

Moduly ekonomického systému s architekturou klient-server

Modules of economical system with client-server architecture

Radek Vala

Bakalářská práce
2007



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně

Fakulta aplikované informatiky

Ústav aplikované informatiky

akademický rok: 2006/2007

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Radek VALA**
Studijní program: **B 3902 Inženýrská informatika**
Studijní obor: **Informační technologie**

Téma práce: **Moduly ekonomického systému s architekturou klient-server**

Zásady pro vypracování:

Vytvořte následující moduly pro ekonomický systém s architekturou klient-server:

- modul instalátoru, který zajistí snadnou instalaci celého systému na libovolný server s PHP a MySQL
- modul pro fakturaci
- modul kniha jízd s rozhraním wap
- modul pro exporty a importy dat z/do jiných systémů pomocí formátů XML, CVS, HTML

Rozsah práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. Zimáček, T.: **Skladový modulární systém s architekturou klient-server, diplomová práce, 2006, UTB ve Zlíně**
2. Smejkal, J.: **Podvojně účetnictví pro neziskové organizace, vydal: Asociace neziskových organizací Ústeckého kraje, 2005**
3. **Wireless Application Protocol Architecture Specification, <http://www.wapforum.org/what/technical.htm>**

Vedoucí bakalářské práce:

Ing. Tomáš Dulík
Ústav aplikované informatiky

Datum zadání bakalářské práce:

13. února 2007

Termín odevzdání bakalářské práce:

24. května 2007

ABSTRAKT

Cílem této práce je rozšířit víceuživatelský ekonomický systém EkonomOS o moduly Kniha jízd, Faktury a modul pro export dat ze systému do formátů XML, CSV a HTML. Snahu vytvořit ze systému EkonomOS silný a především uživatelsky přívětivý nástroj pro vedení podnikové agendy dále naplňuje instalátor systému, který umožní i laikovi velice jednoduše systém nainstalovat po nahrání souborů přes FTP např. na webový server. Systém se snaží vyjít uživateli maximálně vstříc a zefektivnit jeho práci a proto bylo vytvořeno také rozhraní pro přístup k modulu Kniha jízd přes mobilní zařízení pomocí technologie WAP.

Klíčová slova: ekonomický systém, EkonomOS, programování, databáze, klient-server, modulární systém, PHP, MySQL, XHTML, WML, WAP, Kniha jízd, Faktury, export dat, XML, CSV

ABSTRACT

This bachelor thesis is aimed at the upgrading the economical system EkonomOS, which means implementing the new modules “Kniha jízd” (ride book), “Faktury” (invoices), and module for data exporting into XML and CSV format. The new installation system makes from EkonomOS powerful and user-friendly system, and lets an ordinary person started the work very quickly. For the more effective work there is a WAP interface for handling with the “Kniha jízd” module.

Keywords: economical system, EkonomOS, programming, database, client-server, modules, PHP, MySQL, XHTML, WML, WAP, Ride book, Invoices, Data exporting, XML, CSV

Děkuji svému vedoucímu práce panu Ing. Tomáši Dulíkovi za věnovaný čas, který mi poskytl při zpracování mé bakalářské práce.

Prohlašuji, že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků, je-li to uvolněno na základě licenční smlouvy, budu uveden jako spoluautor.

Ve Zlíně

.....
Podpis diplomanta

OBSAH

ÚVOD	9
I TEORETICKÁ ČÁST	10
1 EKONOMOS A OBDOBNÉ SYSTÉMY NA TRHU	11
1.1 SOUČASNÁ SITUACE NA TRHU	11
1.2 EKONOMICKÉ OPEN-SOURCE SYSTÉMY	11
2 TECHNOLOGIE POUŽÍVANÉ V NOVÝCH MODULECH	12
2.1 ADODB – DATABASE ABSTRACTION LIBRARY FOR PHP (AND PYTHON).....	12
2.1.1 BSD licence.....	12
2.1.2 Co je to ADOdb	12
2.1.3 Výhody použití ADOdb	13
2.1.4 Stažení ADOdb	13
2.2 WAP	13
2.2.1 Wireless Markup Language (WML).....	14
2.3 FORMÁT CSV (COMMA-SEPARATED VALUES)	14
2.3.1 Popis formátu CSV v rfc4180.....	14
2.3.2 Oddělovače používané formátem CSV.....	14
2.4 XML (EXTENSIBLE MARKUP LANGUAGE).....	15
2.4.1 Specifikace XML	15
2.4.2 Synaxe XML	15
3 PŘÍNOS NOVÝCH MODULŮ A JEJICH LEGISLATIVNÍ ASPEKTY	17
3.1 MODUL KNIHA JÍZD.....	17
3.1.1 Povinné údaje v knize jízd	17
3.2 MODUL FAKTURY	18
3.2.1 Povinné údaje účetního dokladu	18
II PRAKTICKÁ ČÁST	19
4 INSTALÁTOR SYSTÉMU	20
4.1 PRŮVODCE INSTALÁTOREM	20
4.2 PRINCIP INSTALÁTORU	22
4.2.1 Import dat do databáze	22
4.2.2 Vytvoření přístupového účtu administrátora.....	23
5 MODUL KNIHA JÍZD	24
5.1 PRŮVODCE MODULEM KNIHA JÍZD	24
5.1.1 Vozový park	25
5.1.1.1 Vložení záznamu o voze	25
5.1.1.2 Výpis vozového parku	26
5.1.1.3 Editace a mazání záznamů	26
5.1.1.4 Detailní výpis údajů o voze	26
5.1.2 Kniha jízd	27
5.1.2.1 Vkládání záznamu o jízdě.....	27

5.1.2.2	Výpis záznamů v knize jízd	28
5.1.2.3	Editace a mazání záznamů	29
5.1.2.4	Detailní výpis cesty	29
5.1.3	WAP rozhraní Knihy jízd	31
5.1.3.1	Vkládání záznamů	31
5.1.3.2	Výpis údajů z knihy jízd	32
5.2	DŮLEŽITÉ TŘÍDY A FUNKCE	33
5.2.1	Třídy v souboru class.php	33
5.2.1.1	Class Database	33
5.2.1.2	Class Vuz	33
5.2.2	Funkce v souboru funkce.php	35
5.2.2.1	Funkce orderBy(\$sort,\$posl)	35
5.2.2.2	Funkce vratPalivo(\$cislo)	35
6	MODUL FAKTURY	36
6.1	PRŮVODCE MODULEM FAKTURY	37
6.1.1	Fakturované služby	37
6.1.1.1	Přidání nového typu služby	37
6.1.1.2	Zobrazení seznamu fakturovaných služeb	38
6.1.2	Faktury	39
6.1.2.1	Tvorba faktury	39
6.1.2.2	Výpis faktur	41
6.2	DŮLEŽITÉ TŘÍDY A FUNKCE	42
6.2.1	Třídy v souboru class.php	42
6.2.1.1	Class Database	42
6.2.1.2	Class Faktura	42
7	EXPORT A IMPORT DAT	44
7.1	PRŮVODCE MODULEM EXPORT CSV	44
7.2	PRINCIP EXPORTU DAT DO CSV	44
7.2.1	Problém s relacemi	44
7.2.2	Řešení - export pro každý modul zvlášť a jeho výhody	45
7.3	PRŮVODCE MODULEM IMPORT CSV	46
7.3.1	Struktura importovaného CSV	46
7.4	PRINCIP IMPORTU DAT Z FORMÁTU CSV	46
7.5	PRŮVODCE MODULEM EXPORT XML/HTML	47
7.6	PRINCIP EXPORTU DAT DO XML/HTML	47
7.6.1	Export faktur do XML	47
7.6.2	Export faktur do HTML	47
8	TISK V SYSTÉMU EKONOMOS	48
8.1	PRINCIP SKRYTÍ PRVKŮ NA TISKOVÉM VÝSTUPU	48
	ZÁVĚR	49
	CONCLUSION	50
	SEZNAM POUŽITÉ LITERATURY	51

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	52
SEZNAM OBRÁZKŮ	53
SEZNAM PŘÍLOH.....	54

ÚVOD

V roce 2006 vznikl jako bakalářská práce studentů Davida Malaníka a Tomáše Zimáčka systém EkonomOS. Jedná se o víceuživatelský systém pro vedení podnikové agendy, který je napsán v jazyce PHP a využívá databáze MySQL. Podrobnou dokumentací jádra systému se zabývají právě bakalářské práce jeho tvůrců. Hlavním důvodem vzniku systému EkonomOS byl především nedostatek podobných ekonomických systémů, které by byly snadno dostupné i menším firmám, či neziskovým organizacím.

Cílem mé práce je vytvořit z původní distribuce systému EkonomOS, jejíž instalace byla pro běžného uživatele velmi náročná, produkt, který uvítá laického uživatele jednoduchým instalátorem a umožní mu velmi rychle a jednoduše celý systém zprovoznit.

Mezi další body mé práce patří tvorba nových modulů, které EkonomOS ještě více přiblíží drahým konkurenčním systémům a nabídnou firmám další praktické nástroje pro vedení jejich podnikové agendy. Oproti původní verzi tedy uživatel bude moci využít modul Kniha jízd, pro vedení záznamů o služebních jízdách, který navíc disponuje rozhraním WAP pro mobilní přístup, dále modul Fakturace, který umožní velmi snadné a rychlé generování Faktur, díky propojení se skladovým modulem, modulem s adresami a modulem s číselníky, a také nástroje pro export a import dat.

I. TEORETICKÁ ČÁST

1 EKONOMOS A OBDOBNE SYSTÉMY NA TRHU

Jedním z hlavních důvodů vzniku systému EkonomOS byl nedostatek podobných ekonomických víceuživatelských systémů, které by byly dostupné i pro menší firmy či neziskové organizace, které si nemohou dovolit platit drahé licence.

1.1 Současná situace na trhu

Přestože uplynul takřka rok od zrodu první verze systému EkonomOS, mnoho konkurence se za tu dobu neobjevilo. Na internetu je sice dnes na výběr hned několik systémů pro vedení podnikové agendy, které mají podobnou architekturu klient-server, jako EkonomOS, většinou jsou ovšem psány v jazyce ASP či ASP.NET a jde o drahý software, určený větším podnikům. Levnější modifikace takového systému většinou nabízejí jen základní funkce a pro neziskové organizace mohou být i přes svou nižší cenu stále nedostupné.

1.2 Ekonomické open-source systémy

Přestože EkonomOS v současné verzi nabízí mnoho modulů, které používají velmi drahé konkurenční systémy, kvalitou a množstvím dalších uživatelských funkcí se s profesionálními systémy zatím nemůže srovnávat. Velkou výhodou systému EkonomOS je však fakt, že je open-source, čili volně dostupný. Měl by být tedy srovnáván s podobnými open-source systémy, kterých je ovšem opravdu velmi málo. Na českém internetu není k nalezení dokonce žádný srovnatelný systém psaný v PHP, který by nabízel výhody víceuživatelského prostředí a mírného licencování.

Firma Business systems (<http://www.bsys.cz/>) nabízí ekonomický open-source systém COMPIERE, který je však naprogramován v jazyce JAVA. COMPIERE nabízí např. tyto funkce: komplexní účetnictví společnosti, správa skladového hospodářství, nákupy, řízení dodávek, správa životního cyklu prodeje, webový portál pro řízení zákazníků a dodavatelů (Self-service), internetový obchod, sledování a analýza obchodní výkonnosti, fakturace a platby, vlastní vývojový framework umožňující měnit a rozšiřovat funkcionalitu. Jedná se o kvalitní produkt, který je vyvíjen dlouhou dobu. Jeho instalace je však natolik komplikovaná, že většina uživatelů raději zvolí některou placenou distribuci.

2 TECHNOLOGIE POUŽÍVANÉ V NOVÝCH MODULECH

Nové moduly systému EkonomOS využívají kromě jazyka PHP a XHTML některé další technologie, jež budou popsány v následujícím textu. Jak již bylo zmíněno, je to především databázová mezivrstva ADOdb obstarávající každé připojení k databázi vycházející z nového modulu, dále také WML jazyk, pomocí něž bylo napsáno rozhraní pro přístup k modulu Kniha jízd například z mobilního telefonu a dále bude popsán také značkovací jazyk XML, který je použit pro export dat z databáze.

2.1 ADOdb – Database Abstraction Library for PHP (and Python)

Přístup k databázím není v jazyce PHP nijak standardizován, což vedlo k potřebě vytvořit určitou mezivrstvu, která by setřela rozdíly mezi konektivitou jednotlivých druhů databázových systémů, a umožnila tak programátorovi vytvořit aplikaci, která je jednoduše přenositelná mezi databázovými systémy.

Tuto potřebu se snaží naplnit právě knihovna ADOdb, která byla vytvořena pro jazyk PHP a Python a je šířena pod BSD licencí, což umožňuje její použití i v komerčních projektech.

2.1.1 BSD licence

BSD licence je licencí pro svobodný software, která je jednou z nejsvobodnějších. Umožňuje volné šíření licencovaného obsahu, přičemž vyžaduje pouze uvedení autora a informace o licenci, spolu s upozorněním na zřeknutí se odpovědnosti za dílo.

Zkratka BSD označuje „Berkeley Software Distribution“ – obchodní organizaci při University of California, Berkeley, která tuto licenci vyvinula a používala pro práce nad operačním systémem BSD.

2.1.2 Co je to ADOdb

ADOdb je objektová knihovna, podobná ADO od Microsoftu, jež má určitá vylepšení, která ji dělají unikátní (např. kontingenční tabulky (pivot tables), generování HTML kódu pro vícestránkové výpisy s odkazy *předchozí* a *další*, cacheování výpisů, generování HTML menu, atd.). [1]

2.1.3 Výhody použití ADOdb

- Rychlost. Jedná se pravděpodobně o nejrychlejší open-source knihovnu pro práci s databázemi. Benchmarky jsou k nahlédnutí na: <http://phplens.com/lens/adodb/>.
- Poskytuje rozsáhlou podporu přenositelnosti SQL dotazů napříč databázovými systémy. Typy na tvorbu přenositelných SQL dotazů jsou publikovány na adrese: http://phplens.com/lens/adodb/tips_portable_sql.htm.
- Jednoduchost psaní kódu, zvláště máte-li zkušenosti s používáním ADO konvencí.
- QA (Quality assurance) – záruka kvality, každé vydání je testováno na databázových systémech Access, MySQL, PostgreSQL, MS SQL, Oracle 9.
- Rozsáhlá komunita uživatelů – vyvíjeno od srpna 2000.
- Velmi mírné licencování (BSD). [2]

2.1.4 Stažení ADOdb

Verzi pro PHP 4.1.0 nebo pozdější, fungující i s PHP5, je možno stáhnout zde: http://sourceforge.net/project/showfiles.php?group_id=42718.

2.2 WAP

WAP (Wireless Application Protocol) je síťový komunikační protokol vytvořený organizací WAP forum (<http://www.wapforum.org>) pro připojení mobilních zařízení k internetu. WAP je velmi podobný WWW architektuře. **Wireless Application Environment (WAE)** je nejvýše položená vrstva je založena na kombinaci WWW a technologie mobilních přístrojů. Jejím snahou je nabídnout programátorské prostředí, které umožní rychle vyvíjet aplikace a služby nezávislé na široké škále bezdrátových platform. WAE zahrnuje prostředí mikroprohlížeče obsahující následující funkce:

- **Wireless Markup Language (WML)** - odlehčený značkovací jazyk, podobný HTML, ale optimalizovaný pro použití na prohlížečích mobilních zařízení.
- **WMLScript** - odlehčený skriptovací jazyk, obdoba JavaScriptu.
- **Wireless Telephony Application (WTA, WTAI)** - telefonní služby a programátorské rozhraní.

- **Content Formats** - množina nadefinovaných datových formátů, zahrnující obrázky, záznamy telefonního seznamu a naplánované úkoly.

2.2.1 Wireless Markup Language (WML)

Jak bylo řečeno, svět mobilního internetu je technicky velmi podobný tomu klasickému, takže i WML je jakousi odlehčenou verzí HTML (ovšem plně vycházející ze XML), lépe splňující omezení mobilních internetových prohlížečů. WML se ještě s dalšími technologiemi nachází v té nejvyšší – aplikační vrstvě – WAPu. Jeho možnosti jsou rozšířeny WMLScriptem (ekvivalent JavaScriptu) a programátor také může využít technologie WTA pro přístup k službám telefonu. Kombinace zmíněných technologií je poměrně silným nástrojem ve světě mobilních informací, což posouvá možnosti využití WML stránek od informačních portálů až k podnikovým systémům, jako je EKONOMOS, kde přístup z mobilního telefonu může velmi zefektivnit vedení některých záznamů, či získávání informací „v terénu“ (např. modul Kniha jízd).

2.3 Formát CSV (Comma-separated values)

CSV je velmi jednoduchý souborový formát pro předávání tabulkových dat. Jak vypovídá překlad anglického názvu „čárkami oddělené hodnoty“, byly jednotlivé záznamy původně oddělovány pouze čárkami. Dnes se ovšem používá mnoho modifikací původního CSV, které se liší pouze oddělovačem. Přestože je formát CSV velice často používaný a běžný, nebyl nikdy normalizován.

2.3.1 Popis formátu CSV v rfc4180

Popis formátu CSV se nachází mimo jiné v rfc4180. Podle rfc4180 je pro CSV registrován MIME typ *text/csv*. Běžně se však lze setkat s dalšími typy jako je *application/csv*, *text/x-csv* a další.

2.3.2 Oddělovače používané formátem CSV

Jednotlivé hodnoty jsou nejčastěji oddělovány těmito oddělovači:

- středník (používá MS Excel),
- čárka (původní oddělovač formátu CSV),

- tabulátor (někdy označováno jako TSV – *Tab-separated values*),
- tečka.

V případě, že použitý oddělovač je obsažen v textu hodnoty, musí být tento text uzavřen do uvozovek, jinak bude rozdělen.

Jednotlivé řádky záznamů jsou odděleny ukončením řádku.

2.4 XML (eXtensible Markup Language)

XML – volně přeloženo z angličtiny jako rozšiřitelný značkovací jazyk – se stává v poslední době často používaným jazykem především pro výměnu dat mezi aplikacemi a pro publikování dokumentů.

XML je jednouchý, velice flexibilní textový formát vycházející ze SGML (ISO 8879). Původně byl vyvinut pro elektronické publikování, dnes ale hraje důležitou roli při výměně dat nejen na webu. [3]

2.4.1 Specifikace XML

Jazyk XML byl vyvinut a standardizován konsorciem W3C (<http://www.w3c.org/>). Tato specifikace je volně dostupná a proto může být XML využívání zdarma v libovolných aplikacích. Specifikace je dostupná na adrese: <http://www.w3.org/XML/Core/#Publications>.

2.4.2 Syntaxe XML

XML využívá implicitně kódování UNICODE, ale pokud jsou uvedena, jsou přípustná i jiná kódování.

Správně strukturovaný XML dokument musí splňovat tato kritéria:

- Musí mít právě jeden kořenový (root) element.
- Neprázdné elementy musí být ohraničeny startovací a ukončovací značkou. Prázdné elementy mohou být označeny tagem „prázdný element“.
- Všechny hodnoty atributů musí být uzavřeny v uvozovkách – jednoduchých (') nebo dvojitých ("), ale jednoduchá uvozovka musí být uzavřena jednoduchou a dvojitá dvojitou. Opačný pár uvozovek může být použit uvnitř hodnot.

- Elementy mohou být vnořeny, ale nemohou se překrývat; to znamená, že každý (ne kořenový) element musí být celý obsažen v jiném elementu.

XML je tzv. Case-sensitive, čili rozlišuje velká a malá písmena. [4]

3 PŘÍNOS NOVÝCH MODULŮ A JEJICH LEGISLATIVNÍ ASPEKTY

Jako každý systém, který má sloužit k vedení podnikové agendy, tak i EkonomOS naráží na určité legislativní zákonitosti, které je nutno dodržet a často jsou i upraveny zákonem.

3.1 Modul Kniha jízd

Každý podnikatel či společnost vlastní firemní vozy zapsané do obchodního majetku musí ze zákona vést záznamy o jejich provozování – tzv. knihu jízd. Tuto povinnost ukládá zákon č. 111/1994 Sb. a vyhláška č. 187/1994 Sb. Dříve se tyto záznamy vedly obvykle v „papírové“ podobě. Zákon však nijak neomezuje způsob evidence, což vedlo k poptávce po možnosti vést záznamy o služebních jízdách elektronicky. Jako většina systému zaměřených na podnikovou agendu, tak i EkonomOS nyní nabízí vedení „elektronické“ knihy jízd. Přestože existuje řada volně dostupných programů, pro vedení záznamů o provozu služebních vozidel, určených pro použití na lokálním počítači, trendem rozsáhlejších softwarových balíčků je, obsáhnout co nejvíce potřebných nástrojů, mezi nimiž je i kniha jízd. Právě tento modul, ale může nabídnout více, než jen bezchybné, přehledné a transparentní vedení záznamů, pokud je provozován na víceuživatelském systému s možností přístupu z internetu. V kombinaci s přístupem přes mobilní zařízení (pomocí technologie WAP), může zpřístupnit také informace o okamžitém vytížení jednotlivých vozidel.

3.1.1 Povinné údaje v knize jízd

- Identifikace řidiče a vozu
- Datum jízdy včetně bližšího stanovení času
- Výchozí místo
- Cílové místo
- Stav tachometru před jízdou a po ní
- Čerpání pohonných hmot během jízdy

3.2 Modul Faktury

Pojem faktura není vymezen zákonem, přesto je v obchodním styku běžný a náležitosti faktury odpovídají ustálenému vzoru. Z hlediska účetního zákona (Zákon č. 563/1991 Sb., o účetnictví) lze na fakturu pohlížet jako na účetní doklad.

3.2.1 Povinné údaje účetního dokladu

Účetní doklady jsou průkazné účetní záznamy, které musí obsahovat:

- a) označení účetního dokladu,
- b) obsah účetního případu a jeho účastníky,
- c) peněžní částku nebo informaci o ceně za měrnou jednotku a vyjádření množství,
- d) okamžik vyhotovení účetního dokladu,
- e) okamžik uskutečnění účetního případu, není-li shodný s okamžikem podle písmene d
- f) podpisový záznam podle § 33a odst. 4 osoby odpovědné za účetní případ a podpisový záznam osoby odpovědné za jeho zaúčtování.

Faktury generované systémem EkonomOS tedy obsahují veškeré nutné náležitosti účetního dokladu. Tato automatizovaná tvorba faktur sebou nese nejen časovou úsporu, ale také minimalizuje riziko chyby ve faktuře.

II. PRAKTICKÁ ČÁST

4 INSTALÁTOR SYSTÉMU

Jednou ze slabin původního systému EkonomOS byla poměrně náročná instalace, která vyžadovala alespoň základní znalost databázového systému MySQL. Systém pro svou funkci vyžaduje zavedení databázových tabulek a některých dat do databázového systému na serveru, kde je EkonomOS provozován. Z tohoto důvodu byl vytvořen instalátor systému, který umožní snadné zprovoznění aplikace i laikovi. EkonomOS je aplikace pracující na serveru Apache s podporou PHP4 a databází MySQL, což ji předurčuje pro použití nejen na lokálním počítači, ale především na podnikové síti či internetu. Pokud bude mít uživatel zřízen webhosting s podporou požadovaných technologií, původní náročná instalace v podobě importu tabulek do databáze, se tak zjednoduší na pouhé nahrání systému EkonomOS do kořenového adresáře webhostingu (např. přes FTP) a následné spuštění vstupní adresy (např. <http://www.mujEkonomOS.cz/>). Jakmile systém zjistí, že přístup k databázi ještě není nakonfigurován, spustí automaticky instalačního průvodce. Zde uživatel vyplní jen přístupové údaje k databázovému systému běžícímu na serveru a přístupové jméno a heslo pro účet administrátora.

4.1 Průvodce instalátorem

EkonomickýSystem

Vítejte v instalačním průvodci systému EKONOMOS

Po zadání přístupových údajů k databázovému systému vytvoří instalátor databázi pojmenovanou dle Vámi zadaného jména a naimportuje do ní tabulky a data potřebná k chodu systému.

Nastavení připojení k databázi

DB server	<input type="text"/>	Jméno databázového serveru - např. localhost
DB user	<input type="text"/>	Uživatelské jméno pro připojení k databázi - např. uživatel
DB pass	<input type="text"/>	Heslo pro připojení k databázi.

Nastavení jména databáze

Jméno databáze	<input type="text"/>	Jméno databáze pro umístění tabulek systému - např. ekonomos
-----------------------	----------------------	--

Obrázek 1: Úvodní obrazovka instalátoru

Instalátor systému zkrátí původní náročnou instalaci na pouhé vyplnění několika údajů. Uživatel musí znát především přístupové údaje k databázi běžící na serveru, což je *DB server* (název serveru - např. localhost na lokálním počítači), *DB user* (jméno uživatelského účtu pro přístup k databázi), *DB pass* (heslo uživatelského účtu). Tyto údaje vyplní do formuláře na úvodní stránce spolu s názvem databáze, kterou instalátor vytvoří, a uloží do ní potřebné tabulky a data.

Ekonomický**S**ystém

Systém EKONOMOS byl úspěšně zaveden do databáze. Nyní můžete vytvořit administrátora.

Údaje o administrátorovi

Jméno	<input type="text"/>
Příjmení	<input type="text"/>

Přístupové údaje k administrátorskému účtu

Přístupové jméno	<input type="text"/>	Přihlašovací jméno pro administrátorský účet.
Přístupové heslo	<input type="text"/>	Heslo pro administrátorský účet.
Heslo znovu	<input type="text"/>	Heslo znovu pro ověření.

Obrázek 2: Instalátor - vytvoření účtu administrátora

Po úspěšném zavedení databáze uživatel ve druhém kroku zadá jméno a příjmení administrátora a jeho přístupové údaje k systému.

Ekonomický**S**ystém

Instalace systému EKONOMOS byla úspěšně dokončena.

Přístupový účet pro administrátora byl vytvořen. Nyní se již můžete přihlásit do systému pomocí přístupových údajů, které jste právě zadali.

[Spustit EKONOMOS](#)

Obrázek 3: Dokončení instalace

Po odeslání formuláře je informován o dokončení instalace a může se do systému přihlásit pomocí přístupových údajů, které právě zadal.

4.2 Princip instalátoru

Protože databáze MySQL, pro kterou je systém EkonomOS původně vytvořen, nepodporuje z bezpečnostních důvodů spuštění více dotazů najednou z externího skriptu, vyvstala otázka, jak jednoduše a efektivně docílit toho, aby PHP skript byl schopen importovat data z jediného SQL souboru, ve kterém je uložena struktura tabulek i některá data systému. Pro ilustraci – SQL soubor s tabulkami systému a jeho nezbytnou konfigurací má přes 4000 řádků a při jeho importu se provede 50 dotazů. Velkou část ovšem zabírá databáze poštovních směrovacích čísel ČR, která velmi zpříjemňuje práci při zadávání adres. Zavedení takového SQL souboru do databáze například přes phpMyAdmina není pro zkušeného uživatele nijak náročná operace, ovšem pro laika to může být neschůdný problém. Při řešení problému, jak pomocí PHP skriptu importovat velký SQL soubor do databáze, jsem si vzpomněl na publikační open source systém Mambo (<http://www.mamboserver.com/>), jenž je psán také v PHP a řeší tento problém dvěma funkcemi. První funkce (*populate_db*) otevírá SQL soubor a čte jeho obsah, který posílá funkci *split_sql*, ta jej rozdělí na jednotlivé dotazy, zbaví komentářů a odešle zpět funkci *populate_db*, která se už postará o provedení jednotlivých dotazů. Instalátor pro EkonomOS má podobné potřeby jako Mambo s tím rozdílem, že v Mambu je pro každou tabulku vytvořen samostatný SQL soubor, kdežto systém EkonomOS využívá jediného, který sdružuje všechny záznamy o tabulkách.

4.2.1 Import dat do databáze

Hlavní problém – import SQL souboru do databáze – je tedy vyřešen pomocí výše zmíněných funkcí ze systému Mambo, které jsou upraveny pro potřeby jediného SQL souboru a jsou v souboru *instdb.php*, který je volán po vyplnění přístupových údajů k databázi.

Během instalace také může dojít k situaci, kdy již existuje databáze se stejným jménem, jaké jsme zadali pro pojmenování databáze, kterou bude používat EkonomOS. Nabízí se několik možností, jak takovou situaci řešit. Je možné databázi, která již v systému je, vymazat a nahradit novou databází, jež má být nainstalována, a to buď přímo, nebo po

souhlasu uživatele. Rozhodl jsem se pro následující způsob: Systém vypíše chybové hlášení, které uživatele upozorní, že pod zadaným názvem již existuje databáze. Dále mu však nenabídne jinou možnost, než změnit jméno databáze pro EkonomOS, čímž se předejde jakýmkoliv výmazům stávajících dat v databázi. Dle mého názoru je to nejlepší řešení, protože zkušený uživatel je schopen, vymazat původní databázi i jiným způsobem. Naproti tomu, laik s možností odstranit databázi přes instalátor by mohl napáchat velké škody na stávajícím systému.

4.2.2 Vytvoření přístupového účtu administrátora

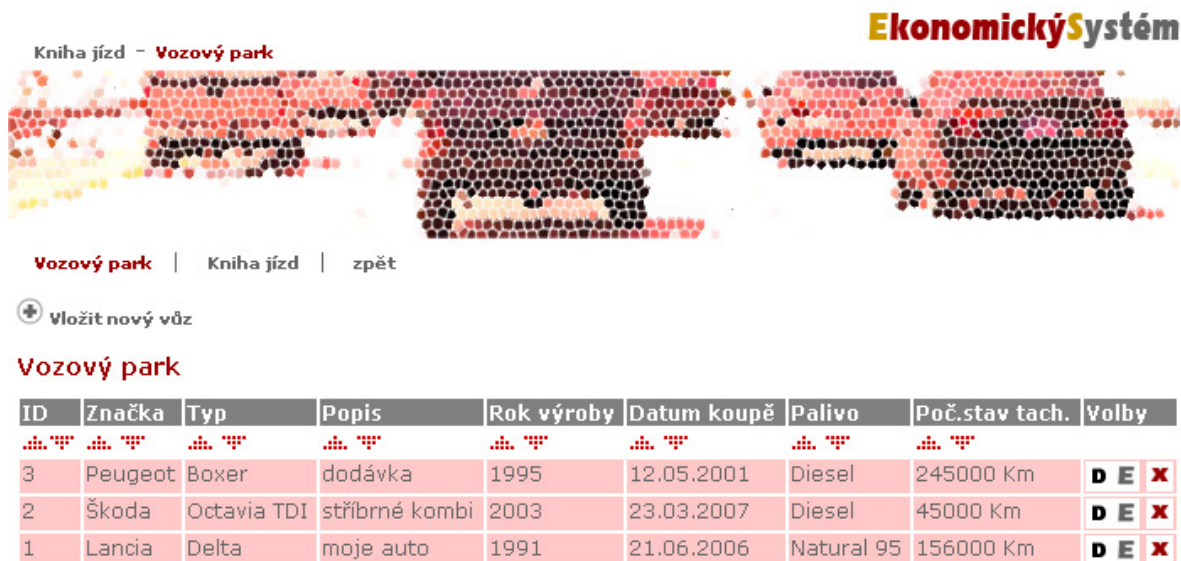
Jakmile je úspěšně vytvořena databáze pro EkonomOS, je potřeba ještě vložit uživatelský účet administrátora. Přístupové údaje, které vyplní uživatel jsou odeslány na skript *addadmin.php*, který uloží záznam o administrátorovi do příslušné tabulky *u_uzivatel*. Je uloženo id uživatele, jeho přístupové jméno a heslo. Pomocí zadaných údajů se administrátor může následně přihlásit do systému.

Připojení k databázi je v instalátoru, stejně jako v ostatních nových modulech, zprostředkováno pomocí mezivrstvy ADOdb (více viz. ADOdb – Database Abstraction Library for PHP (and Python)).

5 MODUL KNIHA JÍZD

Modul kniha jízd je určitě velmi praktickým pomocníkem v systému EkonomOS, a to především z toho důvodu, že vedení záznamů o služebních jízdách je přikázáno zákonem, který ovšem nestanovuje, jakou formou (tištěnou či elektronickou) mají být tyto záznamy vedeny. V dnešní době informačních technologií se přímo nabízí vedení „elektronické“ knihy jízd. Ta poskytuje hned několik výhod oproti „papírové“ knize jízd. Vedoucímu či majiteli firmy může nabídnout stálý a přesný přehled o provozu firemních vozidel, přímo z pohodlí jeho kanceláře. Elektronická kniha jízd minimalizuje chyby v záznamech způsobené lidským faktorem. Systém také přináší další praktické informace, jako je doba jízdy, průměrná rychlost, nebo například průměrná spotřeba vozidla. Modul Kniha jízd může být v systému EkonomOS také velmi užitečná pro jeho případný další vývoj. Pokud by se například v budoucnu realizoval modul pro generování vyúčtování pracovní cesty, záznamy z modulu Kniha jízd by mohly posloužit automatizované tvorbě „cestáku“.

5.1 Průvodce modulem Kniha jízd



Kniha jízd - **Vozový park** **Ekonomický systém**

Vozový park | Kniha jízd | zpět

+ Vložit nový vůz

ID	Značka	Typ	Popis	Rok výroby	Datum koupě	Palivo	Poč.stav tach.	Volby
3	Peugeot	Boxer	dodávka	1995	12.05.2001	Diesel	245000 Km	D E X
2	Škoda	Octavia TDI	stříbrné kombi	2003	23.03.2007	Diesel	45000 Km	D E X
1	Lancia	Delta	moje auto	1991	21.06.2006	Natural 95	156000 Km	D E X

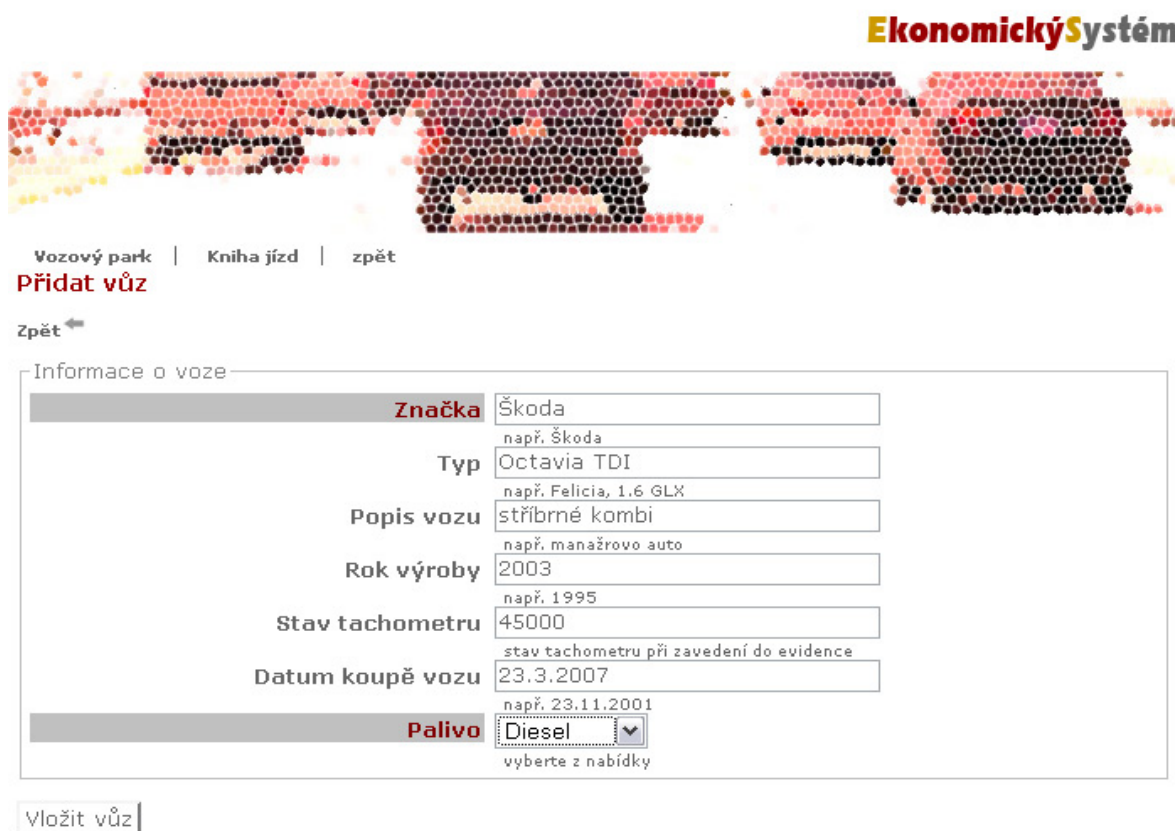
Obrázek 4: Kniha jízd – výpis služebních vozů

5.1.1 Vozový park

Část modulu pojmenovaná Vozový park shromažďuje informace o služebních vozech podniku. Umožňuje přidávat záznamy o vozech, editovat je a mazat. Tyto záznamy jsou pak navázány na samotnou knihu jízd, kde při zadávání nového záznamu dostaneme na výběr právě z vozů, které vložíme do svého vozového parku.

5.1.1.1 Vložení záznamu o voze

Před vložení prvního záznamu do knihy jízd je nejprve potřeba uložit informace o vozovém parku. Vložení služebního vozu se provede kliknutím na tlačítko „+ Vložit nový vůz“ nad výpisem vozů, který je nejprve prázdný. Záznamy o voze se zadávají do formuláře, kde povinným údajem je pouze *Značka* vozu a *Palivo*. Je však vhodné vyplnit i ostatní údaje – *Typ*, *Popis vozu*, *Rok výroby*, *Stav tachometru*, *Datum koupě vozu*. Po kliknutí na tlačítko „Vložit vůz“ je uživatel informován o provedení požadované akce a může zadat další vůz, nebo se vrátit na výpis, kliknutím na tlačítko „Zpět“.



Ekonomický systém

Vozový park | Kniha jízd | zpět

Přidat vůz

zpět ←

Informace o voze

Značka	Škoda <small>např. Škoda</small>
Typ	Octavia TDI <small>např. Felicia, 1.6 GLX</small>
Popis vozu	stříbrné kombi <small>např. manažrovo auto</small>
Rok výroby	2003 <small>např. 1995</small>
Stav tachometru	45000 <small>stav tachometru při zavedení do evidence</small>
Datum koupě vozu	23.3.2007 <small>např. 23.11.2001</small>
Palivo	Diesel <small>vyberte z nabídky</small>

Vložit vůz

Obrázek 5: Kniha jízd – vkládání záznamu o služebním voze

5.1.1.2 Výpis vozového parku

Výpis vozů obsahuje následující záznamy: *ID* – jedinečný číselný identifikátor vozu, *Značka* – značka vozu, *Typ* – model vozu, *Popis* – popis vozu pro bližší identifikaci, *Rok výroby* – rok výroby vozu, *Datum koupě* – datum zakoupení vozu, *Palivo* – druh paliva, který vůz používá, *Poč.stav tach.*- stav tachometru při koupi vozu (začátku provozu). Záznamy se vypisují podle *ID* sestupně, čili poslední přidaný záznam je na prvním místě. Pomocí šipek nahoru a dolů může uživatel třídit záznamy podle každého sloupce vzestupně či sestupně.

5.1.1.3 Editace a mazání záznamů

Jednotlivé záznamy je možno změnit kliknutím na „E“ (editovat) ve sloupci *Volby*. Smazání záznamu se provede kliknutím na červené „X“. Před provedením samotného výmazu je uživatel dotázán, zda se má požadovaná akce opravdu vykonat.

5.1.1.4 Detailní výpis údajů o voze

Detaily o voze si uživatel zobrazí kliknutím na „D“ ve sloupci *Volby*. Tento výpis zobrazí veškeré informace o vybraném voze a navíc i najeté kilometry od koupě vozu, celkový počet natankovaných litrů paliva a také průměrnou spotřebu.

[Vozový park](#) | [Kniha jízd](#) | [zpět](#)
Detailní výpis údajů o voze

[Zpět](#) ←

ID:	1
Značka:	Lancia
Typ:	Delta
Popis:	moje auto
Rok výroby:	1991
Datum koupě:	21.06.2006
Poč.stav tach.:	156000 km
Aktuální stav tach.:	156770 km
Najeto od koupě:	770 km
Celkem tankováno:	60 l (Natural 95)
Spotřeba:	7.8 l/100km

Obrázek 6: Kniha jízd – detailní výpis údajů o voze

5.1.2 Kniha jízd

Kniha jízd umožňuje vedení záznamů o služebních jízdách a obsahuje všechny položky povinně stanovené zákonem. Vedení elektronickou formou je velmi efektivním způsobem zejména pro podniky, kde je více služebních vozů. Spolu s rozhraním WAP, které umožňuje vzdálený přístup k modulu pomocí mobilního zařízení, nabízí aktuální přehled vytíženosti aut.

5.1.2.1 Vkládání záznamu o jízdě

Zápis do knihy jízd provedeme kliknutím na tlačítko „+ Vložit nový záznam do knihy jízd“. Objeví se formulář, ve kterém vybereme auto použité pro služební jízdu (musí být uvedeno ve *Vozovém parku*), zadáme časové rozmezí jízdy, počáteční a cílové místo, počáteční a koncový stav tachometru a dále pokud bylo tankováno palivo, tak kolik a jaký byl stav tachometru při tankování. Posledním údajem je účel cesty.

Protože záznamy v knize jízd by měly být konzistentní – musí sedět poslední cílové místo s počátečním místem nového záznamu a také stavy tachometru, systém tyto položky automaticky doplňuje po vybrání vozidla, podle jeho posledního záznamu v knize jízd. Automaticky doplněná data, se však v případě potřeby dají přepsat.

Povinné údaje jsou zvýrazněny vínovým tučným textem a šedým pozadím.

Kniha jízd - **Kniha jízd**

Ekonomický systém

Vozový park | **Kniha jízd** | zpět

Vložení záznamu o jízdě

Zpět ←

Informace o voze

Vyberte vůz	<input type="text"/>
Od	<input type="text" value="11.05.2007 08:57"/>
Do	<input type="text" value="11.05.2007 09:57"/>
Počáteční místo	<input type="text"/>
Cílové místo	<input type="text"/>
Počáteční stav tachometru	<input type="text" value="např. 134570"/>
Koncový stav tachometru	<input type="text" value="např. 134600"/>
Tankováno litrů	<input type="text"/>
Stav tachometru při tankování	<input type="text"/>
Účel cesty	<input type="text" value="Proč bylo vozidlo použito."/>

Uložit údaje

Obrázek 7: Kniha jízd - vkládání záznamu o služební jízdě

5.1.2.2 Výpis záznamů v knize jízd

Výpis záznamů obsahuje následující údaje: *ID* – jedinečný číselný identifikátor záznamu, *Auto* – značka použitého vozu (pro bližší identifikaci se po najetí kurzorem nad značku zobrazí další informace o voze), *Řidič* – příjmení řidiče (vlastníka účtu, který provedl záznam), *Začátek* – čas počátku jízdy, *Konec* – čas konce jízdy, *Počátek* – místo počátku

jízdy, *Cíl* – cílové místo, *Poč. stav tach.* – stav tachometru před jízdou, *Konc. stav tach.* – stav tachometru po jízdě, *Účel jízdy* – důvod nebo účel služební jízdy.

Záznamy se vypisují, stejně jako záznamy *Vozového parku*, podle *ID* sestupně, čili poslední přidaný záznam je na prvním místě. Pomocí šipek nahoru a dolů může uživatel třídit záznamy podle každého sloupce vzestupně či sestupně.

Ekonomický systém

Kniha jízd - **Kniha jízd**

Vozový park | **Kniha jízd** | zpět

+ Vložit nový záznam do knihy jízd

Kniha jízd - výpis jízd

ID	Auto	Řidič	Začátek	Konec	Počátek	Cíl	Poč. stav tach.	Konc. stav tach.	Účel jízdy	Volby
4	Lancia	Vala	10.05.2007 16:00	10.05.2007 18:00	Zlín	Bruntál	156770	156910	Studium	D E X
3	Lancia	Vala	08.05.2007 10:30	08.05.2007 12:15	Bruntál	Zlín	156640	156770	Konzultace k BP	D E X
2	Lancia	Vala	03.05.2007 15:30	03.05.2007 20:30	Liberec	Bruntál	156320	156640	Reprezentace UTB - akademické hry	D E X
1	Lancia	Vala	01.05.2007 11:30	01.05.2007 16:30	Bruntál	Liberec	156000	156320	Reprezentace UTB - akademické hry	D E X

Obrázek 8: Kniha jízd - výpis záznamů

5.1.2.3 Editace a mazání záznamů

Editace a mazání probíhá opět stejným způsobem jako ve výpisu *Vozového parku*.

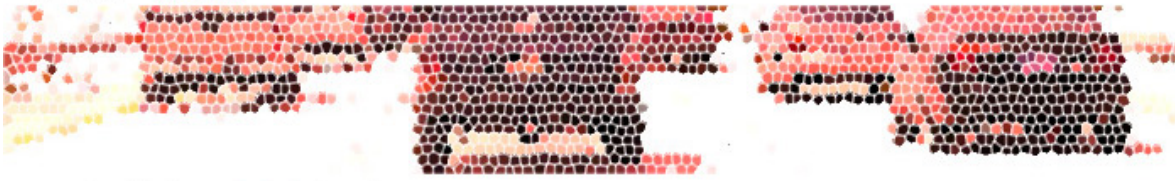
Jednotlivé záznamy je možno změnit kliknutím na „E“ (editovat) ve sloupci *Volby*. Smazání záznamu se provede kliknutím na červené „X“. Před provedením samotného výmazu je uživatel dotázán, zda se má požadovaná akce opravdu vykonat.

5.1.2.4 Detailní výpis cesty

Pod tlačítkem „D“ se skrývá detailní výpis informací o služební jízdě. Nahoře v hlavičce je uveden čas jízdy, délka jízdy v kilometrech, a průměrná rychlost jízdy. Níže jsou uvedeny ostatní záznamy, které uživatel vložil do databáze při zadávání záznamu o služební jízdě.

EkonomickýSystem

Kniha jízd - **Kniha jízd**



Vozový park | **Kniha jízd** | zpět

Detail záznamu z knihy jízd

Zpět ←

Čas jízdy: 5 hod 00 min | Najeto: 320 km | Průměrná rychlost: 64 km/h

ID:	1
Auto:	Lancia Delta (moje auto)
Řidič:	Vala Radek
Účel jízdy:	Reprezentace UTB - akademické hry
Začátek:	01.05.2007 11:30
Konec:	01.05.2007 16:30
Počátek:	Bruntál
Cíl:	Liberec
Poč. stav tach.:	156000 km
Konc. stav tach.:	156320 km
Tank. na kilometru:	156200 km
Tank.:	25 l

Obrázek 9: Kniha jízd - detailní výpis služební jízdy

5.1.3 WAP rozhraní Knihy jízd

WAP rozhraní je přístupné z mobilního telefonu, pokud je systém nainstalován na webovém serveru z adresy – http://www.nazevserveru.cz/kniha_jezd/wap. Uživatel nejprve musí zadat přístupové údaje, které používá při klasickém přihlašování.

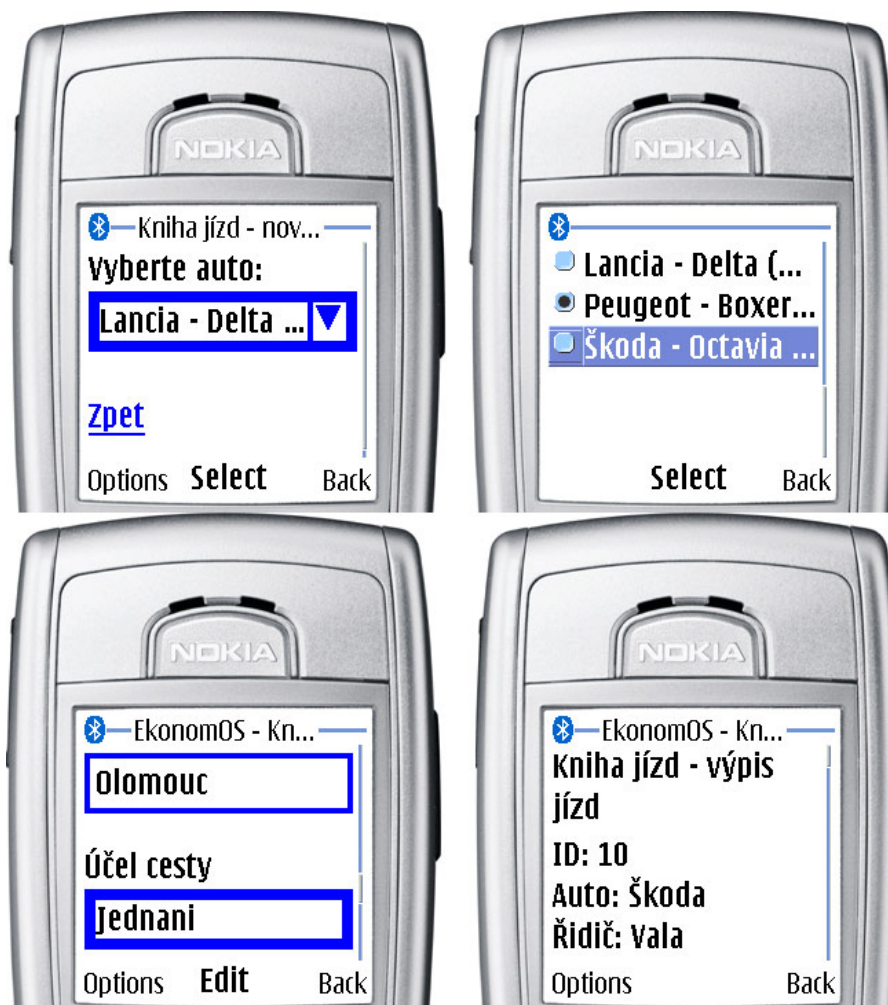
Po přihlášení se mu zobrazí nabídka – *Vložit záznam* (vkládání záznamu o služební jízdě), *Zobrazit záznamy* (zobrazení záznamu o služebních jízdách) a *Odhlasit se* (ukončení práce s modulem Kniha jízd).



Obrázek 10: WAP - Modul kniha jízd - úvodní menu (WAP emulátor NOKIA S40 DP 2.0 SDK)

5.1.3.1 Vkládání záznamů

Záznamy se vkládají pod tlačítkem *Vložit záznam*. Nejprve je nutné vybrat vůz z nabídky vozového parku. Některá další formulářová pole se automaticky předvyplní v závislosti na vybraném voze. Např. *Poč.stav.tach* (počáteční stav tachometru) a *Počátek* (počáteční místo by mělo být shodné s koncovým místem poslední jízdy). Také se vyplní pole s počátečním a koncovým datem a časem jízdy, pro snazší přepis těchto hodnot a urychlení práce. Význam formulářových polí je totožný s klasickým formulářem v modulu *Knihy jízd*. Po vyplnění formuláře je záznam uložen a uživatel je přesměrován na úvodní menu.



Obrázek 11: WAP - Modul kniha jízd – vybrání vozu z vozového parku, vkládání údajů do formuláře pro vložení záznamu do knihy jízd a výpis údajů z knihy jízd

5.1.3.2 Výpis údajů z knihy jízd

Výpis údajů přes WAP provedeme otevřením odkazu *Zobrazit záznamy*. Tím se nám vypíší jednotlivé údaje z *Knihy jízd* v jednoduché textové formě – vždy popis položky a její hodnota. Záznamy jsou řazeny chronologicky od nejnovějšího po nejstarší.

5.2 Důležité třídy a funkce

5.2.1 Třídy v souboru class.php

Nezbytnou součástí modulu je soubor class.php. Přestože podle pravidel Zend Framework (<http://framework.zend.com/manual/en/coding-standard.html>), která jsou určena pro psaní PHP kódu, by každá jednotlivá třída měla být v samostatném souboru, soubor class.php sdružuje dvě třídy používané v modulu. Je to z toho důvodu, aby adresář modulu obsahoval co nejméně souborů a tyto se podle architektury původních modulů nacházely hned v jeho rootu.

5.2.1.1 Class Database

Tato třída obsahuje funkce pro práci s databází pomocí mezivrstvy ADOdb.

Obsahuje funkce **pripojDB()**, **smazZaznam(\$from,\$where,\$where_hodnota,\$db)** a **odpojDB(\$db)**.

- **pripojDB()** – funkce bez parametrů, vytvoří spojení přes vrstvu ADOdb a vrátí objekt \$db pro obsluhu tohoto připojení.
- **smazZaznam(\$from,\$where,\$where_hodnota,\$db)** – tato funkce je volána z některých částí modulu, slouží k mazání záznamu z databáze. Parametrem je název tabulky, ze které se záznam bude mazat, název sloupce v podmínce WHERE SQL dotazu, hodnota sloupce podmínky WHERE a ukazatel na ADOdb objekt.
- **odpojDB(\$db)** – funkce, která se volá pro ukončení spojení s databází, parametrem je ukazatel na objekt připojení ADOdb.

5.2.1.2 Class Vuz

Třída obsahuje funkce vkládání a zpětného výpisu záznamů v modulu *Knihajízda*.

- **vlozVuz(\$popis,\$znacka,\$typ,\$srv,\$palivo,\$startkm,\$skoupeno,\$db)** – funkce pro vložení záznamu o voze do databáze vozového parku. Parametry jsou výstupy z formuláře pro vložení nového vozu a ukazatel na objekt připojení ADOdb. Návratem je TRUE či FALSE v závislosti na úspěšnosti provedení SQL dotazu.
- **nactiHodnotyAuta(\$id,\$db)** – tato funkce má parametry id vozu a ukazatel na objekt připojení ADOdb. Načítá data do těchto proměnných třídy Vuz: *\$znacka*

(značka auta), *\$najeto* (najeto kilometrů), *\$misto* (místo, kde se vůz nachází), *\$tankovano* (celkový počet litrů paliva, které bylo tankováno za dobu používání).

- **vratSelectAuta(\$db)** – funkce pouze provede SQL dotaz, který vybere z databáze všechna potřebná data pro výpis vozového parku. Parametrem je ukazatel na objekt připojení ADOdb. Návratem je ADOdb objekt *\$zaznam*, který obsahuje vybraná data.
- **vratAuto(\$id,\$db)** – tato funkce funguje stejně jako výše uvedená funkce **vratSelectAuta(\$db)**, s tím rozdílem, že vrací pouze data o konkrétním vozidle, potřebná pro některé výstupy v modulu. Přibyl tedy parametr *\$id*, podle něž je z databáze vybráno konkrétní vozidlo. Návratem je opět ADOdb objekt *\$zaznam*, který obsahuje vybraná data.
- **editujHodnoty(\$ida,\$popis,\$znacka,\$styp,\$srv,\$palivo,\$startkm,\$skoupeno,\$db)** – funkce pro editaci záznamu v databázi o konkrétním voze. Parametry jsou výstupy z editačního formuláře a ukazatel na objekt připojení ADOdb. Funkce provede SQL UPDATE dotaz s patřičnými hodnotami, parametr *\$ida* udává, nad kterým řádkem databáze se dotaz provede. Funkce vrací TRUE či FALSE v závislosti na úspěchu provedení SQL dotazu.
- **pridejJizdu(\$id_auta,\$id_uzivatel,\$od,\$do,\$start,\$stop,\$km1,\$km2,\$stank,\$stankkm,\$sucel,\$db)** – funkce pro přidání záznamu o služební jízdě do tabulky *kjizd_jizdy*. Parametry jsou výstupy z formuláře pro přidávání záznamů do knihy jízd a ukazatel na objekt připojení ADOdb. Funkce vrací TRUE či FALSE v závislosti na úspěchu provedení SQL dotazu.
- **vratHodnotyJizda(\$idk,\$db)** – funkce pro vypsání všech záznamů o služební jízdě z databáze. Parametrem je identifikátor řádku v databázi *\$idk* a ukazatel na objekt připojení ADOdb. Návratem je objekt *\$zaznam* obsahující výstup z databáze.
- **editujJizdu(\$id_auta,\$idk,\$od,\$do,\$start,\$stop,\$km1,\$km2,\$stank,\$stankkm,\$sucel,\$db)** – funkce pro editaci záznamu o služební jízdě v databázi. Parametry jsou výstupy z editačního formuláře a ukazatel na objekt připojení ADOdb.

5.2.2 Funkce v souboru funkce.php

Soubor funkce.php je zachován z původních modulů systému EkonomOS a obsahuje mimo nové funkce také původní funkce potřebné ke stanovení přístupových práv uživatele (více informací o původních funkcích v bakalářských pracích T. Zimáčka a D. Malanika z roku 2006). V následujícím textu budou popsány pouze nově přidané funkce.

5.2.2.1 Funkce *orderBy(\$sort,\$posl)*

Tato funkce slouží k třídění dat z databáze podle zadaného sloupce vzestupně či sestupně. Parametr *\$sort* obsahuje jméno sloupce, jenž bude kritériem pro třídění, a parametr *\$posl* obsahuje číslo 1 či 0 pro vzestupné, či sestupné třídění. Funkce je volána přímo z těla SQL dotazu, kde by měla být vložena podmínka ORDER BY (třídít podle) a jejím návratem je řetězec podmínky.

5.2.2.2 Funkce *vratPalivo(\$cislo)*

Tato pomocná funkce se volá, pokud je potřeba zobrazit, jaké palivo používá vybraný automobil. Druh paliva je v databázi totiž uložen číselně (1 – Natural 95, 2 – Natural 98, 3 – Diesel, 4 – Speciál, 5 – LPG). Funkce je tedy volána s parametrem, který odpovídá číslu paliva, a pomocí semaforu *switch* vrací řetězec s názvem odpovídajícího druhu paliva.

6 MODUL FAKTURY

Faktury jsou nezbytnou součástí agendy obchodních firem. Tyto účetní doklady je potřeba vytvářet, tisknout a archivovat. „Ruční“ tvorba faktur je v dnešní době naprosto neefektivní, proto existuje mnoho specializovaného softwaru, který tuto činnost značně zjednodušuje. Programy pro samotnou tvorbu faktur bývají často zdarma ke stažení na internetu, případně můžete využít přímo on-line tvorby faktury (např. <http://www.faktura.cz>).

Tyto specializované nástroje ovšem narážejí na jeden podstatný problém. Veškeré údaje je nutno do faktury zadat ručně, což značně zpomaluje práci.

V systému EkonomOS s modulem *Sklady* (skladové zásoby) a *Adresy* (kontakty na osoby či firmy) se přímo nabízí propojení zmíněných modulů s modulem *Faktury*. Faktura totiž obsahuje údaje o dodavateli, odběrateli a fakturovaných položkách. Firmy obvykle vedou záznamy o kontaktech na své obchodní partnery v modulu *Adresy*. Toho právě využívá modul *Faktury* a data o odběrateli i dodavateli čerpá právě z kontaktů v modulu *Adresy*. Jednotlivé položky faktury jsou vybírány z položek vedených v modulu *Sklady*. Předmětem faktury mohou být i služby. Typy služeb pro fakturaci lze vkládat v modulu *Faktury*. Tento způsob tvorby faktur je velmi rychlý a pohodlný a minimalizuje vznik jakékoliv chyby na účetním dokladu.

6.1 Průvodce modulem Faktury

FAKTURA - DAŇOVÝ DOKLAD č. 300005						
Dodavatel: CykloŠvec Stránského 12 76501 Otrokovice 1 IČ: 99887766 DIČ: CZ99887766 Číslo účtu: 353936570207/0100			Odběratel: IČ: 43642443 DIČ: CZ4553455 Miloslav Hruška - cykloservis Okružní 3 79201 Bruntál 1			
			Variabilní symbol: 300005 Konstantní symbol: 308			
Forma platby: převodem na účet						
Datum vystavení: 19.05.2007						
Datum splatnosti: 19.06.2007						
Datum uskutečnění plnění: 19.05.2007						
Označení dodávky:	Množství	Cena/ks bez DPH	Cena celkem	%DPH	DPH	Kč celkem
Fakturuje Vám zboží dle objednávky:						
Výměna řídítek	1	50.0	50.0	5%	2.5	52.5
KHS ALITE 4000	1	21000.0	21000.0	19%	3990.0	24990.0
Součet položek:			21050.0		3992.5	25042.5
CELKEM K ÚHRADĚ VČETNĚ DPH:						25043
Převzal:			Razítko:			
CykloŠvec - ičo: 99887766 - dič: CZ99887766						

Obrázek 12: Příklad faktury generované modulem Faktury v systému EkonomOS

6.1.1 Fakturované služby

Modul fakturace se snaží o maximální urychlení tvorby faktur a proto se předmět faktury zadává ze seznamu zboží (modul *Sklady*), nebo služeb. Tento seznam služeb je dostupný právě pod odkazem *Fakturované služby*.

6.1.1.1 Přidání nového typu služby

Chceme-li fakturovat nějakou službu, například tvorbu grafického návrhu webových stránek, musíme tuto službu nejprve vložit do seznamu *Fakturovaných služeb*.

Kliknutím na *Přidat službu* zobrazíme formulář pro zadání *popisu* služby (bude zobrazen jako předmět faktury) a *ceny bez DPH*. Službu přidáme kliknutím na *Přidat službu*.

EkonomickýSystém

Faktury - **Fakturované služby**

Vytvořit novou fakturu | Výpis faktur | **Fakturované služby** | zpět

Vložení nové služby

zpět na výpis služeb ←

Fakturovaná služba

Služby	grafický návrh www stránek
<small>Popis služby bude uveden v předmětu faktury.</small>	
Cena bez DPH	2500 Kč bez DPH
<small>Cena za službu bez DPH.</small>	

Přidat službu

Obrázek 13: Modul Faktury - přidání položky do seznamu fakturovaných služeb

6.1.1.2 Zobrazení seznamu fakturovaných služeb

Seznam fakturovaných služeb, se zobrazí ihned po kliknutí na tlačítko *Fakturované služby*, pokud obsahuje nějaké záznamy. Zobrazí *ID* záznamu, *popis* služby, který bude uváděn v předmětu faktury, a *cenu bez DPH*, jež bude za jednotlivou službu účtována. *Volby* výpisu nabízejí editaci záznamu (pod tlačítkem *E*) a mazání záznamu (pod tlačítkem *X*). Řazení záznamů je možné podle všech sloupců jak vzestupně, tak sestupně, pomocí šipek dolů a nahoru.

Vytvořit novou fakturu | Výpis faktur | **Fakturované služby** | zpět

➕ Přidat službu

Seznam fakturovaných služeb

ID	Popis	Cena bez DPH	Volby
⌵ ⌴	⌵ ⌴	⌵ ⌴	
5	Designový návrh stránek - 5 hod	2500	E X
4	Programování 1hod	300	E X
3	Tvorba www stránek	12000	E X
2	Instalace a konfigurace LAN kamery	500	E X
1	Přepis textů - 4 hod	300	E X

Obrázek 14: Modul Faktury - seznam fakturovaných služeb

6.1.2 Faktury

Samotná tvorba faktur v modulu *Faktury* je velmi rychlý a jednoduchý proces, za předpokladu, že průběžně zavádíme do systému EkonomOS data o firemních kontaktech, skladových zásobách a nabízených službách. Vycházel jsem z předpokladu, že na fakturách se velmi často opakují položky *odběratel*, *dodavatel* a *předmět* faktury. Předmět či službu nefakturujeme zpravidla jednou, nýbrž opakovaně, a proto se vyplatí tyto údaje zavést do systému EkonomOS a později je na fakturu pouze „naklikat“, čímž odpadá opisování adres a údajů.

6.1.2.1 Tvorba faktury

[Vytvořit novou fakturu](#) | [Výpis faktur](#) | [Fakturované služby](#) | [zpět](#)

Vytvořit novou fakturu

Fakturační údaje

Druh faktury - přijatá/vydaná faktura

Faktura Vydaná Vyberte zda se jedná o fakturu přijatou či vydanou.

Číselník

Číselník VF | Vystavené faktury Vyberte číselník ze seznamu.

Odběratel

Odběratel 3, CykloŠvec, Stránského, 12, 99887766, CZ99887766, Otrokovice 1. Vyberte odběratele ze seznamu firem.

Dodavatel

Dodavatel 2, Miloslav Hruška - cykloservis, Okružní, 3, 43642443, CZ4553455, Bruntál 1. Vyberte dodavatele ze seznamu firem.

Datum vystavení

Datum vystavení 19.05.2007 Datum vystavení faktury.

Datum splatnosti 19.06.2007 Datum splatnosti faktury.

Datum zdaň. plnění 19.05.2007 Datum zdanitelného plnění.

Obrázek 15: Modul Faktury – tvorba faktury – zadání fakturačních údajů

Tvorbu faktury zahájíme kliknutím na *Vytvořit novou fakturu*. Nejprve vybereme, zda jde o fakturu vydanou či přijatou a dále zadáme, podle jakého číselníku má být faktura

číslována. Výběr číselníku je propojen s modulem *Číselníky*, a je tedy nutné, aby tento modul obsahoval záznamy o číselnících. Data o odběrateli a dodavateli jsou čerpána z modulu *Adresy*. Pokud tedy např. fakturu vydáváme, vložíme své fakturační údaje do modulu *Adresy* a vlastní profil pak zadáme jako profil dodavatele (Pozn. modul *Faktury* zobrazuje pouze fakturační adresy jednotlivých subjektů.). V nabídce *odběratel* či *dodavatel* jsou zobrazeny informace o subjektu oddělené čárkou v následujícím pořadí: *Název firmy, ulice, č.p., IČO, DIČ, město*. Dále je možno zadat datum vystavení, splatnosti a zdanitelného plnění. Pro ulehčení zápisu je automaticky vloženo dnešní datum a datum splatnosti je za 1 kalendářní měsíc od data vystavení. Údaje o datech lze přepsat.

Stisknutím tlačítka *Pokračuj* přejdeme na zadávání předmětu faktury. Předmětem může být jak zboží, tak služba.

[Vytvořit novou fakturu](#) | [Výpis faktur](#) | [Fakturované služby](#) | [zpět](#)

Vytvořit novou fakturu

Předmět faktury

ID položky	Název	Cena bez DPH	Kusů	DPH	
ID: Z1	KHS ALITE 4000	21000	3	19	X
ID: Z2	KHS ALITE TEAM	78000	1	19	X

Fakturační údaje

Předmět faktury - zboží

Předmět faktury - zboží 1, KHS ALITE 4000, 21000 Kč bez DPH

Vyberte fakturovanou položku ze seznamu skladových zásob.

Předmět faktury - služba není vybráno

Vyberte službu, jako předmět faktury. Služby lze přidávat a měnit pod záložkou Fakturované služby.

Počet kusů 1

Vložte počet kusů fakturované položky.

DPH 19 %

Zadejte daň z přidané hodnoty.

Způsob úhrady faktury

Forma úhrady Hotově

Vyberte způsob uhrazení faktury.

Sleva z celkové ceny %

Zadejte slevu - pouze číslo (v procentech).

Obrázek 16: Modul Faktury – tvorba faktury – vkládání předmětu faktury

Vždy vybíráme buď zboží (data z modulu *Sklady*), nebo službu (data z modulu *Faktury – Fakturované služby*) a zadáváme počet kusů a DPH. Přidání vybrané položky do předmětu faktury provedeme kliknutím na „+“.

Tabulka *Předmět faktury* zobrazuje námi zadaný předmět faktury. Jednotlivé položky lze odstranit tlačítkem „X“.

Jakmile máme zadány všechny položky, které mají být uvedeny v předmětu faktury, vybereme způsob úhrady. Pokud zadáme volbu převodem z účtu, zobrazí se nám pole pro zdání čísla bankovního účtu a konstantního symbolu.

Také můžeme zadat případnou slevu z celkové ceny bez DPH.

Pro vygenerování faktury klikneme na tlačítko *Pokračuj*. Systém zobrazí hotovou fakturu, kterou můžeme ihned vytisknout. Takto vytvořené faktury můžeme později najít ve *Výpisu faktur*.

6.1.2.2 Výpis faktur

Seznam všech faktur vytvořených v systému EkonomOS lze najít pod odkazem *Výpis faktur*. Záznamy jsou řazeny podle čísla faktury sestupně. Také je možno zobrazit pouze přijaté nebo vydané faktury. Ve výpisu jsou zobrazovány tyto údaje: *Číslo faktury* (podle zadaného číselníku), *Vystaveno* (datum a čas vytvoření faktury), *Splatnost* (datum splatnosti faktury), *Vystavil* (přihlašovací jméno uživatele, který fakturu vytvořil), *Celková cena* (cena vystavené faktury).

Výpis faktur nabízí kromě zobrazení hlavních údajů také signalizaci data splatnosti.

U faktur také můžeme potvrdit např. odeslání či přijetí fakturované částky tím, že klikneme na přeškrtnutý symbol . Touto akcí zakážeme signalizaci termínu splatnosti u faktury a nepřeškrtnutý symbol nám bude ukazovat, že platba byla provedena/přijata. Najedeme-li kurzorem nad symbol objeví se kdo a kdy fakturu „potvrdil“.

6.2 Důležité třídy a funkce

6.2.1 Třídy v souboru class.php

6.2.1.1 Class Databaze

Popis této třídy je uveden v kapitole 5.2.1.1

6.2.1.2 Class Faktura

Třída slouží k uchování a přenosu dat mezi jednotlivými kroky vytváření faktury a obsahuje funkce pro vkládání a výpis dat z databáze a samotné generování faktury.

- **novaFaktura(Sid_odberatel, Sid_dodavatel, \$datum, \$datumspla, \$datumdan, \$druh, \$ciselnik)** – tato funkce ukládá data z výstupu formuláře pro zadávání faktur do proměnných třídy a tím umožňuje tyto údaje přenášet v serializovaném objektu mezi jednotlivými stránkami formuláře pro zadávání faktury.
- **nactiOdberatele(\$db)** – tato funkce jejímž parametrem je ukazatel na objekt připojení ADOdb načte z databáze fakturační adresy z modulu *Adresy* do objektu *\$odberatel*, který je po návratu využit pro sestavení nabídky kontaktů, které můžeme uvést jako odběratele či dodavatele.
- **vratCisloFaktury(\$db)** – tato funkce, jejímž parametrem je pouze ukazatel na objekt připojení ADOdb, vrací číslo faktury. To se odvíjí od počáteční hodnoty použitého číselníku a od toho, zda již nějaká faktura pro daný číselník existuje. Pokud neexistuje, použije první možné číslo číselníku a pokud existuje, použije číslo poslední faktury na daném číselníku zvýšené o jedničku.
- **vratFakturu(\$idf,\$session,\$db)** – je nejdůležitější funkcí modulu *Faktury*. Pokud chceme generovat výstup faktury na obrazovku, je volána právě tato funkce. Parametr *\$idf* udává číslo faktury, jež se má vygenerovat. Parametr *\$session* obsahuje 1 či 0 a udává, zda se mají data faktury brát z proměnných objektu faktura (v případě, že faktura ještě nebyla uložena do databáze – prvotní generování), nebo ze záznamů v databázi (druhotné generování – zobrazení již vytvořených faktur z *Výpisu faktur*). Parametr *\$db* je ukazatel na objekt připojení ADOdb, aby se

k databázi mohlo přistupovat i přímo z funkce. Návratovou hodnotou funkce je tedy HTML řetězec s naformátovanou fakturou.

7 EXPORT A IMPORT DAT

Jedním z nejběžněji používaných formátů pro přenos dat je formát CSV. Stal se velmi oblíbeným pro svou jednoduchost a především kompatibilitu s tabulkovým programem MS Excel. Proto jsem vytvořil modul *Export CSV* a *Import CSV*, který umožňuje exportovat a importovat některá data právě do formátu CSV.

Pro modul *Faktury* jsem připravil export do formátů XML a HTML.

7.1 Průvodce modulem Export CSV



Obrázek 17: Export dat - modul Export CSV - výběr tabulky a kódování

Exportování dat je velice jednoduché a rychlé. Vybereme, z kterého modulu chceme data exportovat, následně tabulku dat, kterou chceme exportovat a požadované kódování. Po kliknutí na tlačítko Export se zobrazí dialogové okno pro stažení souboru. Exportovaná data tedy můžeme buď uložit na disk, nebo přímo otevřít např. v Excelu.

7.2 Princip exportu dat do CSV

7.2.1 Problém s relacemi

Přestože export do CSV se může jevit jako banální záležitost, s provázaností dat v systému, složitost řešení tohoto problému narůstá. Bylo by samozřejmě možné nabídnout uživateli výpis všech tabulek systému generovaný jedním SQL dotazem, z nichž by vybral tu, kterou chce exportovat, a další část skriptu by jednoduše provedla export všech hodnot

z databázové tabulky. Toto řešení by bylo sice velmi efektivní pro programátora, který by si ušetřil práci, a vše by fungovalo pomocí několika řádků skriptu, ale běžnému uživateli by nepřineslo takřka nic. Zmiňovaná provázanost záznamů (relace mezi tabulkami) je klíčová pro výpisy smysluplných dat. Při výpisu jednotlivých tabulek by uživatel dostal dílčí data, např. číselné identifikátory, které slouží k propojení jednotlivých záznamů, a aby se dopracoval k hodnotnému výstupu, musel by tato data opět pospojovat relacemi, jež se používají v systému EkonomOS. Toto řešení by tedy rozhodně nebylo přínosem, proto jsem se rozhodl řešit problém pro data z každého modulu zvlášť.

7.2.2 Řešení - export pro každý modul zvlášť a jeho výhody

Řešení spočívá v tom, že jsem pro každý modul sestavil možnosti exportu (např. modul Kniha jízd – tabulka s výpisem vozového parku a tabulka s výpisem knihy jízd) a především samostatný SQL dotaz, který již spojuje tabulky relacemi a dává na výstup zpracovaná a hodnotná data.

Každý modul by tedy měl mít vlastní soubor pro export dat (např. modul Kniha jízd - soubor `e_kjizd.php`), který nabídne uživateli formulář s výběrem tabulky k exportu a kódováním dat. Tabulka ve výběru však není skutečnou tabulkou z databáze, ale fiktivní tabulkou, která vzniká často propojením několika skutečných databázových tabulek. Uživatel tedy nemusí znát strukturu databáze, což je pozitivní i z hlediska bezpečnosti systému, ale především odpadá nutnost řešit relační propojení exportovaných dat.

Název vybrané tabulky a požadované kódování je odesláno na soubor `export.php`, který obsluhuje exporty všech modulů. `Export.php` nejprve nastaví pro komunikaci s databází požadované kódování a poté pomocí semaforu `switch` vybere SQL dotaz, který vyhledá data z databáze, jež uživatel požaduje. V cyklu `while` jsou jednotlivé záznamy naformátovány podle specifikace CSV, kterou používá MS Excel (záznamy oddělené středníkem).

Nakonec je hlavičkami odeslán požadavek na stažení souboru a vygenerovaný výstup, který má MIME typ `text/csv`, je možno uložit na disk.

Toto řešení je mimo jiné vhodné i pro další rozvoj systému, jež si vyžádá i export dat z modulů, které teprve možná vzniknou. V tom případě stačí pouze vytvořit nový soubor, který nabídne uživateli výběr tabulky a kódování a dále rozšířit soubor `export.php` o další SQL dotaz.

7.3 Průvodce modulem Import CSV

V modulu *Import CSV* je možno importovat data pomocí CSV souboru. Prozatím je připraven import dat pro modul *Kniha jízd*. Uživatel může vybrat, zda importovat do tabulky *Vozový park*, nebo tabulky *Kniha jízd*. Import se provede vybráním tabulky, souboru CSV a použitého kódování.

Obrázek 18: Modul Import CSV - formulář pro výběr importovaného souboru

7.3.1 Struktura importovaného CSV

Soubor pro import musí mít pevně danou strukturu, která se liší pro každou tabulku. Struktura CSV pro import do tabulky *Kniha jízd* vypadá následovně: identifikační číslo řidiče; identifikační číslo vozu; datum začátku jízdy ve formátu RRRR-MM-DD HH:MM; datum konce jízdy ve formátu RRRR-MM-DD HH:MM; počáteční místo jízdy; koncové místo jízdy; počáteční stav tachometru; koncový stav tachometru; tankováno litrů; stav tachometru při tankování; účel jízdy.

Struktura CSV pro import do tabulky *Vozový park* vypadá následovně: popis vozu; značka vozu; typ vozu; rok výroby; palivo (označené číselným identifikátorem: 1 – Natural 95, 2 – Natural 98, 3 – Diesel, 4 – Speciál, 5 – LPG); počáteční stav tachometru; datum koupě.

7.4 Princip importu dat z formátu CSV

Data z formátu CSV musí mít pevně danou strukturu (počet záznamů na řádku, jejich význam a pořadí), protože jsou parsována a vkládána do pevně dané struktury databáze.

Skript pro import – *import.php* – otevře soubor, který je vybrán pomocí formuláře pro import, přečte ho, rozdělí na jednotlivé záznamy a uloží data do pole. Poté vloží přečtená data do databáze.

7.5 Průvodce modulem Export XML/HTML

Tento modul nabízí export Faktur do XML a HTML formátu. Obsluha je velmi jednoduchá, stačí vybrat číslo faktury, kterou chceme exportovat, dále kódování a formát – XML, nebo HTML. Po odeslání požadavku se objeví dialogové okno s možností uložení souboru na disk.

7.6 Princip exportu dat do XML/HTML

7.6.1 Export faktur do XML

Výstup zajišťuje funkce **vratFakturuXML(\$idf,\$skodovani,\$db)** s parametry: *\$idf* – číslo faktury, *\$skodovani* – požadované kódování výstupního dokumentu, *\$db* – ukazatel na objekt připojení ADOdb. Uvnitř funkce se provedou SQL dotazy, které vracejí data faktury, a ty jsou zasazeny do XML formátu výstupu.

Výstup je odeslán pomocí hlaviček (*header("Content-Type: application/force-download")*) prohlížeči s požadavkem na stažení souboru.

Formát XML souboru vychází ze schématu, které používají velmi rozšířené faktury systému POHODA (<http://www.stormware.cz/>). To umožňuje pohodlný import do systémů, používajících této formát.

7.6.2 Export faktur do HTML

Výstup zajišťuje funkce **vratFakturu(\$idf,\$skodovani,\$db)** s parametry viz. 7.6. Funkce vrací řetězec s kompletním HTML kódem faktury. Výstup je odeslán stejně jako výstup s XML formátem pomocí funkce *header* s požadavkem na stažení souboru.

8 TISK V SYSTÉMU EKONOMOS

Novinkou oproti původní verzi systému EkonomOS je velmi praktický tisk dat ze systému. Princip spočívá v tom, že designové prvky systému, jako například logo, grafika v hlavičce, údaje o přihlášeném uživateli atd., jsou zakázány pro tiskový výstup přímo v souboru kaskádového stylu. Uživateli tak stačí v kterémkoliv modulu u výpisu dat pouze stisknout tlačítko tisk v prohlížeči a vytisknou se pouze tabulky výpisu bez okolních prvků stránky.

8.1 Princip skrytí prvků na tiskovém výstupu

Data pro tisk nejsou nijak upravována. Skrytí nepotřebných prvků pro tisk obstarává přímo soubor *style.css* v adresáři *inc*. Proti původnímu jsem v něm rozlišil stylování prvků pro výstup na obrazovku a pro tisk. Toto rozlišení je provedeno pomocí zápisů:

```
@media screen
{
    /* obsahuje styly výstupu na obrazovku*/
}

@media print
{
    /* obsahuje styly pro tisk*/
}
```


ZÁVĚR

Podle mého názoru největší přínos mé práce pro systém EkonomOS je zjednodušení instalace, což EkonomOS může přiblížit i běžnému uživateli. Proto jsem vytvořil instalační modul, jenž vytvoří kompletní databázi potřebnou pro systém EkonomOS, a ušetří tak i zkušenému uživateli práci se zaváděním dat do databáze.

Jelikož EkonomOS je modulární systém, který umožňuje různě konfigurovat sestavu modulů i uživatelské menu, vytvořil jsem již nakonfigurovanou funkční distribuci, jejíž menu obsahuje odkazy na všechny doposud vytvořené moduly. I v tomto případě je tak administrátor ušetřen vytváření položek menu a hned po prvním spuštění jsou mu k dispozici funkce všech modulů.

Dalším cílem práce bylo přiblížit EkonomOS ještě více profesionálním produktům z nabídky placeného softwaru pro vedení podnikové agendy. Proto vznikly nové moduly: Kniha jízd, Faktury a moduly pro export a import dat. Kniha jízd obsahuje také WAP rozhraní pro vzdálený přístup z mobilního telefonu. Modul Faktury, pomocí návaznosti na původní moduly Adresy, Sklady a Číselníky, umožní velmi rychlé a efektivní generování faktur. Tyto faktury je také možno exportovat do XML a HTML formátu. Další export/import dat je možný pomocí formátu CSV.

Velmi praktický je také způsob tisku dat ze systému. Je realizován pomocí potlačení designových prvků, jež nejsou určeny pro tisk, přímo v souboru kaskádových stylů. Proto může uživatel velmi jednoduše vytisknout jen důležitá data z kteréhokoliv modulu.

Doufám, že implementace nových modulů a instalátoru nabídne koncovému uživateli opět o něco mocnější nástroj, který osloví i laického uživatele bez znalosti programování a nabídne mu uživatelsky přívětivou instalaci i obsluhu. EkonomOS ovšem stále nabízí prostor pro vývoj, především co se týče exportu dat do jiných formátů než CSV. Snad jeho rozvoj neustane a za rok třeba opět nabídne další užitečné funkce.

CONCLUSION

In main opinion the most important contribution of my work is implementation the new and very simple installation system of EkonomOS.

EkonomOS is a modulation system with adjustable menu and module arrangement. So I made a distribution, which is installed by the first installation and lets user very quickly start the work with system EkonomOS without other module and menu configuration.

The other point of my work was improving the EkomomOS functions and making him more powerful in competition with professional products which are not open-source. Therefore I made new modules *Kniha jízd* (ride book), *Faktury* (invoices) and modules for data exporting/importing. The module *Kniha jízd* (ride book) has the WAP interface for the remote access from some mobile device. We should export data from the module *Faktury* (invoices) into XML or HTML format. For other data exporting/importing we should use the CSV data type.

Very useful is printing in system EkonomOS. The printing output is realized with CSS property `media print`. This property disables some design parts which should not be printed. So we get only data-sets we need without pictures etc.

I hope this new modules makes from EkonomOS more powerful and more user-friendly system. Perhaps the evolution of this open-source system is not at the end.

SEZNAM POUŽITÉ LITERATURY

- [1] LIM, John. *ADOdb Frequently Asked Questions : What is ADOdb?* [online]. c2000-2004 [cit. 2007-03-09]. Dostupný z WWW: <<http://adodb.sourceforge.net/adodb-faq.html>>.
- [2] LIM, John. *ADOdb Database Abstraction Library for PHP (and Python)* [online]. c2000-2004 [cit. 2007-03-09]. Dostupný z WWW: <<http://adodb.sourceforge.net/index.html>>.
- [3] W3C. *Extensible Markup Language (XML) : Introduction* [online]. c1996-2003 , 2007/05/08 17:40:49 [cit. 2007-05-20]. Dostupný z WWW: <<http://www.w3.org/XML/>>.
- [4] *XML : Syntaxe XML* [online]. 2007 , 23:19, 14. 5. 2007 [cit. 2007-05-20]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/XML#Syntaxe_XML>
- [5] *RFC - Wikipedie, otevřená encyklopedie : RFC* [online]. 2007 , 22:01, 10. 3. 2007 [cit. 2007-05-19]. Dostupný z WWW: <<http://cs.wikipedia.org/wiki/RFC>>.
- [6] ZIMÁČEK, Tomáš. *Skladový modulární systém s architekturou klient-server*. [s.l.], 2006. Vedoucí bakalářské práce Ing. Tomáš Dulík.
- [7] MALANÍK David. *Ekonomický modulární systém s architekturou klient-server*. [s.l.], 2006. Vedoucí bakalářské práce Ing. Tomáš Dulík.

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

ADODB	ADODB je objektová knihovna, v jazyce PHP či Python, pro přístup k databázím.
BSD licence	BSD licence je licence pro svobodný software.
MIME	MIME - (Multipurpose Internet Mail Extensions) udává o jaký typ dokumentu jde. Obrázek, HTML, text...
RFC	RFC je zkratka anglického výrazu request for comments (žádost o komentáře), která se používá pro označení řady standardů a dalších dokumentů popisujících Internetové protokoly, systémy apod. Jak už název napovídá, RFC jsou oficiálně považovány spíše za doporučení než normy v tradičním smyslu, přesto se podle nich řídí drtivá většina Internetu.
CSS	Kaskádové styly - pro stylování HTML dokumentů (http://www.w3c.org/css)

SEZNAM OBRÁZKŮ

Obrázek 1: Úvodní obrazovka instalátoru	20
Obrázek 2: Instalátor - vytvoření účtu administrátora.....	21
Obrázek 3: Dokončení instalace	21
Obrázek 4: Kniha jízd – výpis služebních vozů	24
Obrázek 5: Kniha jízd – vkládání záznamu o služebním voze.....	25
Obrázek 6: Kniha jízd – detailní výpis údajů o voze.....	27
Obrázek 7: Kniha jízd - vkládání záznamu o služební jízdě	28
Obrázek 8: Kniha jízd - výpis záznamů.....	29
Obrázek 9: Kniha jízd - detailní výpis služební jízdy.....	30
Obrázek 10: WAP - Modul kniha jízd - úvodní menu (WAP emulátor NOKIA S40 DP 2.0 SDK).....	31
Obrázek 11: WAP - Modul kniha jízd – vybrání vozu z vozového parku, vkládání údajů do formuláře pro vložení záznamu do knihy jízd a výpis údajů z knihy jízd.....	32
Obrázek 12: Příklad faktury generované modulem Faktury v systému EkonomOS	37
Obrázek 13: Modul Faktury - přidání položky do seznamu fakturovaných služeb.....	38
Obrázek 14: Modul Faktury - seznam fakturovaných služeb.....	38
Obrázek 15: Modul Faktury – tvorba faktury – zadání fakturačních údajů	39
Obrázek 16: Modul Faktury – tvorba faktury – vkládání předmětu faktury	40
Obrázek 17: Export dat - modul Export CSV - výběr tabulky a kódování.....	44
Obrázek 18: Modul Import CSV - formulář pro výběr importovaného souboru.....	46

SEZNAM PŘÍLOH

P I - Struktura tabulek přidanych modulů

PŘÍLOHA P I: STRUKTURA TABULEK PŘIDANÝCH MODULŮ

kjizd_auta

Sloupec	Typ
ida	tinyint(3)
datetime	timestamp
popis	varchar(255)
znacka	varchar(100)
typ	varchar(255)
rv	varchar(4)
palivo	varchar(100)
startkm	mediumint(7)
koupeno	date

kjizd_jizdy

Sloupec	Typ
idk	mediumint(6)
idu	mediumint(4)
ida	mediumint(4)
datetime	timestamp
od	datetime
do	datetime
start	varchar(255)
stop	varchar(255)
km1	mediumint(6)
km2	mediumint(6)
tank	tinyint(3)
tankkm	mediumint(6)
ucel	varchar(255)

faktury_dodavky

Sloupec	Typ
id	mediumint(6)
id_faktury	mediumint(6)
id_polozky	mediumint(6)
id_sluzby	mediumint(6)
kusu	smallint(5)
dph	tinyint(3)

faktury_priznaky

Sloupec	Typ
id_faktury	mediumint(10)
ok	varchar(1)
zauctoval	mediumint(6)
datetime	datetime

faktury_seznam

Sloupec	Typ
id	mediumint(10)
cislo	mediumint(10)
datetime	datetime
idzakaznik	mediumint(6)
iddodavatel	mediumint(6)
platba	tinyint(1)
datum	date
datumspla	date
datumdan	date
sleva	tinyint(1)
druh	tinyint(1)
ciselnik	mediumint(6)
zauctoval	mediumint(6)
bu	varchar(50)
ks	varchar(5)

faktury_sluzby

Sloupec	Typ
id	mediumint(6)
nazev	varchar(255)
cena	float

