro daný průměr, rozdíl těchto hodnot se poté ještě vydělí 2, pro rovnoměrné posunutí.

```
$("#car-diameters .diameter").each(function(index){
    var diameter = parseInt($(this).attr("rel"));
    tmp = new Object();
```

```

    tmp.widthRear = maxWidthRear - (maxDiameter - diameter) *
4;
    tmp.heightRear = maxHeightRear - (maxDiameter - diameter) *
4;
    tmp.leftRear = maxLeftRear + ((maxWidthRear - tmp.widthRear)
/ 2);
    tmp.topRear = maxTopRear + ((maxHeightRear - tmp.heightRear)
/ 2);
    tmp.widthFront = maxWidthFront - (maxDiameter - diameter) *
4;
    tmp.heightFront = maxHeightFront - (maxDiameter - diameter)
* 4;
    tmp.leftFront = maxLeftFront + ((maxWidthFront -
tmp.widthFront) / 2);
    tmp.topFront = maxTopFront + ((maxHeightFront -
tmp.heightFront) / 2);
    diametersSize[diameter] = tmp;
});

```

Zdrojový kód 5.12: JQuery kód určující velikost a pozice disků pro jednotlivé průměry

Samotnou změnu velikosti disků zajišťuje funkce **resizeRims()**, která přijímá jeden parametr a to průměr disku na který se má změnit. Tato funkce kontroluje, zda se jedná o ilustrační obrázek auta nebo o reálnou fotografii. Při ilustračních obrázcích ke změně velikosti disků nedochází, protože hlavním prvkem se stávají disky samotné a je tedy nutné, aby byl zobrazen v co největším možném rozměru. Samotná změna velikosti je realizována pomocí JQuery funkce **animate()**, která animuje změnu velikosti prvku, podle zadaných CSS vlastností. Tyto parametry funkce získá z globální proměnné **diameterSize**, do které byly dříve vloženy velikosti disků v jednotlivých průměrech.

```

$("#rearRims").animate({
    left: diametersSize[diameter].leftRear,
    top: diametersSize[diameter].topRear,
    width: diametersSize[diameter].widthRear,
    height: diametersSize[diameter].heightRear
}, 500);

```

Zdrojový kód 5.13: JQuery kód pro změnu velikosti disku

6 Testování v reálném nasazení

Jelikož testování při programování aplikace nikdy neodhalí všechny chyby, bylo nutné aplikaci otestovat nezávislou skupinou, která bude složena z různorodých jedinců. Toto testování mělo za úkol odhalit, jak se budou uživatelé chovat na stránce, zda jsou hlavní prvky správně rozloženy a zda práce s nimi nepředstavuje problém.

Jakmile byla aplikace spuštěna, vybrali jsme skupinu asi 20 lidí, kteří prováděli testování. Tito lidé byly vybráni z řad přátel a známých, při výběru jsme brali v potaz věk, úroveň znalosti práce s počítačem, ale také znalost problematiky nákupu disků a pneumatik. Testování probíhalo takovou formou, že jsme každému uživateli zadali stejný úkol a pozorovali jeho chování na stránce.

Vybrané úkoly při testování:

1. Přečtěte si obchodní a dodací podmínky
2. Vyhledejte disky na automobil Audi A6 od r. 2011
3. Změňte barvu automobilu na tmavě zelenou
4. Nakupte disk Dezent RG Dark v průměru 17“ a ET 40 na tento automobil
5. Nakupte pneumatiky Michelin Primaci 3 v rozměru 225/50/R17/98
6. Přejděte do nákupního košíku
7. Zaregistrujte se
8. Proveďte nákup

Po zhodnocení těchto úkolů jsme dospěli k pár změnám v uživatelském rozhraní. Přesunuli jsme některé prvky na lepší pozice, aby je návštěvník snáze našel. Také jsme zjistili, že většina lidí se orientuje v základních parametrech pneumatik a kladou menší důraz na index rychlosti a nosnosti pneumatik. S parametry disků už byl výsledek horší, málo kdo věděl, co znamená parametr ET, avšak všem se podařilo najít a zakoupit daný disk. Přejít od nákupního košíku, registrace a nákup nečinila problémy nikomu. Přejít z grafického konfigurátoru zpět na portál internetového obchodu však činil větší části problémy, je tedy potřeba přehodnotit umístění navigace a její zvýraznění.

S výsledky testování jsme byli spokojeni a naplnilo naše očekávání. Jedinou nesrovnalostí, kterou testování odhalilo je umístění odkazů zpět do internetového obchodu v části grafického konfigurátoru. Tento problém byl provizorně vyřešen přidáním odkazu na logo v hlavičce konfigurátoru a pracujeme na přepracování navigace.

7 Závěr

Aplikace byla vypracována podle požadavků zákazníka v plném rozsahu. Samozřejmě, že tímto vývoj aplikace nekončí a už se chystají nové funkce a vylepšení. Zákazník plánuje rozšíření tohoto portálu i do dalších evropských zemí. Věřím, že aplikace má k tomu dobrý potenciál, avšak předtím je potřeba aplikaci ještě doladit. V současné době zaostává hlavně co se týče zpracování SEO a rychlosti internetového obchodu.

Aplikace je spuštěna v ostrém provozu na adrese <http://www.disky.sk> a každý den na ni přichází okolo 200 návštěvníků a tento počet se stále zvyšuje. Grafický konfigurátor je dostupný jak ze stránky internetového obchodu, tak i na adrese <http://konfigurator.disky.sk>. Zákazník je s aplikací více než spokojen, avšak má-li naplnit své ambice je potřeba na systému stále pracovat.

Práce na této aplikaci pro mě jako programátora byla velkým přínosem. Integraci na ERP systém Money S5 byla neocenitelná zkušenost, kterou můžeme využít i v dalších projektech. Vývoj aplikace nebyl zrovna procházkou růžovou zahradou, přinesl spoustu problému, hlavně co se týče databáze a její rychlosti. Zákazník měl někdy takové požadavky, že jsem nevěřil, že je dokážu splnit, avšak nakonec jsem nějaké řešení vždy našel. Dalším velkým přínosem bylo osvojení si frameworku Nette, který rozhodně usnadnil programování celé aplikace. S Nette jsem začínal a proto byly začátky opravdu obtížné a časově náročnější. Avšak jakmile jsem se naučil používat nástroje obsažené v tomto frameworku, bylo programování aplikace o dost jednodušší a přehlednější než normálně.

Jak jsem již poukazoval, aplikace se dále vyvíjí a jsou už přichystaná další rozšíření. Jako první krok je však nutné přepracovat internetový obchod do frameworku Nette, pro snadnější zapracování chystaných změn a zpřehlednění zdrojového kódu. Také je potřeba provést důkladnou SEO analýzu a podle výsledků upravit aplikaci, tak aby se umísťovala na co nejvyšších pozicích. Je také žádoucí využít sociálních sítí, pro zviditelnění portálu, takže se chystají různé soutěže a hlavně možnost sdílení obrázků automobilu z grafického konfigurátoru. Jelikož portál má ambice stát se evropským, je také nutné zapracovat automatické nástroje pro vyhodnocování objednávek, sledování cen jiných obchodů apod. Pro manažerské vyhodnocování bude nutné vytvářet velmi specifické statistické přehledy. Grafický konfigurátor plánujeme rozšířit o prodej nálepek, které si bude moci návštěvník sám vybrat a umístit na auto nebo si je dokonce navrhout. Možností pro rozšíření je spousta a jestli zákazník chce naplnit svoje cíle, je nutné tuto aplikaci dále rozvíjet a vymýšlet stále nové a nové věci.

Literatura

- [1] HRUŠKA, T.; BURGET, R.: *Internetové aplikace (WAP) IV., část Programování serveru (PHP)*. Brno: FIT VUT v Brně, 2007
- [2] MICROSOFT.CZ: *Microsoft SQL server* [online], [rev. 2012], [cit. 2012-04-23]. Dostupné z: <http://www.microsoft.com/sqlserver/cs/cz/default.aspx>
- [3] ORACLE CORPORATION: *MySQL Development History*. [online], [rev. 2010] [cit. 2012-04-23]. Dostupné z: <http://dev.mysql.com/doc/refman/5.1/en/development-history.html>
- [4] RIEHLE, Dirk. *Framework Design: A Role Modeling Approach*. Ph.D. Thesis, No. 13509. Zürich, Switzerland, ETH Zürich, 2000.
- [5] STOUPA, Václav: *Přehled a vývoj PHP frameworků* [online], [rev. 2008-03-28], [cit. 2012-04-23]. Dostupné z: <http://www.root.cz/clanky/prehled-a-vyvoj-php-frameworku/>
- [6] GRUNDL, David: *Začínáme s Nette framework* [online], [rev. 2009-03-10], [cit. 2012-04-23]. Dostupné z: <http://www.zdrojak.cz/clanky/nette-framework-zvyste-svoji-produktivitu/>
- [7] WEBOVE-APLIKACE.COM: *MVC framework* [online], [cit. 2012-04-26]. Dostupné z: <http://www.webove-aplikace.com/article/mvc-framework>
- [8] DANĚK, Petr: *Velký test PHP frameworků* [online], [rev. 2008-09-11], [cit. 2012-04-27]. Dostupné z: <http://www.root.cz/clanky/velky-test-php-frameworku-zend-nette-php-a-ror/>
- [9] BURGET, R; ZEMAN, D.: *Tvorba webových stránek ITW*. Brno: FIT VUT v Brně, 2006
- [10] SHOPCENTRIK.SZ: *ERP systém* [online], [cit. 2012-04-27]. Dostupné z: <http://www.shopcentrik.cz/slovník/erp-system.aspx>
- [11] ELENOSOFT.COM: *ERP* [online], [cit. 2012-04-27]. Dostupné z: <http://www.elensoft.com/erp.html>
- [12] MONEY.CZ: *Money S5* [online], [rev. 2010], [cit. 2012-04-27]. Dostupné z: <http://www.money.cz/money-s5>
- [13] MONEY.CZ: *JetWeb S5* [online], [rev. 2010], [cit. 2012-04-27]. Dostupné z: <http://www.money.cz/jetweb-s5/>

Seznam příloh

Příloha 1. CD se zdrojovými soubory