VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

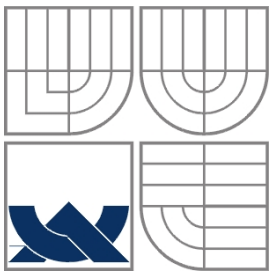
# WEBOVÝ SYSTÉM PRO VEDENÍ KNIHY JÍZD

BAKALÁŘSKÁ PRÁCE
BACHELOR´S THESIS

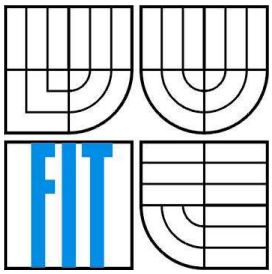
AUTOR PRÁCE
AUTHOR

LUKÁŠ NEDVĚD

BRNO 2007



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

# WEBOVÝ SYSTÉM PRO VEDENÍ KNIHY JÍZD

WEB-BASED SYSTEM FOR REGISTER OF JOURNEYS

BAKALÁŘSKÁ PRÁCE
BACHELOR´S THESIS

AUTOR PRÁCE
AUTHOR

LUKÁŠ NEDVĚD

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. ROMAN SKŘIVÁNEK

BRNO 2007

# Abstrakt

Tato práce pojednává o tvorbě on-line aplikace kniha jízd, pro správu jízd na internetu. Popisuje návrh aplikace a prostředky použité při její tvorbě, dále se zabývá zajímavými problémy, které musely být při implementaci řešeny. V závěru pak nastiňuje možnosti dalšího rozšíření.



# Klíčová slova

Kniha jízd, PHP, MySQL, CSS, JavaScript, internetová aplikace.



# Abstract

This project presents the creation of on-line application register of company travels, for the administration of these travels on the Internet. It describes design of the applications and devices used in its formation, dealing with interesting problems that need to be resolved in the implementation. In conclusion, it outlines the possibility of further extension.



# Keywords

Register of Journeys, PHP, MySQL, CSS, JavaScript, web application.



# Citace

Nedvěd Lukáš: Webový systém pro vedení knihy jízd. Brno, 2008, bakalářská práce, FIT VUT v Brně.

# Webový systém pro vedení knihy jízd

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pan ing. Skřivánka. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

……
Lukáš Nedvěd
14. května

## Poděkování

Děkuji všem, kdo mi pomohli s touto prací, především pak svojí rodině za poskytnutí příjemného zázemí.

# Obsah

# Úvod

V dnešním moderním světě, kterému vládnou počítače a internet, je potřeba uchovávat většinu dat v elektronické podobě. A tak i kniha jízd, což je tiskopis obsahující předtištěné listy formuláře, do nichž jsou údaje o cestách ručně vypisovány, potřebuje novou modernější formu odpovídající své době.

Elektronická on-line kniha jízd s sebou navíc přináší další nesporné výhody oproti papírové verzi, která se mohla jednoduše ztratit, poškodit či jinak znehodnotit. Je to především možnost uchovávat data v elektronické podobě, kontrola návaznosti kilometrů, jednoduché filtrování řidičů i vozidel a přehledné zpracování pro tisk.

# 1 Požadavky na knihu jízd

Povinnost vést záznamy o provozu vozidla určuje zákon č. 111/1994 Sb. a vyhláška č. 187/1994 Sb. Z ní vyplývá, že knihu jízd potřebujete pro vozy v obchodním majetku, v nájmu, pořízené na leasing a vypůjčené. V případě, že firemní vozidlo používáte i k soukromým cestám, musíte je do knihy jízd také uvádět.

## 1.1 Co vše musí kniha jízd obsahovat

Jsou stanoveny jisté minimální údaje, které musí být v každé knize jízd. V knize jízd by měla být bližší identifikace řidiče a vozidla včetně jeho SPZ, datum jízdy, její cíl a účel pokud se jedná o jízdu služební, dále pak je nutné uvést stav tachometru před jízdou a na konci jízdy. Pokud během jízdy došlo k čerpání pohonných hmot, je nutné taktéž tuto skutečnost uvést a to jak množství, tak i cenu. Dále je možné uvádět různé poznámky, například zda při cestě do zahraničí bylo placeno mýtné, podrobnosti o závadách a další.

## 1.2 Jak má být kniha jízd vedena

Správně vedená kniha jízd musí obsahovat chronologicky uspořádané záznamy a veškerá data na sebe musí navazovat. To se týká především stavu tachometru a jeho počátečních a koncových stavů.

## 1.3 Jak je to s elektronickou knihou jízd

Jelikož zákon nevyžaduje knihu jízd ve voze a ani policie u běžných firemních vozidel v případě kontroly knihu jízd nevyžaduje, je možné vést knihu jízd elektronicky, jelikož papírová kniha jízd je již zastaralá. Elektronická kniha jízd tak zajišťuje určitý komfort při vedení tohoto dokladu, ať už jde o jednoduché filtrování a pohodlné přidávání dat, tak i o jejich následnou správu a kontrolu.

## 1.4 Definice pojmů

Seznam a definice důležitých pojmů.

- **Kniha jízd** – evidence jízd, které jsou vykonány služebními nebo soukromými vozidly za účelem podnikání, může být v tištěné či elektronické formě

- **Služební jízda** – cesta související s výkonem povolání nebo také jízda pro hospodářské účely (jízdy do opraven a zkušební jízdy po opravě)
- **Soukromá jízda** – cesta konaná za privátním účelem

# 2 Požadavky na aplikaci

Cílem práce je s pomocí vhodných modelovacích nástrojů navrhnout systém pro vedení knihy jízd prostřednictvím webového rozhraní. Systém by měl být provozuschopný na vlastním serveru nebo na nějakém freehostingu, tudíž je potřeba knihu jízd zabezpečit heslem. Aplikace musí umožňovat rychlé zadávání údajů o provedené cestě. Další možností by mělo být třídění dat podle určitého filtru (řidiči, vozidla) a exporty dat pro případné zpracování v jiných programech.

## 2.1 Architektura aplikace

Architektura on-line aplikace Kniha jízd lze rozdělit do následujících tří vrstev.

- **datová vrstva** – zde se jedná o vlastní data uložená v databázi
- **aplikační vrstva** – je v podstatě vlastní aplikace. Zajišťuje operace nad daty, jejich kontrolu, propojení mezi uživatelským rozhraním a databázovým systémem.
- **klientská vrstva** – je to vlastně uživatelské rozhraní, které slouží pro interakci s uživatelem

## 2.2 Zvolené technologie

Technologie byly zvolené s ohledem na použití, jak na vlastním, tak na cizím, například freehostingovém serveru.

### 2.2.1 UML

UML je zkratka pro Unified Modeling Language, což je v softwarovém inženýrství grafický jazyk pro vizualizaci, specifikaci, navrhování a dokumentaci programových systémů. Důležitá je srozumitelnost, rychlost nakreslení a snadnost změny či navržení alternativ řešení.

Při návrhu bylo využito diagramu tříd – class diagrams (ER diagram), pro návrh struktury databáze.

### 2.2.2 MySQL

MySQL je multiplatformní relační databázový systém, který vytvořila Švédská firma MySQL AB. Jak už název vypovídá komunikace probíhá pomocí jazyka SQL (Structured Query Language). Podobně jako u ostatních SQL databází, se jedná o dialekt tohoto jazyka s některými rozšířeními.

Mezi jeho největší klady patří implementovatelnost, lze jej tedy použít jak v nejrozšířenějším operačním systému MS Windows, tak i v operačním systému Linux, ale i v jiných operačních systémech. Díky své rychlosti a především díky tomu, že se jedná o volně šiřitelný software je v poslední době téměř nejoblíbenějším systémem a to i přesto, že má některá omezení, které konkurenční systémy nemají.

Největší omezení je, že verze 4 nepodporuje databázové pohledy, triggery ani uložené procedury, které již MySQL ve verzi 5 podporuje.

I přes tato omezení bylo pro implementaci zvoleno MySQL ve verzi 4, protože v současné době je stále nejrozšířenější a díky tomu se právě objevuje na většině freehostingových službách. Současně s touto volbou bylo potřeba veškerou logiku přesunout do programování v PHP.

### 2.2.3 PHP

PHP je zkratka pro Hypertext Preprocessor (původně Personal Home Page). Jedná se o skriptovací jazyk určený hlavně pro generování dynamických webových stránek., kde se skripty zpracovávají na straně serveru (nejčastěji APACHE) a k uživateli je přenášen pouze výsledek skriptu.

Syntaxe jazyka kombinuje hned několik programovacích jazyků (Perl, C, Pascal a Java). PHP je nezávislý na platformě, takže skripty fungují i bez úprav na mnoha různých operačních systémech. PHP obsahuje rozsáhlé knihovny funkcí pro zpracování textu, grafiky, práci se soubory, přístup k většině databázových serverů včetně MySQL.

Díky tomu, že kombinuje vlastnosti více programovacích jazyků a přitom si zachovává svou jednoduchost, je PHP jedním z nejpoužívanějších technologií při programování a tvorbě webových aplikací.

S verzí PHP 5 se výrazně zlepšil přístup k objektově orientovanému programování podobný Javě. Tyto změny jsou oproti předchozí verzi dost značné, takže je možné, že některé aplikace napsané objektově pod PHP ve verzi 4 nemusí v novějších verzích PHP fungovat správně. I díky tomuto rozdílu bylo zvoleno PHP 4, které je stále nejrozšířenější (viz. článek Verze PHP v ČR – únor 2008 dostupný na URL http://php.vrana.cz/verze-php-v-cr-unor-2008.php).

Při vývoji systému byl upřednostněn až na výjimky procedurální přístup před objektovým, a to zejména kvůli rozdílům mezi jednotlivými verzemi PHP a jejich přístupem k objektovému programování.

### 2.2.4 XHTML

XHTML je zkratkou pro anglické extensible hypertext markup language, tedy do češtiny přeloženo jako rozšiřitelný hypertextový značkovací jazyk. XHTML se vyvinulo ze staršího HTML a předpokládalo se, že bude právě nástupcem HTML u kterého byl vývoj ve verzi 4.01 ukončen. Avšak v roce 2007 došlo k založení skupiny, která oživila vývoj HTML a v současné době se pracuje na verzi s označení HTML 5.0. XHTML je stále paralelně vyvíjeno a nyní se pracuje na verzi 2.0.

Jaký je tedy rozdíl mezi HTML a XHTML? V XHTML na rozdíl od HTML musí být všechny tagy ukončené a to včetně těch nepárových. Dále všechny tagy musí být zapsány malými písmeny a všechny hodnoty atributů musí být uzavřeny do uvozovek. Dokument musí začínat XML deklarací. Například XHTML 1.0 existuje ve třech verzích.

- **XHTML 1.0 Strict** – se používá pro strukturované dokumenty, kde nejsou použity formátovací značky související s rozvržením stránky, či značky použité k formátování vzhledu výsledného dokumentu. I přestože se předpokládá stylování pouze pomocí CSS, tak tato verze obsahuje některé formátovací elementy jako například tag <b> či <i>.
- **XHTML 1.0 Transitional** – narozdíl od nejpřísnější verze XHTML 1.0 Strict povoluje některé zavržené elementy určené pro formátování stránek a je tak vhodný pro starší prohlížeče, které nezvládají kaskádové styly CSS.
- **XHTML 1.0 Frameset** – umožňuje stejně tak jako XHTML 1.0 Transitional používat některé zastaralé značky a navíc přidává i podporu pro rámce.

Hlavním důvodem pro volbu XHTML je jeho přísnější specifikace. Ke kódování stránek v aplikaci byla použita nejpřísnější verze XHTML 1.0 Strict, díky níž by výsledné zobrazení mělo být stejné napříč všemi prohlížeči. I když podle této specifikace XHTML bychom měli dokument zasílat s jiným MIME typem než klasické HTML dokumenty, tak aplikace posílá vygenerované stránky s klasickým typem text/html místo doporučovaného application/xhtml+xml.

Tato volba byla ovlivněna především tím, že při jakékoliv syntaktické chybě vůči správné sestavenosti, musí prohlížeč zobrazit chybové hlášení a nepokračovat dále ve vykreslování stránky. Tudíž například nekompletně stažená stránka se nemusí vůbec zobrazit. Další problém je, že se stránka v některých prohlížečích zobrazí až po úplném načtením stránky, za což může právě prvotní kontrola správného sestavení dokumentu. Navíc MIME typu application/xhtml+xml nerozumí v současné době docela velké procento interpretů, například i jeden z nejpoužívanějších prohlížečů Internet Explorer s ním umí pracovat až od současné verze 7.

### 2.2.5 CSS

Kaskádové styly, neboli CSS z anglického Cascading Style Sheets, česky překládané jako tabulky kaskádových stylů. Jazyk byl navržen stejně jako XHTML standardizační organizací W3C, což je mezinárodní konsorcium, jehož členové společně s veřejností vyvíjejí webové standardy pro internet. I když je CSS standarizované, tak podobně jako HTML je jeho interpretace v různých prohlížečích odlišná.

CSS slouží pro popis formátování textu dokumentu psaného ve značkovacím jazyce, jakým je XHTML. Hlavním smyslem je umožnit návrhářům oddělit vzhled dokumentu od jeho struktury a obsahu. Dřívější verze HTML také poskytovali možnosti formátování, ale tento vývoj z hlediska sémantiky webu nebyl žádoucí.

CSS má oproti formátování přímo v HTML řadu výhod, mezi něž patří rozsáhlejší možnosti formátování, které se projeví při formátování bloku, kde by se okraje elementu museli v HTML tvořit za pomoci vnořených tabulek. Dále díky CSS mají všechny stránky konzistentní styl. Tedy stejné úrovně nadpisů, stejné styly jednotlivých elementů a jiných zdůraznění textu. Dosáhnutí podobného výsledku by v HTML bylo také možné, ale kód by se tím znepřehlednil a nikdy by nebyla úplná jistota, zda jsme některou značku HTML nezapomněli naformátovat.

Největší výhodou je pravděpodobně oddělení struktury a stylu a z toho navazující dynamická práce se styly. Změna designu se tak díky za pomoci CSS může odehrát během pár minut a změny v barevném tónování jsou otázkou několika sekund. Jelikož soubor .css většinou bývá uložený zvlášť, tak se také ukládá do mezipaměti počítače a tím urychluje načítání stránek.

Další výhodou je, že může existovat několik různých stylů pro různá výstupní zařízení. Můžeme tedy ovlivnit, jak výsledek bude vypadat na počítači, či v upravené podobě na PDA, nebo na výstupu na tiskárnu. Této možnosti bylo v aplikaci využito při formátování výpisů k tisku, kde se některé zbytečné elementy pro výstup na tiskárně skryjí.

Jedinou možnou nevýhodou je právě rozdílná interpretace v různých prohlížečích, ale pokud je webdesigner seznámen s odlišnostmi jednotlivých prohlížečů, jde se těmto problémům z části nebo zcela vyhnout.

Pro aplikaci byla využita volně šiřitelná šablona z webu www.freecsstemplates.org, která byla upravena pro potřeby aplikace.

### 2.2.6 JavaScript

JavaScript, který vznikl ve stejné době jako PHP (tedy v roce 1995), byl původně vyvíjen společností Netscape, což mělo za následky neúplnou podporu v tehdejším prohlížeči Microsoft Internet Explorer 3. V době vývoje už ovšem byl Internet Explorer 3 zastaralý a JavaScript je podporován napříč všemi prohlížeči.

JavaScript je často zaměňován s Javou. Java je samostatný programovací jazyk. Má s JavaScriptem pouze podobnou syntaxi. JavaScript je klientský skript. To znamená, že se program odesílá se stránkou do prohlížeče a teprve tam je vykonáván, což je rozdíl oproti serverovému skriptování PHP, kde jsou zdrojové kódy zpracovány na serveru a až výsledek je odesílán do prohlížeče. Existují i varianty pro server, ale ty nejsou tak běžné.

V aplikaci je JavaScript použit na zpříjemnění práce s aplikací, ale vše je plně funkční i se zakázaným JavaScriptem. Na tento cíl byl kladen nesmírný důraz, protože aplikace musí být plně funkční i bez podpory JavaScriptu (kvůli přísupu z mobilních zařízení atd.).

### 2.2.7 Export CVS

CSV je zkratkou pro Comma-separated values, tedy hodnoty oddělené čárkami. Jedná se o jednoduchý souborový formát určený pro výměnu tabulkových dat. Soubor ve formátu CSV sestává z řádků, ve kterých jsou jednotlivé položky odděleny znakem čárka. Hodnoty pak mohou, ale nemusí být umístěny do uvozovek, čímž se v hodnotách dá použít také znaku čárka. Pokud by se v textu vyskytovali uvozovky, tak ty musí být zdvojeny.

Pro jeho jednoduchost, nenáročnost a čitelnost i bez specializovaného software se tento formát používá pro výměnu informací mezi různými systémy. Dnes se již spíše přistupuje k univerzálnějšímu a modernějšímu XML, které je ovšem složitější.

Díky jednoduchosti byl pro export dat z aplikace zvolen formát CVS, který je velmi snadno převeditelný do velmi často používaného programu Microsoft Excel.

### 2.2.8 Export XML

XML za anglické zkratky pro eXtensible Markup Language, česky rozšiřitelný značkovací jazyk není přímo jazyk, ale meta-jazyk. XML je technologie, která umožňuje jednoduše, přehledně a efektivně pracovat s čistými daty. Jazyk je určen především pro výměnu dat mezi aplikacemi a pro publikování dokumentů. Jazyk umožňuje popsat strukturu dokumentu z hlediska věcného obsahu jednotlivých částí, nezabývá se sám o sobě vzhledem dokumentu nebo jeho částí. Prezentace dokumentu (vzhled) se potom definuje připojeným stylem.

XML dokument je text, vždy Unicode, v Česku obvykle kódovaný jako UTF-8, ale jsou přípustná i jiná kódování. V XML vždy musí být vždy obsažen jeden kořenový element, pro export v aplikaci byl zvolen kořenový element <export>.

Vzhledem k snadnému parserování takovéhoto XML dokumentu byl tento formát zvolen i pro importy dat do aplikace.

## 2.3 Struktura aplikace

Následuje popis jednotlivých prvků a komponent, které se v systému vyskytují, spolu se specifikací jejich obsahu.

### 2.3.1 Členění stránek aplikace

Struktura všech stránek v aplikaci je totožná. Skládají se z hlavního menu, kde právě otevřená sekce je zvýrazněna jinou barvou. Pod menu je prostor pro logo a nadpis aktuální stránky, který se zobrazuje také v titulku. Pod tímto nadpisem je tzv. drobečková navigace, která by měla usnadňovat navigaci a zjednodušit tak přehled, na které ze stránek aplikace se uživatel nachází.

Obsah je pak členěn na dva sloupce, kde v levém širším sloupci jsou různé formuláře, výpisy a v pravém je nápověda týkající se levého sloupce, občas doplněna souvisejícími odkazy.

#### 2.3.1.1 Přihlášení

Dokud se uživatel nepřihlásí, je po něm pro práci s knihou jízd vyžadováno přihlášení za pomoci přihlašovacího jména a hesla.

#### 2.3.1.2 Úvodní stránka

Na úvodní stránce jsou informace o posledním přihlášení a odkazy na nejčastější činnosti, které lze s on-line knihou jízd provádět, jako je například přidávání nových jízd, vozidel či řidičů.

Pokud do knihy jízd zatím není vloženo žádné vozidlo či řidič, je na úvodní stránce červeně zvýrazněné chybové hlášení, které informuje o potřebě tyto údaje doplnit, jinak není s knihou možná další práce. Zároveň jsou v tomto hlášení obsaženy i přímé odkazy na formuláře, kde lze tyto data doplnit.

#### 2.3.1.3 Kniha jízd

Stejně tak jako na úvodní stránce, tak i zde pokud není v knize doposud vložen žádný řidič či vozidlo, je uživatel na tuto skutečnost upozorněn chybovým hlášením, doplněným odkazy na vložení chybějících údajů.

Jestliže jsou data již vložená a kniha jízd obsahuje nějaké cesty, je v hlavním okně zobrazen chronologický výpis cest a nad ním přehledná tabulka o počtu jízd, ujetých kilometrech a údaje o čerpání pohonných hmot.

Data ve výpisu mají přehlednou strukturu, kdy na jednom řádku je menším písmem datum jízdy, její typ (soukromá, či služební) a jméno řidiče. Úplně na konci řádku jsou odkazy na případnou editaci jízdy či její vymazání.

Na dalším řádku, který je psaný větším písmem je cíl cesty a její účel, pokud se jednalo o služební cestu, v opačném případě, je zde jen informace o tom, že jízda byla soukromá a tudíž text nemusel být zadán. Následují informace o počtu ujetých kilometrů (pokud je nad tento údaj umístěn kurzor myši, zobrazí se stav tachometru na začátku a na konci jízdy) a případných čerpáních pohonných hmot.

Pokud řidič klikne na červený křížek označující možnost vymazání jízdy, je upozorněn, že se jedná o nevratnou akci a dotázán, jestli si je opravdu jist, že chce danou jízdu ze systému vymazat.

V pravém sloupci je umístěn filtr, kde lze nastavit zobrazování přehledu pouze pro vybraná vozidla či vybrané řidiče. Po potvrzení se vypíše přehled i jízdy dle vybraného filtru.

Po tímto filtrem jsou odkazy na další relevantní akce, jakými jsou přidání nové jízdy, nebo možnost aktuálně zobrazeného výpisu převést do verze pro tisk.

Úplně poslední ve sloupci je kratičká nápověda vztahující se k přehledu a možnostem filtrování.

#### 2.3.1.4 Přehled pro tisk

Je speciální stránka, která nemá stejný design, jako ostatní stránky v aplikaci. Jedná se o stránku na kterou se lze dostat z knihy jízd zvolením možnosti „Přehled pro tisk“. Na této stránce jsou pouze nutné údaje a tabulky s přehledy. Tato stránka má přiřazeny dva css styly jeden určený k zobrazení a druhý určený pro tisk.

Ve verzi k zobrazení je dostupný odkaz „změnit rozmezí“, po jehož rozkliknutí se zobrazí formulář, na kterém je možné vybrat časové rozmezí pro zobrazovaná data v přehledu. Tento odkaz je v tiskovém stylu skrytý.

#### 2.3.1.5 Administrace – vložení řidiče

Stránka obsahuje velmi jednoduchý formulář, ve kterém stačí zadat pouze jméno řidiče. Pokud je řidič vložen v pořádku, je o tom uživatel informován a je mu dána možnost nechat si vypsat všechny vložené řidiče.

#### 2.3.1.6 Administrace – vložení vozidla

Formulář pro vložení vozidla je již o trochu složitější. Je potřeba zadat typ vozu (např. Ford Escort), pro pozdější identifikaci, jeho SPZ a počáteční stav tachometru.

Pokud se zadaná SPZ již v systému nachází, je o tom uživatel informován chybovou hláškou a problémové pole formuláře je zvýrazněno červenou barvou.

Stejně jako v případě vkládání vozidel je při úspěšném vložení uživatel informován a je mu dána možnost data zkontrolovat ve výpisu všech vozidel.

#### 2.3.1.7 Administrace – vložení jízdy

Pokud kniha jízd obsahuje více řidičů či vozidel, je uživateli nabídnut jejich seznam v roletkovém menu. V opačném případě je zobrazeno pouze jméno řidiče či vozidla.

Následuje pole pro vložení data, které je doplněno odkazem na aktivní kalendář, ze kterého se dá datum v požadovaném formátu doplnit pouhým kliknutím.

Další je výběr typu jízdy. Tedy zda je jízda soukromá či služební. V případě výběru soukromé jízdy se skryjí některá políčka, která jsou povinná pouze pokud se jedná o služební jízdu.

V poli počáteční stav tachometru je předvyplněn stav tachometru podle vyplněných předchozích jízd vybraného vozila. Toto pole je needitovatelné, aby nemohlo docházet k chybám. Další pole je již možné vyplnit a je jím koncový stav tachometru.

V případě, že se jedná o služební jízdu je zobrazeno pole místo. Pokud do něj uživatel začne psát začínají se předvyplňovat nejčastější cíle, podle toho jaká zrovna píše písmena. V případě, že mezi nejčastější cíle patří například Hradec Králové, tak už při napsání znaku „H“, je do pole vyplněn celý název města.

Pod polem místo, je pole pro účel jízdy, který jde buď ručně vypsat, nebo je zde možnost ho doplnit z nejčastějších používaných, které se zobrazují v roletkovém menu.

Formulář je ukončen poli pro údaje o čerpání pohonných hmot.

V případě jakýchkoliv nesmyslných údajů, nebo nevyplněných hodnot při odesílání formuláře, je opět uživatel informován chybou hláškou a všechny nesprávně vyplněná pole jsou zvýrazněna červenou barvou, aby se zajistila jejich snadná oprava.

#### 2.3.1.8 Administrace správa řidičů a vozidel

Správa řidičů, stejně tak jako správa vozidle je řešena formou tabulky, ve které jsou všechny údaje vypsány pod sebou a doplněny tlačítky editovat a křížkem, který slouží pro smazání dat.

V případě editace je uživatel přepnut na formulář, který vypadá stejně jako formulář pro vkládání, jen jsou v něm předvyplněny potřebné hodnoty.

V případě mazání dat, je uživatel varován, že smazáním daného řidiče, či vozidla dojde i k vymazání všech jízd, kterých se řidič či vozidlo účastnili. K vymazání dojde až po potvrzení této skutečnosti.

#### 2.3.1.9 Administrace – import a export dat

Na stránce s exporty jsou k dispozici soubory s data ve formátu csv a xml. Data jsou rozdělena do tří logických kategorií a to na řidiče, vozidla a jízdy.

V případě importu (který je možný pouze z XML) se typ dat automaticky rozpozná a údaje jsou uloženy do potřebných tabulek v databázi.

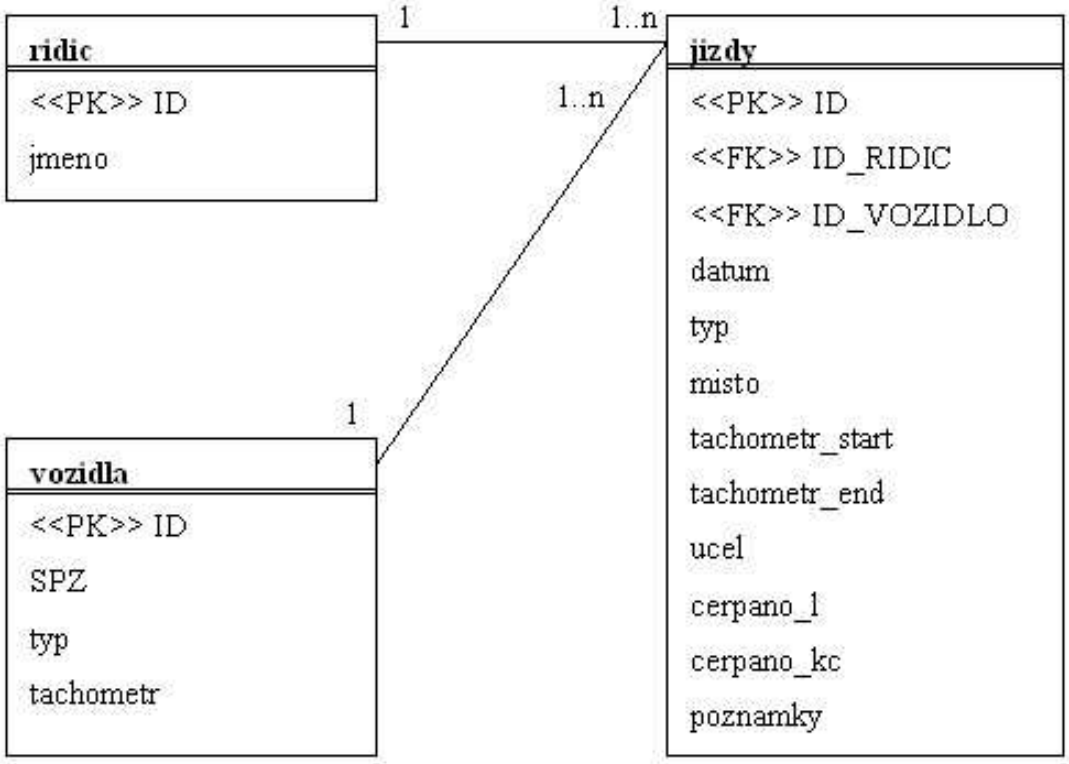
#### 2.3.1.10 Administrace – nastavení přístupu

Poslední položkou v administraci je možnost nastavení si vlastního přihlašovací jména a hesla. Jakoukoliv změnu v těchto údajích, je ale vždy potřeba potvrdit současným heslem, aby nemohlo dojít k nechtěné změně.

# 2.4 Datový rozbor

## 2.4.1 Vztahy entit

Entita *jizdy* představuje jeden řádek v klasické knize jízd. Každá taková jízda musí obsahovat právě jednoho řidiče a právě jedno vozidlo, kterým byla vykonaná, proto jsou mezi jednotlivými entitami kardinalita typu 1..n.




**obrázek 1 - ER diagram**

## 2.4.2 Vlastnosti entit

### 2.4.2.1 Tabulka jízd

Tabulka j*izdy* je nejdůležitější entita datového modelu. Kromě toho, že definuje každou jízdu uloženou v aplikaci obsahuje i všechny údaje o ní.

Atributy tabulky jízdy lze rozdělit na dvě skupiny. V první skupině jsou primární klíče ukazující na příslušné entity a druhá skupina obsahuje jednotlivé hodnoty.

- **id** – je identifikátor jízdy, sloužící při následné editaci či odstranění
- **id_ridic** – je identifikátor řidiče ukazující na příslušnou entitu řidiče v tabulce *ridic*
- **id_vozidlo –** je indetifikátor vozidla ukazující na příslušnou entitu vozidla v tabulce vozidla
- **datum –** obsahuje časový údaj o dnu konání jízdy. Ze zákona není možné uskutečňovat jízdy v intervalu delším jak jeden den, proto je nutné delší cesty rozepsat do více jízd a dnů
- **typ –** určuje zda se jedná o služební jízdu (typ 0), nebo o jízdu soukromou (typ 1)
- **tachometr_start a tachometr_end –** jsou čísla, která určují počáteční a koncový stav tachometru
- **misto –** je textový popis cíle jízdy
- **ucel –** je textový popis účelu či důvodu jízdy
- **cerpano_l** a **cerpano_kc –** jsou číselné ukazatele o množství a ceně načerpaného plaiva
- **poznamky –** je nepovinný textový údaj s doplňujícími informacemi

| atribut | datový typ | rozsah |
|---|---|---|
| id | mediumint(5) | 0-99 999 |
| id_ridic | tinyint(2) | 0-99 |
| id_vozidlo | tinyint(3) | 0-127 |
| datum | date | |
| typ | enum('0', '1') | |
| místo | varchar(150) | 150 znaků |
| ucel | text | 65535 znaků |
| tachometr_start | mediumint(6) | 0-999 999 |
| tachometr_end | mediumint(6) | 0-999 999 |
| cerpano_l | smallint(4) | 0-9999 |
| cerpano_kc | mediumint(5) | 0-99 999 |
| poznamky | text | 65535 znaků |

**obrázek 2 - tabulka atributů entity jízdy**

#### 2.4.2.2 Tabulka řidičů

Tato entita obsahuje všechny osoby, které mají oprávnění řídit služební vozidlo. Obsahuje pouze dva atributy:

- **id** – je jednoznačný identifikátor řidiče
- **jmeno** – obsahuje jméno a příjmení řidiče včetně titulů

| atribut | datový typ | rozsah |
|---|---|---|
| id | mediumint(5) | 0-99 999 |
| jmeno | varchar(50) | 50 znaků |

**obrázek 3 - tabulka atributů entity řidiči**

#### 2.4.2.3 Tabulka vozidel

Jejím obsahem jsou vozidla registrovaná v České republice, která jsou vlastněna živnostníkem, či soukromou osobou používána pro služební účely.

- **id** – je jednoznačný identifikátor vozidla
- **spz** – je státní poznávací značka, která by měla být jedinečná
- **typ** – je typ vozu
- **tachometr** – je počáteční stav tachometru při nabytí vozidla

| atribut | datový typ | rozsah |
|---|---|---|
| id | mediumint(5) | 0-99 999 |
| spz | varchar(10) | 10 znaků |
| typ | varchar(50) | 50 znaků |
| tachometer | mediumint(6) | 0-999 999 |

**obrázek 4 - tabulka atributů entity vozidla**

# 3 Implementace

Následující kapitoly popisuje detailněji implementaci jednotlivých částí systému.

## 3.1 Tvorba databázových tabulek

Každá tabulka v databázi má primární klíč ID, který je jednoznačným unikátním identifikátorem v systému. Pro tyto primární klíče je v implementaci všude využito modifikátoru AUTO_INCREMENT, který zajišťuje automatické generování následující číselné hodnoty.

Dále byly některým sloupcům v tabulkách přiděleny indexy, jimiž je systém žádán o dohled nad jednotlivými hodnotami sloupců, či nad jejich kombinacemi. Výsledkem indexování, je zvýšený výkon při načítání a vyhledávání záznamů, i když to může vést ke zpomalení vkládání nových položek a zvětšení celé databáze.

Vzhledem k možnému použití na starších systémech nebylo použito cizích klíčů a jejich klauzulí ON DELETE, jelikož tato klauzule funguje až u systému MySQL 4.0 a vyšší a tak je toto integritní omezení řešeno manuálním testování a případným vymazáním referenčních záznamů.

### 3.1.1 Manuální správa databáze

Pro manuální správu databáze a případné zálohy SQL tabulek je možné využít open source aplikaci PHPMyAdmin. Jedná se především o potřebný import dat z přiloženého SQL skriptu. Pro ten je možné použít i konzolovou verzi MySQL klienta, ale vzhledem k velké rozšířenosti tohoto nástroje se počítá s jeho přítomností na hostingovém serveru.

## 3.2 Implementace zajímavých funkcí

V této podkapitole je popsána implementace některých zajímavých funkcí systému na úrovni aplikační vrstvy. Popisované skripty jsou většinou implementovány ve skriptovacím jazyce PHP, který je podrobněji popsán v kapitole 2.2.3.

### 3.2.1 Přihlášení

I když PHP podporuje standardní HTTP autentizaci prostřednictvím proměnných *$_SERVER['PHP_AUTH_USER']* a *$_SERVER['PHP_AUTH_PW']*, je výhodnější implementovat přihlašování vlastní.

Mezi hlavní nevýhody HTTP autentizace patří, že přihlašovací data se přenášejí při každém stažení stránky nešifrovaně a navíc ve standardizované podobě, takže je pro různé sniffery jednodušší je získat. Další nevýhodou pak jsou problémy s odhlášením.

Proto se v aplikaci využívá vlastního formuláře, kde po ověření přihlašovacích údajů jsou údaje o přihlášenosti uloženy do session. Odhlášení je díky tomu triviální a stačí vymazat obsah session a uživatel je odhlášen. Jelikož je přihlašovací formulář v případě aktuálního nepřihlášení vložen do aktuální stránky, je možné ihned po přihlášení zobrazit požadovanou stránku. Díky tomu je možné posílat odkazy na zabezpečené stránky i e-mailem a uživateli se po přihlášení zobrazí přesně to, co očekává, že na zadané adrese najde. [5]

### 3.2.2 Solení hesel

I když aplikace prozatím počítá s pouhým jediným uživatelem (adminem), je dobré myslet na budoucí rozšíření a tak při ukládání hesel do databáze bylo využito techniky, které se česky říká solení hesel.

Hesla se běžně do databáze ukládají v hashované podobě. Díky tomu i když útočník získá data z databáze, nebude schopen v rozumném čase získat hesla uživatelů. Běžně se používají funkce MD5 a SHA1, i když dnes se již více přiklání k druhé zmiňované, vzhledem k tomu, že u MD5 byly objeveny bezpečnostní trhliny. [10]

I když jsou takto hesla šifrována, stále se je zde ukryto další bezpečnostní riziko. Při tomto způsobu v případě, že dva uživatelé mají stejné heslo, tak má i stejný hash a když to jeden z uživatelů jakkoliv zjistí, může se přihlásit i na druhého uživatele. Toto se řeší právě zmíněným solením hesel - při hashování hesla se na vstup navíc ještě přidává nějaký další řetězec, takzvaná sůl. Sůl je pokaždé jiná, výsledný hash je tak i pro stená hesla pokaždé jiný. Solí může být cokoliv. Může jít o náhodně vygenerovanou hodnotu, ale tím nám nastává problém, tuto hodnotu si někde ukládat. Proto je mnohem výhodnější jako sůl použít již nějaký uložený údaj, například uživatelské jméno.

Kromě zmíněné výhody, že i uživatelé se stejným heslem mají různé hashe svého hesla, přináší tato technika další výhody. Osolená hesla jsou například odolnější dolnější proti útokům s předgenerovanými tabulkami (rainbow tables), nebo před slovníkovými útoky. [7]

### 3.2.3 Obrana proti SQL Injection a XSS

SQL injection je technika napadení databázové vrstvy programu vsunutím (odtud „injection“) kódu přes neošetřený vstup a vykonání vlastního, samozřejmě pozměněného, SQL dotazu. Jeden způsob je ošetřit veškerá vstupní data vhodným způsobem, druhý je použít direktivu magic_quotes_gpc a veškeré hodnoty uzavírat do apostrofů.

Jenže u různých hostingů se nemůžeme na nastavení direktivy magic_quotes_gpc spolehnout a tak je potřeba vstupy ošetřit funkcí addslashes. Jenže pokud bude direktiva magic_quotes_gpc zapnuta, dojde k escapovaní na dvakrát – jednou kvůli magic_quotes_gpc a jednou funkcí addslashes.

Proto je nutné vytvořit vlastní funkci, která kontroluje direktivu magic_quotes_gpc a podle jejího nastavení se použije, či nepoužije funkce addslashes.

Dalším bezpečnostním problémem je Cross-site scripting, označovaný obvykle zkratkou XSS. XSS útok spočívá v tom, že se útočníkovi podaří do napadené stránky vložit vlastní HTML kód, který se při následném zobrazení v prohlížeči interpretuje jako HTML.

Toto nebezpečí má velkou škálu možností. Od pouhého narušení formátování stránky až po využití JavaScriptu, kdy útočník může důvěryhodný obsah stránek nahradit obsahem svým vlastním. Ještě větší nebezpečí přináší fakt, že podstrčený JavaScriptový kód se provádí v kontextu dané stránky. Útočník tak má například plný přístup k uživatelově cookies, může mu třeba prostřednictvím XSS ukrást jeho aktuální SID včetně platného přihlášení do aplikace.

Spolehlivá ochrana před XSS je přes všechny uvedené složitosti překvapivě jednoduchá – ošetřit důsledně všechny výstupy z aplikace funkcí htmlspecialchars().Ošetření dat pomocí htmlspecialchars() by mělo být pouze na výstupu. To především díky tomu, že do databáze patří pouze čistá aplikační data, nezatížená a nekontaminovaná jakýmkoliv konkrétním typem View. V případě exportů do jiných formátů, by nám mohli znaky převedené funkcí htmlspecialchars() vadit. [5][7]

# 4 Návrhy na rozšíření

Aplikace byla navržena otevřeně, takže je ji možné neustále rozšiřovat. Největším přínosem by byla podpora více uživatelů s rozdílnými přístupovými právy, kde by každý uživatel mohl přidávat, editovat a procházet pouze data, která by s ním souvisela. Administrátor, by pak mohl přistupovat ke všem datům.

Další možná rozšíření je možná implementovat v oblasti čerpání pohonných hmot. Další detailnější informace, například o místě čerpání, by mohli zpřehlednit tyto údaje a pro majitele by vznikla větší možnost kontroly těchto dat.

Zajímavým rozšířením by mohl být import dat z GPS umístěných ve vozidlech a díky napojení na některý mapový server skrz jeho API vykreslování tras na grafických mapách.

Po případných rozšířeních by se aplikace dala používat i v menších podnicích, zatímco teď je navržena pro živnostníka s malým vozovým parkem a nízkým počtem zaměstnanců – řidičů.

# 5 Závěr

Cílem této bakalářské práce bylo vytvořit funkční internetovou knihu jízd pro menší firmu, který splňuje všechny požadavky definované v zadání. Systém poskytuje příjemné, jednoduché a přehledné uživatelské rozhraní se snadným ovládáním a možnostmi sestavovat přehledné tabulky pro tisk.

I přesto, že aplikace je oproti komerčním systémům jednodušší, je plně funkční a obsahuje všechny důležité funkce pro snadnou a pohodlnou práci se služební evidencí cest, tudíž svůj účel splňuje. Její výraznou předností je nasazení na většině serverů a to včetně různých freehostingů. Její případná další rozšíření popsaná v kapitole 4 ji přiblíží komerčním nástrojům.

Díky této bakalářské práci jsem si rozšířil rozhled na poli tvorby internetových aplikací a všeobecně i řešení některých problémů v PHP a MySQL. Při testování uživatelského rozhraní aplikace v různých prohlížečích jsem si prohloubil znalosti v HTML kódování a v rozdílech je mezi jednotlivými prohlížeči.

# Literatura


[1] Kosek, J. *PHP, tvorba interaktivních internetových aplikací*
Grada Publishing, 1999, ISBN: 807169-373-1

[2] The PHP Group: *PHP: Hypertext Preprocessor*.
Dokument dostupný na URL: http://www.php.net, 1.05.2008

[3] MySQL AB: *MySQL Documentation*.
Dokument dostupný na URL: http://www.mysql.com, 1.05.2008

[4] Janovský, D.: *O tvorbě internetových stránek*.
Dokument dostupný na URL: http://www.jakpsatweb.cz, 1.05.2008

[5] Vrána, J.: *PHP triky - Weblog o elegantním programování v PHP pro mírně pokročilé*
Dokument dostupný na URL: http://php.vrana.cz, 1.05.2008

[6] Grundl, D.: *La Trine(kategorie PHP)*
Dokument dostupný na URL: http://latrine.dgx.cz/category/php, 1.05.2008

[7] Tichý, J.: *PHP GURU*
Dokument dostupný na URL: http://www.phpguru.cz, 1.05.2008

[8] *Wikipedie otevřená encyklopedie*
Dokument dostupný na URL: http://cs.wikipedia.org, 1.05.2008

[9] *Interval.cz*
Dokument dostupný na URL: http://www.interval.cz, 1.05.2008

[10] Klíma, J. *Hašovací funkce MD5 a další prolomeny!*
Dokument dostupný na URL: http://www.root.cz/clanky/hasovaci-funkce-md5-a-dalsi-prolomeny/, 2.05.2008

[11] *Internetová kniha jízd*
Dokument dostupný na URL: http://www.internetovaknihajizd.cz/, 2.05.2008

[12] *Specifikace webových standardů*
Dokument dostupný na URL: http://www.w3c.org, 2.05.2008

# Seznam příloh

Příloha 1. CD se zdrojovými texty