

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV INFORMAČNÍCH SYSTÉMŮ  
FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INFORMATION SYSTEMS

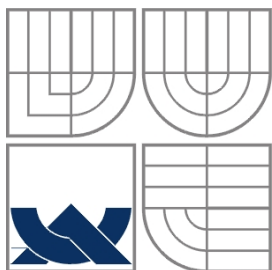
INFORMAČNÍ SYSTÉM CYKLOSERVISU

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

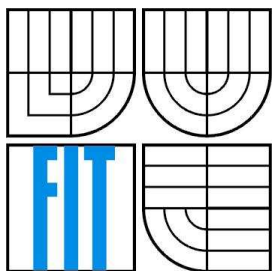
AUTOR PRÁCE  
AUTHOR

PETR ŠKAREK

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

# INFORMAČNÍ SYSTÉM CYKLOSERVISU

BIKESERVIS INFORMATION SYSTEM

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

AUTOR PRÁCE  
AUTHOR

PETR ŠKAREK

VEDOUCÍ PRÁCE  
SUPERVISOR

MGR. TOMÁŠ MASOPUST

BRNO 2007

## **Abstrakt**

Tématem této bakalářské práce je Informační systém cykloservisu vytvořený pomocí jazyka PHP a databázového systému MySQL. Projekt sestává ze dvou základních částí. V první jsou uvedeny obecné poznatky o informačních systémech, jazyce PHP, databázovém systému MySQL a dalších technologiích použitých v projektu. Druhá část popisuje použitou databázi pomocí jazyka UML a samotný informační systém, jeho rozdělení podle stupně uživatelských oprávnění, způsob zabezpečení a dále jednotlivé funkce a použité skripty. Praktickou část této bakalářské práce tvoří samotný informační systém a jeho ukázková databáze přiložené na CD.

## **Klíčová slova**

Informační systém, HTML, CSS, PHP, MySQL, Databáze

## **Abstract**

The topic of this bachelor's thesis is Bikeservice information system created in PHP language and MySQL database system. The work consists of two sections. The first one includes basic knowledge about information systems, PHP language, MySQL database system and additional technologies used in the work. The second part describes the used database in UML language and the information system, its categorization by the level of user rights, security, and hereafter individual functions and scripts used in the work. The practical part of this work is the information system and its prototypical database enclosed on a CD.

## **Keywords**

Information system, HTML, CSS, PHP, MySQL, Database

## **Citace**

Petr Škarek: Informační systém cykloservisu, bakalářská práce, Brno, FIT VUT v Brně, 2007

# Informační systém cykloservisu

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Mgr. Tomáše Masopusta. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....  
Petr Škarek  
10.5.2007

## Poděkování

Tímto děkuji vedoucímu práce Mgr. Tomáši Masopustovi za odbornou pomoc, užitečné rady a informace k vypracování bakalářské práce.

© Petr Škarek, 2007.

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

# Obsah

Obsah .....	1
Úvod .....	3
1 Teorie a základní pojmy .....	4
1.1 Informační systém .....	4
1.2 Třívrstvá architektura .....	4
1.3 Databázový systém MySQL .....	4
1.4 Skriptovací jazyk PHP .....	5
1.4.1 Funkce include a require .....	5
1.4.2 Uživatelský vstup .....	6
1.4.3 Metody GET a POST .....	6
1.4.4 Kombinované skripty .....	6
1.4.5 Validace dat .....	7
1.5 Značkovací jazyk HTML .....	7
1.6 Kaskádové styly CSS .....	8
2 Analýza a specifikace požadavků .....	9
2.1 Klienti cykloservisu .....	9
2.2 Zaměstnanci cykloservisu .....	9
2.2.1 Zaměstnanci prodejny .....	9
2.2.2 Zaměstnanci servisu .....	10
2.2.3 Administrátoři systému .....	10
3 Návrh systému .....	11
3.1 Diagram případů užití .....	11
3.1.1 Klient cykloservisu .....	11
3.1.2 Zaměstnanec prodejny .....	12
3.1.3 Zaměstnanec servisu .....	13
3.1.4 Administrátor systému .....	14
3.1.5 Systém .....	15
3.2 Vytvoření struktury databáze .....	15

4	Implementace informačního systému .....	16
4.1	Neuautorizovaná část .....	16
4.1.1	Úvodní strana .....	16
4.1.2	Katalog zboží .....	16
4.1.3	Nejprodávanější zboží .....	17
4.1.4	Ověření stavu zakázky .....	17
4.1.5	Aktuality .....	17
4.1.6	Články .....	18
4.1.7	Přihlášení do autorizované části .....	18
4.2	Autorizovaná část .....	19
4.2.1	Zabezpečení .....	19
4.2.2	Rozdělení uživatelských rolí .....	20
4.2.3	Bezpečnostní záznamy informačního systému .....	21
4.2.4	Funkce informačního systému .....	22
4.3	Grafické uživatelské rozhraní .....	22
5	Závěr .....	23
	Literatura .....	24
	Seznam příloh .....	25

# Úvod

Informační systémy především webové se staly v posledních letech častým prezentačním a reklamním prostředkem nejen firem a škol, ale i různých institucí a soukromých osob. Stále narůstajícím zájmem o neomezené připojení k internetu i z domovů nabývají informační systémy čím dál většího významu a postupně pronikají do všech odvětví lidské činnosti.

Nejen z tohoto důvodu, ale i z mého zájmu o svět jízdních kol a jejich servis jsem si jako téma mé bakalářské práce vybral vytvoření webového informačního systému pro cykloservis. Po konzultaci a na přání majitele jednoho cykloservisu v Brně jsem vytvořil databázi a webový systém pro správu dat potřebných k zajištění chodu cykloservisu. Pro jeho tvorbu jsem použil skriptovací jazyk PHP a databázi MySQL. Grafická prezentace je založena na značkovacím jazyku HTML a CSS, díky tomu je systém přehledně zobrazitelný ve webovém prostředí pomocí většiny běžně používaných prohlížečů (úspěšně testováno na: IE 7, Firefox 2.0.0.3 a Opera 9.10).

Důraz byl kladen především na přehledné prostředí správy dat, snadnost ovládání jednotlivých funkcí systému, zabezpečení přístupu do systému, důsledné rozdělení uživatelských rolí administrátora, zaměstnance servisu a prodavače a v neposlední řadě také na ukládání bezpečnostních záznamů o dění uvnitř IS.

# 1 Teorie a základní pojmy

V této kapitole jsou popsány obecné pojmy informačních systémů, jejich třívrstvé architektury, skriptovacího jazyka PHP, databázového systému MySQL, značkovacího jazyka HTML a kaskádových stylů CSS.

## 1.1 Informační systém

Informační systém je systém sběru, uchování, analýzy a prezentace dat určených pro poskytování informací mnoha uživatelů různých profesí. Informace v těchto systémech jsou jen ta data, která dokážeme využít, přiřadit jim význam či smysl.

## 1.2 Třívrstvá architektura

Informační systém je postaven na modelu takzvané třívrstvé architektury. Základem aplikace je databázová vrstva, kterou tvoří databázový systém (anglicky Database management system, DBMS; česky systém řízení báze dat, SŘBD). Ten umožňuje přístup a zajišťuje integritu dat. V tomto případě se jedná o relační databázový systém MySQL.

Nad databázovou vrstvou je postavena složitá prostřední vrstva, která obsahuje většinu aplikační logiky a zajišťuje komunikaci mezi ostatními vrstvami. K jejímu vytvoření jsem použil skriptovací jazyk PHP.

Třetí vrstvou je vrstva klientská. Tu reprezentuje software internetového prohlížeče zajišťující vlastní interaktivní komunikaci uživatele s aplikací. Tento software zobrazuje zdroje informací ve formátu HTML, vydává požadavky HTTP na potřebná data a zpracovává odpovědi.

## 1.3 Databázový systém MySQL

Databázová vrstva je základem každé databázové webové aplikace. Pro úspěšný vývoj webové databázové aplikace musíme ze všeho nejdříve správně rozpoznat systémové požadavky, vybrat si vhodný software pro databázovou vrstvu, navrhnout strukturu databáze a celou vrstvu sestavit.

V aplikaci s třívrstvou architekturou zajišťuje databázová vrstva veškerou správu dat. Mezi její úkoly zpravidla patří ukládání, načítání a aktualizace dat. Databáze musí umožňovat současný neboli paralelní přístup více než jednoho procesu prostřední vrstvy, poskytovat potřebné zabezpečení,



zajišťovat integritu dat a provádět různé podpůrné operace, jako je například zálohování dat. Ve webových databázových aplikacích zajišťuje tyto služby relační databázový systém a data jsou uložena v relační databázi.

Pro správu relačních dat je potřeba dosti složitý software relačního databázového systému. Většina databázových systémů je ale naštěstí dobře navržena, takže veškeré jejich složité aspekty jsou skryty uvnitř. Pro efektivní práci s databázovým systémem stačí umět navrhnout databázi a zformulovat příkazy a dotazy vůči databázovému systému. Většina databázových systémů používá k dotazování jazyk SQL (Structured query language, česky strukturovaný dotazovací jazyk) – rozumět vnitřní architektuře databázového systému je pro většinu uživatelů zbytečné.

Jelikož celá moje databáze je postavena na relačním databázovém systému MySQL, popíši zde alespoň jeho základní vlastnosti. MySQL je systém střední velikosti ale dokáže spravovat i velmi velké objemy dat. Systém MySQL je znám svou rychlostí a je vhodný pro takové aplikace, kde je operace načítání častější než aktualizace, což je případ mého informačního systému. Další velkou výhodou MySQL je stejně jako u použitého skriptovacího jazyka PHP otevřený zdrojový kód.

## 1.4 Skriptovací jazyk PHP

PHP (rekurzivní akronym pro "PHP: Hypertext Preprocessor") je rozšířený univerzální skriptovací jazyk s volně dostupným zdrojovým kódem, který je obzvláště vhodný pro vývoj webových aplikací a lze jej zapouzdřit do HTML.

Podstatná část aplikační logiky třívrstvých webových databázových aplikací se skrývá v prostřední vrstvě, v tomto projektu reprezentované skripty jazyka PHP. Prostřední vrstva řídí strukturu a obsah dat zobrazovaných uživateli a zpracovává vstup od uživatele na jehož základě formuje dotazy pro čtení a zápis dat v databázi. Navíc musí do protokolu HTTP doplnit správu stavu. Aplikační logika prostřední vrstvy tak spojuje webové prostředí s databázovým systémem.

Jazyk PHP se zvláště dobře hodí pro vývoj webových databázových aplikací, neboť má k dispozici dobré nástroje pro integraci webového a databázového prostředí. Důležitá je zejména vysoká flexibilita spojená s možností vkládání skriptů do stránek HTML. Podpora integrace databázové vrstvy je zde také vynikající a PHP má k dispozici sadu knihoven pro komunikaci s téměř všemi používanými databázovými systémy.

### 1.4.1 Funkce include a require

Abychom mohli jednotlivé funkce využívat v různých skriptech a usnadnili si tak například ověřování stavu přihlášení uživatele či připojování k databázi, podporuje jazyk PHP příkaz include a direktivu require. Častěji užívanou funkci uložíme do speciálního souboru *funkce.php* jehož veškerý kód musí

být uzavřen počáteční a koncovou značkou skriptu. V samotném skriptu potom funkci zpřístupníme příkazem *include("funkce.php");*.

Pokud vyžadujeme, aby byl soubor do skriptu začleněn vždy – například ověření autentizace uživatele – použijeme direktivu *require*. Ta se totiž zpracovává ještě před vykonáním vlastního skriptu a její obsah se do skriptu začleňuje vždy.

## 1.4.2 Uživatelský vstup

Data, která mají ve skriptu řídit proces dotazování, můžeme předávat třemi různými způsoby.

- Ruční zadávání adresy URL, ve které se zadá nejen adresa pro načtení skriptu, ale i parametry.
- Zadávání dat pomocí formulářů jazyka HTML. Pomocí nich lze načíst textový vstup, nebo například reagovat na vstup vybráním výběrového seznamu nebo klepnutím na tlačítko.
- Skript PHP se může načíst a spustit včetně potřebných parametrů.

Nejčastěji se používají webové formuláře jazyka HTML a kliknutí na hypertextové odkazy. Oba tyto postupy jsou také použity v mém informačním systému.

## 1.4.3 Metody GET a POST

Uživatelská data neboli parametry se předávají z webového prohlížeče do webového serveru prostřednictvím protokolu HTTP. V tomto protokolu lze data předávat jednou ze dvou metod GET nebo POST.

Metoda GET data předává jako součást požadované adresy URL. Metoda GET tak načte zadaný prostředek(skript) přímo s udáním parametrů, které modifikují jeho podobu nebo způsob načtení.

U metody POST se data kódují odděleně od vlastní adresy URL a je umožněno odesílání většího objemu dat. Tato metoda se proto používá především pro zpracování dat z formulářů jazyka HTML, zatímco u vloženého odkazu lze použít pouze metodu GET.

## 1.4.4 Kombinované skripty

Formuláře i další obsah v kódu HTML a skripty v PHP pro jeho zpracování lze ukládat odděleně do samostatných souborů, nebo je implementovat do jediného skriptu, jehož kód může podle parametrů získaných metodami GET nebo POST vykonat požadovanou akci, tj. například vygenerovat formulář,

nebo vypsát požadovaná data na základě odeslaného formuláře. Kombinované skripty jsou velice užitečné a často používané prostředky pro tvorbu informačních systémů.

### 1.4.5 Validace dat

Validace neboli ověřování platnosti dat je ve webových databázových aplikacích nutností. Je důležité aby data vyhovovala požadavkům uživatele i systému a splňovala stanovená databázová omezení. Jedná se například o kontrolu typů načtených dat.

Provádět validaci lze v klientské vrstvě například pomocí JavaScriptu. Tato validace na jednu stranu umožňuje z pohledu uživatele velice rychlou kontrolu s nulovou režii komunikace po síti, na druhou stranu ji však lze obejít špatnou konfigurací nebo absencí podpory v použitém prohlížeči.

Validace na straně serveru je mnohem důležitější, umožňuje provádět úplnou a složitější validaci, kterou často nejde provést jinde než v prostřední vrstvě a také ji ze strany uživatele nelze obejít. Můj informační systém využívá především validaci na straně serveru.

## 1.5 Značkovací jazyk HTML

HTML(Hypertext Markup Language) je jazyk pro strukturování informací v aplikacích webového prohlížeče. Jazyk je podmnožinou dříve vyvinutého rozsáhlého univerzálního značkovacího jazyka SGML(*Standard Generalized Markup Language*). Vývoj HTML byl ovlivněn vývojem webových prohlížečů které zpětně ovlivňovaly definici jazyka.

Vývoj jazyka HTML byl ukončen verzí 4.01. Dalším pokračováním vývoje jazyků pro psaní dokumentů na webu je jazyk XHTML. Důvodů pro tuto změnu je celá řada. Překotný vývoj prohlížečů umožnil vznik obrovského množství dokumentů, které neodpovídají specifikaci HTML. Současné prohlížeče tolerují chybné značkování (neuzavřené, překřížené, či špatně vnořené elementy) a řadu dalších chyb a při parsování se s těmito chybami snaží vyrovnat („myslí“ za uživatele), zpravidla však ale ne dle představ tvůrce. Tento princip je ale příliš náročný pro nový druh aplikací, které vznikají pro malá méně výkonná zařízení - mobilní telefony, přenosné počítače (palmtopy) atd. Tyto aplikace potřebují přesně definovanou strukturu s přesnými pravidly. To poskytuje jazyk XML, ale HTML není aplikací XML a těmto novým požadavkům nevyhovuje. Pokud by však nové verze prohlížečů začaly vyžadovat přesně napsané HTML dokumenty bez chyb, pravděpodobně většina dokumentů na webu by nešla zobrazit.

## 1.6 Kaskádové styly CSS

CSS (Cascading Style Sheets) je jazyk pro popis způsobu zobrazení stránek napsaných v jazycích HTML, XHTML nebo XML. Jazyk byl navržen standardizační organizací W3C.

Hlavním smyslem je umožnit návrhářům oddělit vzhled dokumentu od jeho struktury a obsahu. Původně to měl umožnit už jazyk HTML, ale v důsledku nedostatečných standardů a konkurenčního boje výrobců prohlížečů se vyvinul jinak. Starší verze HTML obsahují celou řadu elementů, které nepopisují jen obsah a strukturu dokumentů, ale i způsob jeho zobrazení. Z hlediska zpracování dokumentů a vyhledávání informací však není takový vývoj žádoucí. Používání CSS ve srovnání se samotným HTML v praxi přináší spoustu výhod, pro které je jeho použití dnes již samozřejmostí.

CSS nabízí rozsáhlejší formátovací možnosti než samotné HTML. Např. pro formátování bloku textu – tj. určení vzdálenosti od jejich elementu či okraje stránky nenabízí HTML nic. CSS má vlastnosti padding a margin. V HTML by bylo potřeba vytvořit složitou konstrukci vnořených tabulek. Provést změnu stylu webu, který pro formátování vzhledu využívá jen možnosti HTML, znamená najít a nahradit všechny značky a změnit atributy mnoha dalších značek. V případě používání CSS znamená změna stylu webu přepsání jediného souboru – souboru stylů.

## **2 Analýza a specifikace požadavků**

Systém by měl poskytovat podporu dvěma základním skupinám uživatelů – klientům cykloservisu a zaměstnancům. Měl by být přehledný a snadno ovladatelný i pro počítačově méně zdatné uživatele.

Autorizovaná sekce, přístupná pouze zaměstnancům systému, by měla být odpovídajícím způsobem zabezpečena proti zneužití zvenku a i rozdělení uživatelských rolí uvnitř systému by mělo být dostatečně důsledné, aby předešlo neoprávněnému zneužití funkcí, které nejsou pro daného uživatele přístupné.

### **2.1 Klienti cykloservisu**

První skupina jsou klienti cykloservisu, kteří mají možnost si prohlížet pouze neautorizovanou sekci stránek. Tyto stránky budou mít čistě informační charakter, tzn. budou obsahovat především základní informace o cykloservisu, katalog a ceník zboží nabízeného cykloservisem k prodeji, statistiku nejprodávanějších komponentů, sekci aktualit a článků ze světa jízdních kol, soutěží a servisu. Dále umožní každému klientovi cykloservisu sledovat aktuální stav jeho zakázky a informují klienta o dokončení a celkové ceně zakázky a to na základě čísla zakázky, bez zobrazování osobních údajů.

### **2.2 Zaměstnanci cykloservisu**

Druhou základní skupinu tvoří zaměstnanci cykloservisu, kteří mají přístup do informačního systému samotného po ověření uživatelského jména a hesla. Do systému se přihlašují pomocí formuláře na úvodní straně.

Podle stupně uživatelských oprávnění rozlišuje systém tři typy uživatelů – zaměstnanců cykloservisu, a to Administrátora (vedoucího prodejny), zaměstnance servisu a zaměstnance prodejny. Pro každý typ účtu umožňuje systém využívání určitých funkcí.

#### **2.2.1 Zaměstnanci prodejny**

Zaměstnanec prodejny je základní typ uživatelského účtu. Umožňuje uživateli zobrazovat seznam klientů cykloservisu, přidávat nového klienta a editovat údaje o klientovi. Dále umožňuje zobrazovat seznamy dodavatelů, zboží a faktur prodejny a především prodej zboží a s tím spojené vystavování faktur. Při prodeji zboží systém automaticky generuje faktury, upravuje stav zboží ve skladu, statistiku prodejnosti a seznam prodaného zboží. Přihlášený uživatel informačního systému může

kdykoliv změnit svoje přihlašovací heslo a samozřejmě může – měl by svoji práci ukončit odhlášením ze systému. V případě dlouhodobější nečinnosti by měl být ze systému odhlášen automaticky.

## **2.2.2 Zaměstnanci servisu**

Druhým v pořadí je účet zaměstnance servisu, který rozšiřuje pravomoci zaměstnance prodejny o další funkce spojené se servisem kol a správou skladu. Seznam dodavatelů zboží je rozšířen o možnost přidat nového dodavatele. Seznam zboží pak o možnosti vytvoření nové kategorie zboží a přidávání a editování zboží. Novými funkcemi jsou sledování stavu skladu a automatické generování objednávky nedostatkového zboží, dále přidávání zboží do skladu, vystavování nových servisních zakázek, dokončování a fakturace hotových zakázek a zobrazování faktur servisu.

## **2.2.3 Administrátoři informačního systému**

Nejvyšší uživatelský účet umožňující využití všech dostupných funkcí informačního systému. Tento účet rozšiřuje možnosti uživatele o prohlížení seznamu uživatelů informačního systému, vytvoření nového uživatele a přidělení příslušných pravomocí, mazání položek veškerých evidovaných seznamů, správu jednoduchého redakčního systému, který umožňuje vkládání, editování a mazání aktualit a dlouhých článků včetně obrázků, zobrazovaných v neautorizované sekci webu cykloservisu. Dále umožňuje prohlížet detailní výpisy prodaného a nakoupeného zboží a bezpečnostních záznamů informačního systému o přihlášených uživateli a uskutečněných operacích uvnitř systému. Umožňuje také přístup k funkcím pro stornování faktury, vynulování statistiky prodejnosti, či smazání bezpečnostních záznamů a v neposlední řadě také k funkcím pro generování ucelených účetních záznamů za dané období i k jejich vymazání.

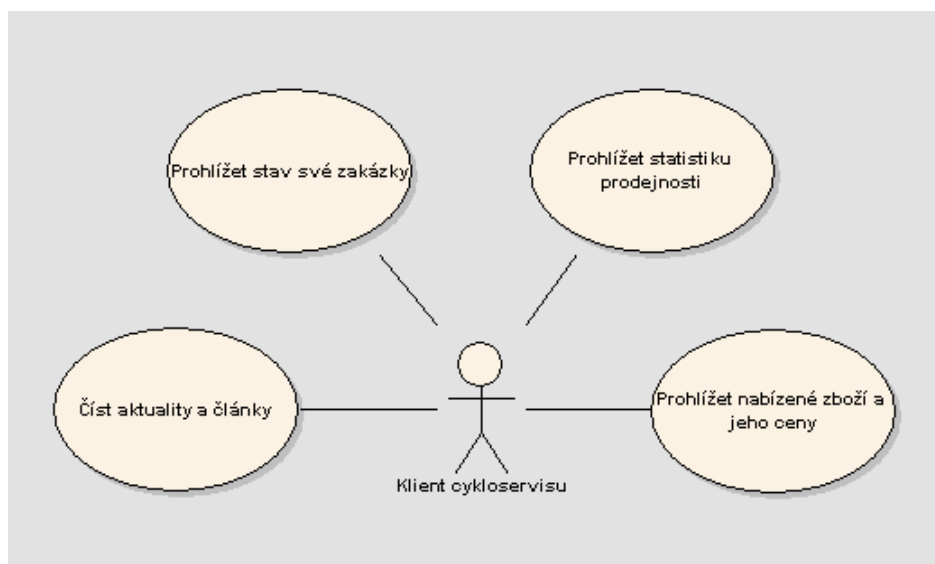
## 3 Návrh systému

Návrh je jednou z nejdůležitějších částí celé tvorby webového databázového informačního systému a měl by být prvním krokem při vývoji takového systému. Při správném navrhnutí systému se můžeme vyvarovat různým mnohdy velmi zásadním chybám při tvorbě programové části. Mnoho vývojářů návrh často podcení nebo dokonce vynechá, což se jim dříve nebo později vymstí například nutností předělávat již hotové části projektu z důvodu přidání opomenuté funkce.

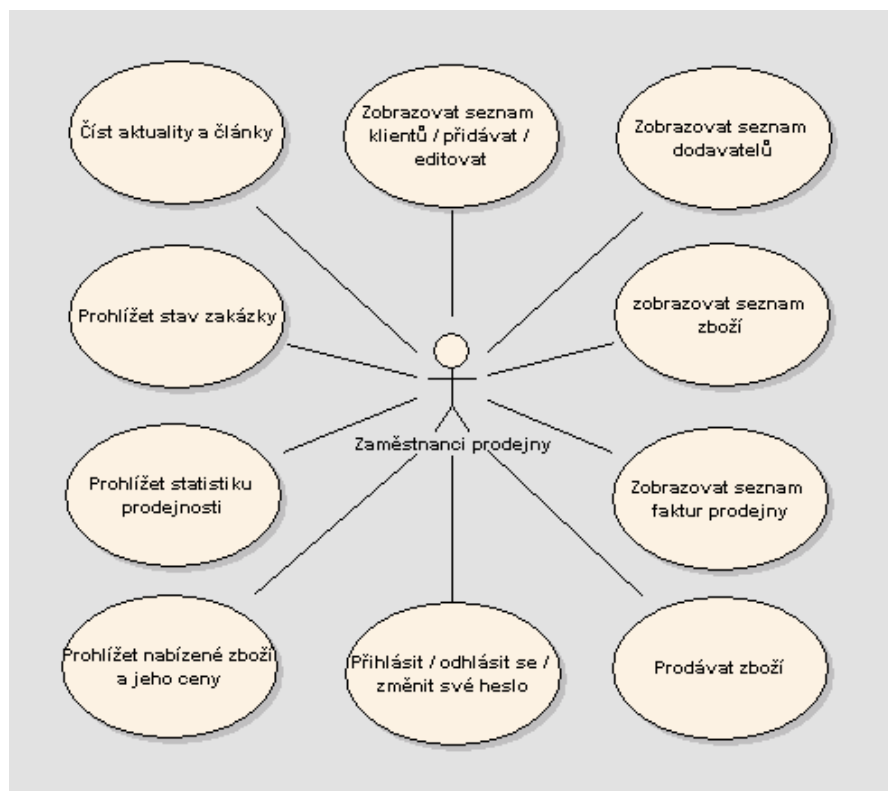
### 3.1 Diagram případů užití

Use case diagram, neboli diagram případů užití je grafický zápis reprezentující případy užití systému. Podle vytvořené analýzy a specifikace požadavků jsem vytvořil tyto diagramy případů užití.

#### 3.1.1 Klient cykloservisu

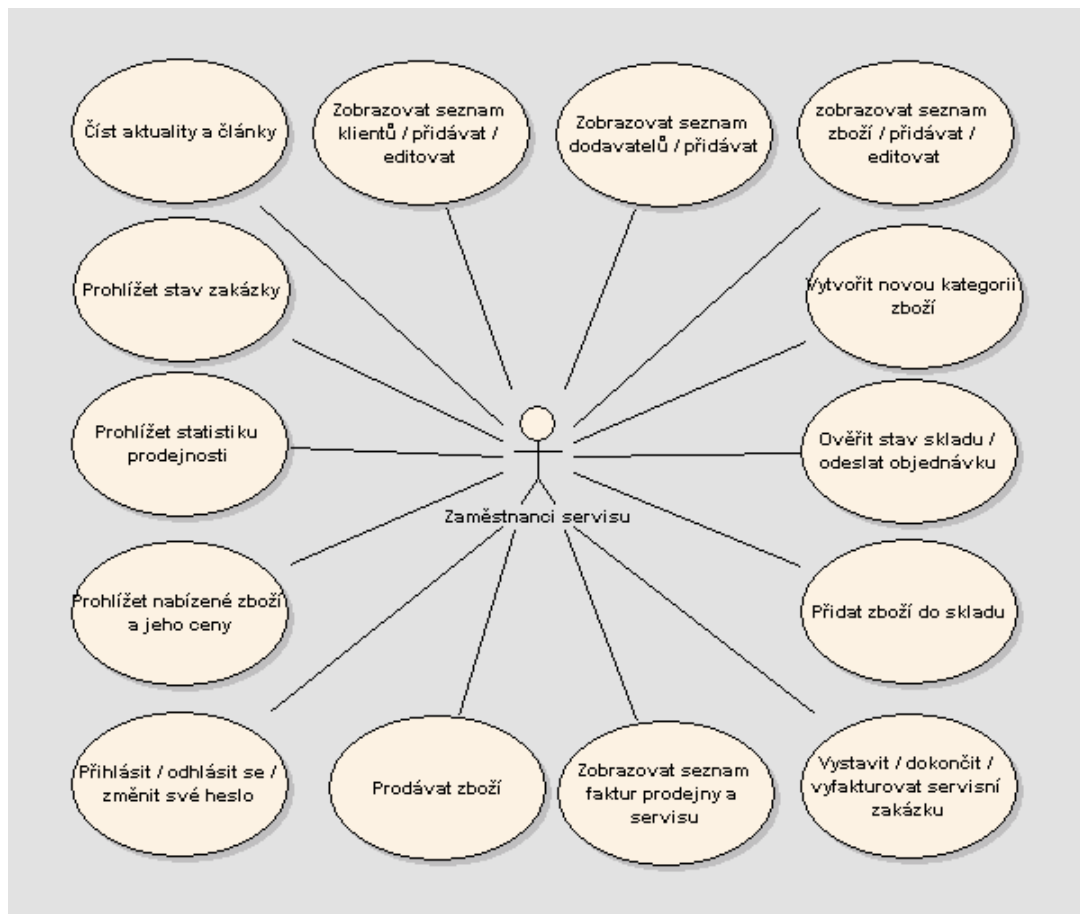


### 3.1.2 Zaměstnanec prodejny

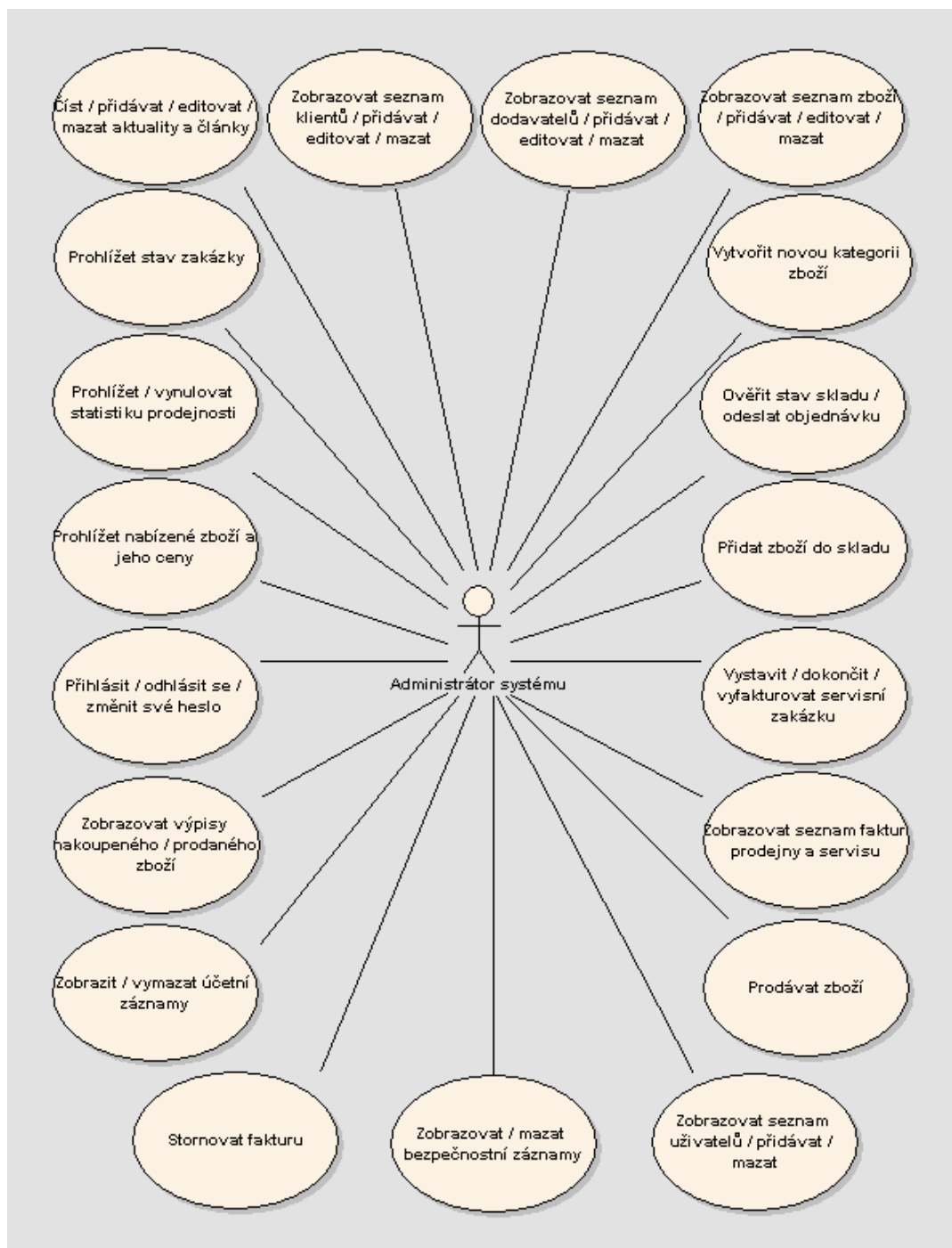




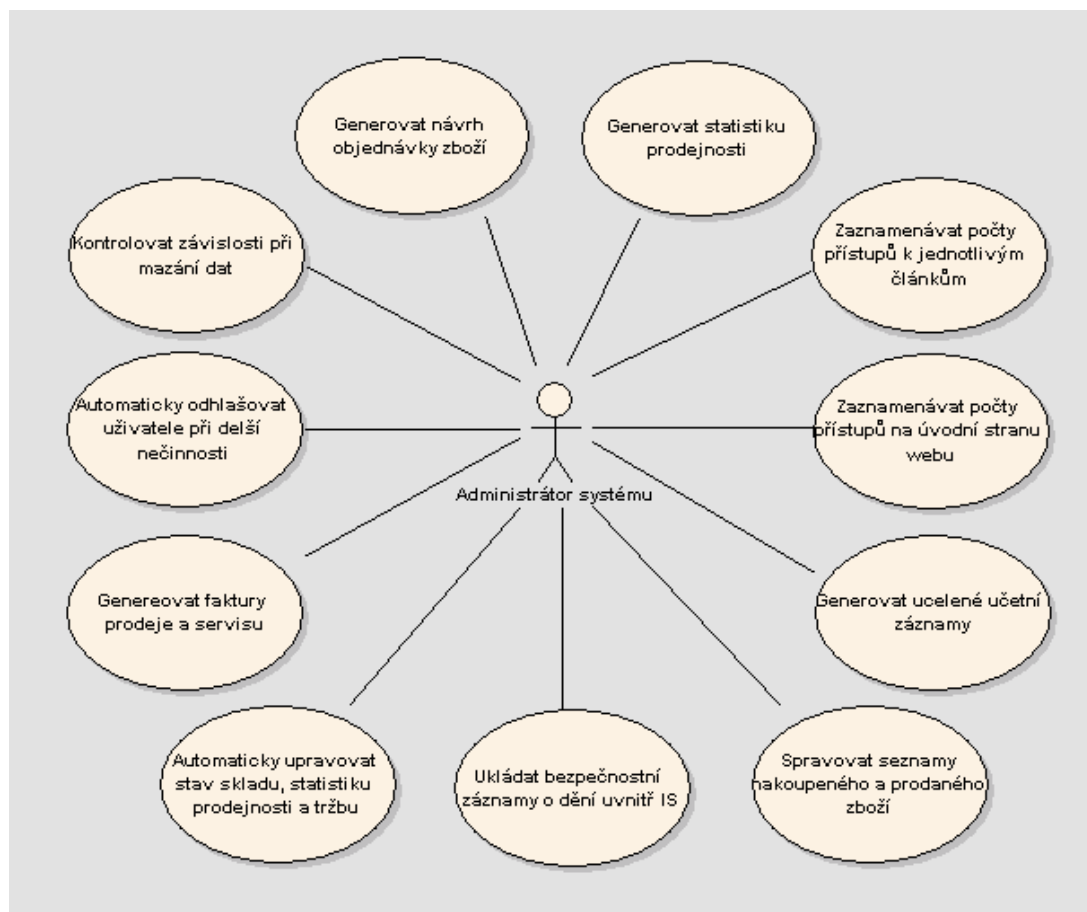
### 3.1.3 Zaměstnanec servisu



### 3.1.4 Administrátor systému



### 3.1.5 Systém



## 3.2 Vytvoření struktury databáze

K vytvoření struktury databáze jsem použil jazyk SQL a aplikaci PHP MyAdmin pro správu databázového systému MySQL.

Skript pro vytvoření struktury databáze tvoří přílohu A tohoto dokumentu.

## 4 Implementace informačního systému

K implementaci informačního systému jsem použil model tzv. třívrstvé architektury, databázový systém MySQL, značkovací jazyk HTML, kaskádové styly CSS a skriptovací jazyk PHP. O těchto technologiích pojednává kapitola 1 Teorie a základní pojmy na stranách 4-8 této práce.

### 4.1 Neautorizovaná část

Tato podkapitola se věnuje způsobu řešení neautorizované, tedy volně přístupné, části webových stránek cykloservisu, která má především informační charakter a slouží klientům cykloservisu k získávání informací o zboží, stavu jejich servisních zakázek a o dění ve světě cyklistiky a servisu.

#### 4.1.1 Úvodní strana

Úvodní strana webu cykloservisu sestává z několika základních oddílů, které jsou popsány v následujících částech této práce. Jedná se o katalog zboží, nabízeného cykloservisem, statistiku nejprodávanějšího zboží, kontaktní informace, aktuality a články. V hlavní – prostřední části úvodní strany se nacházejí úvody článků redakčního systému i s obrázky.

Převážná většina obsahu úvodní strany je systémem vygenerována z dat uložených v databázi. Čistě statická data tvoří pouze nadpis, formulář pro přihlášení do autorizované části informačního systému, kontakt na cykloservis a copyright.

#### 4.1.2 Katalog zboží

Katalog zboží je tvořen abecedně uspořádaným seznamem kategorií zboží, který je vygenerován podle seznamu kategorií uložených v databázi a umístěn v levé části stránky na absolutní pozici.

Po kliknutí na jednotlivé kategorie reprezentované hypertextovými odkazy dochází k zobrazení vygenerované tabulky obsahující data načtená z databáze pomocí dotazu k vypsání zboží podle kategorie. Tabulka se zobrazuje do prostřední části stránky na absolutní pozici. Tabulka obsahuje kromě údajů o zboží načtených z databáze i obrázek zboží, který je ve formátu JPEG uložen na serveru. Při kliknutí na obrázek dochází v novém okně, nebo panelu prohlížeče k jeho zobrazení v původní velikosti. Toto okno, nebo panel lze zavřít opětovným kliknutím na obrázek, nebo hypertextový odkaz pod obrázkem.

V případě, že zadaná kategorie neobsahuje žádné zboží, zobrazí se místo tabulky pouze textová informace, že kategorie neobsahuje žádné zboží. K tomuto stavu by však v praxi nemělo dojít, neboť zkušený prodejce jistě nedá do nabídky zboží kategorii, ze které nemůže klientovi nic nabídnout.

### 4.1.3 Nejprodávanější zboží

K zobrazení statistiky nejprodávanějšího zboží slouží hypertextový odkaz v levé části stránky pod katalogem zboží, zvýrazněný odlišnou barvou. Po kliknutí dochází v prostřední části stránky k vygenerování obdobné tabulky jako v případě katalogu zboží. V tabulce se však nacházejí ty komponenty, jejichž prodejnost je nejvyšší. Jejich seznam a prodané množství je načteno z databázové tabulky zboží, která k tomuto účelu obsahuje i sloupec s počtem prodaných kusů.

### 4.1.4 Ověření stavu zakázky

Jedním ze základních požadavků na informační systém bylo poskytnout klientovi cykloservisu možnost sledovat stav své servisní zakázky a informovat se na výslednou cenu a možnost vyzvednutí zakázky. Bylo požadováno řešení, které umožní jednoduše sledovat stav zakázek na základě přiděleného jednoznačného čísla zakázky, bez použití osobních údajů klienta.

Formulář *Moje zakázka* v levé části úvodní strany slouží k zadání čísla zakázky, které se nachází na klientově zakázkovém listu. Údaje na zakázkovém listu, kromě popisu zakázky, jsou generovány systémem bez možnosti zásahu uživatele a jedinečnost čísla zakázky je tak zaručena, neboť toto číslo tvoří primární klíč tabulky zakázek v databázi cykloservisu.

Po odeslání čísla pomocí formuláře systém zkontroluje, zda existuje v databázi záznam o zakázce s tímto číslem. Pokud je zakázka nalezena dochází v prostřední části stránky k vygenerování tabulky obsahující informace o stavu zakázky, případně termínu dokončení a celkové ceně.

Pokud příslušná zakázka není v databázi nalezena generuje systém namísto tabulky pouze chybové hlášení.

### 4.1.5 Aktuality

Aktuality, neboli bleskovky slouží k rychlému informování návštěvníků stránek o tom, že se něco děje, dělo, nebo bude dít. Například oznámení o změně otevírací doby příští týden je nejjednodušší realizovat pomocí aktuality na úvodní straně.

Sloupec aktualit tvoří pravou část všech stránek neautorizované sekce systému, neboli veřejně přístupné části webu cykloservisu. Jejich obsah je generován na základě tabulky aktualit v databázi

systému a maximální zobrazený počet, v pořadí od nejnovější po nestarší, je staticky omezen, aby byla zachována struktura a celistvost webu.

### **4.1.6 Články**

Jednou z požadovaných součástí informačního systému byl také jednoduchý redakční systém, poskytující přihlášeným uživatelům informačního systému přidávat, editovat a odstraňovat na webu cykloservisu krátké aktuality a delší články včetně obrázků, a to bez znalostí jazyka HTML či PHP. Tento redakční systém by měl na úvodní straně webu cykloservisu zobrazovat od každého článku vždy nadpis a úvod, případně obrázek, dále datum vydání článku, autora a počet přečtení článku.

V databázi informačního systému je k tomuto účelu zřízena tabulka článků, obsahující položky číslo článku, datum vydání, nadpis, úvod článku, text článku, autor, obrázek a přečtení. Ve sloupci obrázek je pomocí jedničky nebo nuly uložena informace o tom, zda daný článek obsahuje obrázek. Pokud ano, číslo článku udává číslo obrázku ve složce obrázků článků uloženého na serveru. Položka přečtení uchovává počet zobrazení celého textu článku.

Články se nacházejí v prostřední – hlavní části úvodní strany webu cykloservisu. Jednotlivé články tvoří buňky tabulky o jednom sloupci. Vzhled obsahu každé buňky je dán stylovým předpisem CSS a celá buňka je tak rozdělena na několik částí. Začíná nadpisem vlevo nahoře, pod ním se nachází informace o datu vydání, autorovi článku a počtu přečtení článku. Pokud je autor článku uložený v systému včetně emailové adresy, slouží položka jména autora jako hypertextový odkaz typu mailto:. Následuje odstavec úvodního textu, který tvoří plovoucí blok v levé části buňky. Vpravo od něj se nachází miniatura obrázku souvisejícího s příslušným článkem, která současně funguje jako hypertextový odkaz na zobrazení obrázku v původní velikosti. Poslední součástí vlevo dole pod textem úvodu je hypertextový odkaz na celý článek. Při kliknutí na tento odkaz dochází nejen k zobrazení celého článku, ale i k inkrementaci počítadla přečtení článku. Pro návrat zpět z detailu celého článku slouží hypertextový odkaz zpět na hlavní stranu, umístěný vlevo dole pod článkem.

### **4.1.7 Přihlášení do autorizované části**

K přihlášení do autorizované části informačního systému slouží formulář umístěný v levé horní části úvodní strany, do kterého je k úspěšnému přihlášení nutné zadat login a heslo a odeslat jej kliknutím na tlačítko přihlásit. V poli k zapsání hesla se z bezpečnostních důvodů místo zapsaných znaků objevují hvězdičky, či jiné zástupné znaky v závislosti na použitém webovém prohlížeči.

Způsob ověření – autorizace uživatele a jeho práv je podrobně rozveden v následující kapitole.

## 4.2 Autorizovaná část

Obsahem této podkapitoly je způsob řešení autorizované části webových stránek cykloservisu, tedy jádra samotného informačního systému. Především jeho zabezpečení, autorizaci i autentizaci jednotlivých uživatelů, ukládání bezpečnostních záznamů, rozdělení uživatelských rolí uvnitř systému a také všech jeho možností a funkcí ze strany přihlášeného uživatele.

### 4.2.1 Zabezpečení

V této části se budu věnovat použitému způsobu přihlašování, uložení údajů o uživateli systému, ověřování platnosti přihlášení uživatele a autentizaci uživatelů pro použití jednotlivých funkcí informačního systému.

#### 4.2.1.1 Přihlašování a zabezpečení přístupu pomocí seancí

Celý informační systém využívá zabezpečení přístupu pomocí seancí. Po odeslání uživatelského jména a hesla z výše popsaného formuláře na úvodní straně webu dochází k ověření platnosti uživatelského jména a hesla a IP adresy počítače (skript pro ověření přihlašovacích údajů tvoří přílohu B tohoto dokumentu) a k vytvoření nové seance. Do proměnných registrovaných v této seanci je uloženo uživatelské jméno přihlášeného uživatele a také IP adresa počítače, ze kterého se přihlásil.

Každý PHP skript uvnitř zabezpečené části informačního systému podléhá nutnosti ověření platnosti seance. K tomu je určen speciální PHP skript, který ověří platnost seance na základě uživatelského jména a IP adresy počítače. Tento skript je začleněn pomocí direktivy require jako první a nedílná součást všech zabezpečených skriptů. V případě úspěšného ověření pokračuje původní skript v další činnosti, v případě neúspěchu dojde ke zrušení aktuální seance a přesměrování na úvodní, resp. přihlašovací stranu.

Způsob zabezpečení autorizované části systému pomocí seancí také umožňuje velice snadné splnění požadavku na automatické odhlášení uživatele v případě delší nečinnosti. Konkrétní doba nečinnosti nutná k automatickému odhlášení je dána nastavením na straně serveru, v tomto případě Apache.

#### 4.2.1.2 Ověření platnosti uživatelského jména a hesla a IP adresy počítače

Před přihlášením do systému je nutné ověřit platnost přihlašovacích údajů a také umístění příslušného počítače v síti, neboť přístup do informačního systému je na přání majitele cykloservisu umožněn jen z konkrétní firemní podsítě.

Ke kontrole IP adresy je použita velmi jednoduchá ale účinná funkce jazyka PHP pro porovnání dvou textových řetězců (`if ( strcmp("192.168.0.10", $REMOTE_ADDR, 12) != 0 ) return false;`). Na základě této jednoduché podmínky vrátí funkce true jen v případě, že IP adresa uživatele počítače je z rozsahu 192.168.0.100 – 192.168.0.109.

Kontrola uživatelského jména a hesla je nepatrně složitější. Uživatelské jméno, heslo i práva uživatele jsou uloženy v databázové tabulce users. Uživatelské heslo je však z důvodu zvýšení bezpečnosti uloženo v jednosměrně zašifrované podobě pomocí standardní PHP funkce crypt(). K ověření uživatelského hesla je tedy nutné pomocí stejné funkce a s použitím stejného klíče zašifrovat heslo přijaté od uživatele a to potom srovnat s heslem uloženým v databázi. Pomocí SQL příkazu SELECT se zeptáme na výskyt řádku obsahujícího příslušné jméno a heslo v tabulce uživatelů. Pokud je nalezen právě jeden řádek, je ověření uživatele úspěšné.

V případě úspěšného ověření IP adresy i přihlašovacích údajů uživatele je vytvořena nová seance a uživatel přesměrován do zabezpečené části informačního systému.

## 4.2.2 Rozdělení uživatelských rolí

Jedním z požadavků na informační systém bylo důsledné rozdělení rolí jednotlivých uživatelů a jejich pravomocí. Z tohoto důvodu je v tabulce uživatelů v databázi informačního systému kromě uživatelského jména a hesla také sloupec s oprávněními jednotlivých uživatelů. Oprávnění jsou reprezentována číslicemi od jedné do tří přičemž číslice 1 značí práva administrátora systému – vedoucího prodejny, číslice 2 práva zaměstnance servisu – servisního technika a číslice 3 práva zaměstnance prodejny – prodavače.

Jednotlivé funkce informačního systému zobrazené v menu a proveditelné uživatelem, a to i v případě ručního zadání úplné adresy včetně všech parametrů, jsou přesně rozděleny podle rolí uživatelů. Ty funkce, které potřebují oprávnění 2 nebo 1, jsou zabezpečeny tak, aby je mohl vykonat ne každý přihlášený uživatel, ale jen přihlášený uživatel s adekvátním oprávněním, tzn. s oprávněním jehož číslice je rovna, nebo menší, než je pro danou operaci požadováno.

Stejným způsobem jsou na základě uživatelského oprávnění omezeny, resp. rozšířeny některé funkce o další vlastnosti jako například práce s položkami seznamu, který mohou zobrazit všichni, ale editovat položky vyžaduje oprávnění 2(nebo menší – viz výše) a mazat položky dokonce oprávnění 1.

### 4.2.2.1 Menu – nabídka funkcí informačního systému

Každý ze tří typů uživatelských účtů má k dispozici menu s nabídkou funkcí, které může daný uživatel využívat na základě svých uživatelských práv. Příslušné menu je vybráno po úspěšném přihlášení uživatele a následném načtení jeho oprávnění z tabulky uživatelů.



#### **4.2.2.2 Zabezpečení jednotlivých funkcí**

Každý zabezpečený skript informačního systému před svým vykonáním ověřuje oprávnění přihlášeného uživatele voláním funkce, uložené ve skriptu pro ověření platnosti seance, začleněného pomocí direktivy require(tento skript je popsán v kapitole 4.2.1.1).

Pokud by tak uživatel například zadal absolutní cestu včetně případných parametrů k funkci, která mu není určena, vrátí se mu pouze textová zpráva informující o tom, že pro přístup k požadované funkci nemá potřebná oprávnění.

### **4.2.3 Bezpečnostní záznamy**

Informační systém automaticky ukládá do databáze dva základní typy bezpečnostních záznamů.

#### **4.2.3.1 Přihlašování do informačního systému**

Prvním typem jsou záznamy o přihlašování uživatelů do systému. Systém vždy ukládá název příslušné operace, v tomto případě buď přihlášení, nebo neúspěšný pokus o přihlášení, dále datum, čas, uživatelské jméno zadané do přihlašovacího formuláře a IP adresu počítače.

#### **4.2.3.2 Operace v informačním systému**

Druhým základním typem záznamů jsou záznamy o proběhlých operacích, resp. použitých funkcích. Systém zaznamenává všechny operace, které jakýmkoliv způsobem zasahují do databáze. Neukládají se pouze nejjednodušší operace, jako například výpis seznamu zboží, které z databáze jen načítají data a nic nemění. Bezpečnostní záznam obsahuje podobně jako u záznamů přihlašování údaje o typu operace(např. *export účetních záznamů za období 2005-31-12 - 2007-04-23*), dále datum a čas, login uživatele, který operaci provedl a IP adresa počítače, ze kterého byla operace provedena.

#### **4.2.3.3 Přístup k bezpečnostním záznamům a jejich smazání**

Přístup ke čtení a mazání bezpečnostních záznamů informačního systému mají pouze uživatelé s oprávněním administrátor informačního systému.

K prohlížení bezpečnostních záznamů mají uživatelé v menu informačního systému kategorii Bezpečnostní záznamy, a v ní položky Přihlašování do IS a Operace v IS. Při kliknutí na jednotlivé položky následuje výběr období, za které chtějí bezpečnostní záznam prohlížet. Systém nabízí pro výběr období možnosti dnes, týden, měsíc, rok a celkem. Tato funkce pro výběr období je stejným způsobem použita i v mnoha dalších funkcích informačního systému, vypisujících záznamy s datem uskutečnění, jako například seznamy faktur, zakázek servisu, prodaného zboží atd.

Uživatelé s právy administrátora systému mají také možnost smazat všechny bezpečnostní záznamy systému. K tomuto účelu mají v menu položku Smazat bezpečnostní záznamy. Po kliknutí na tuto položku vyžaduje systém potvrzení smazání obsahu bezpečnostních záznamů. Pokud uživatel potvrdí svoji volbu dojde k jejich vymazání pomocí vyprázdnění(truncate) tabulky bezpečnostních záznamů. Ihned poté však dojde k zapsání bezpečnostního záznamu o vymazání tabulky bezpečnostních záznamů zpět do této tabulky. Takříkajíc ze strany uživatele není tedy z důvodu bezpečnosti možné uklidit zcela beze stopy.

#### **4.2.4 Funkce informačního systému**

Při implementaci funkcí informačního systému a jejich rozdělení podle uživatelských oprávnění jsem postupoval přesně podle Analýzy a specifikace požadavků(kapitola 2) a podle Návrhu systému(kapitola 3). Všechny implementované funkce odpovídají výše zmíněným požadavkům a návrhu systému a jejich vlastnosti byly do detailu doladěny pro potřeby cykloservisu.

### **4.3 Grafické uživatelské rozhraní**

Přehledné grafické uživatelské rozhraní je jistě jednou z nejdůležitějších součástí každého informačního systému.

Základními požadavky na informační systém cykloservisu byla také přehlednost a jednoduchost ovládání. Velkou výhodou pro každého vývojáře je případ, kdy klient má jasnou představu nejen o tom co by měl systém umět, ale i jak by měl vypadat. Odpadá tím totiž nutnost návrhu více grafických designů, ze kterých by si měl klient případně vybírat. K mému štěstí při vývoji tohoto informačního systému přispěl fakt, že majitel cykloservisu aktivně spolupracoval a měl zcela jasnou představu o vzhledu webu a dokonce i barevném provedení.

Při návrhu grafického prostředí byl kladen důraz na přehlednost, jednotnost a celistvost designu jednotlivých stránek informačního systému.

Kompletně celý vzhled informačního systému je vytvořen pomocí stylového předpisu CSS uloženého v jediném souboru. V samotných HTML souborech, resp. kombinovaných skriptech se prvky ovlivňující vzhled stránek vyskytují zcela ojediněle a pouze v opodstatněných případech.

## 5 Závěr

Rozvoj internetu stále více zasahuje do všech odvětví lidské činnosti, především v oblasti obchodu a služeb, které prostřednictvím internetu zviditelňují svojí nabídku a bojují tak se stále rostoucí konkurencí. Kvalitní prezentace nabízených služeb výrazně zvyšuje jejich prodejnost a známost v oboru. V době, kdy se internet rozšířil a stal se přístupný většině obyvatel této země, je nabídka služeb a reklama na internetu jedním z nezanedbatelných předpokladů úspěšného podnikání.

Tvorbou informačních systémů se v dnešní době zabývá velké množství firem a většina z nich je jistě schopna nabídnout kvalitněji propracované systémy než je výsledný produkt praktické části této bakalářské práce, nicméně většinou jejich cena začíná na několika desítkách tisíc korun. Vzhledem k tomu, že tento informační systém je zaměřen na klientelu z řad malých cykloservisů, které sice objemem prodaných kol nedosahují žádných převratných výsledků, i když v oblasti servisu patří ke špičce, je tímto také výrazně omezena cena, kterou jsou schopny a ochotny za něj nabídnout.

Informační systém byl implementován pomocí PHP a MySQL, které při dodržení určitých podmínek lze užívat zcela bezplatně. Skriptovací jazyk PHP a databázový systém MySQL byly vybrány také s ohledem na to, že jsou v dnešní době v praxi pravděpodobně nejrozšířenější a jejich znalost se mi bude v budoucnu jistě hodit.

Grafický design byl vytvořen s ohledem na maximální přehlednost a jednoduchost ovládání a také s ohledem na požadavky majitele cykloservisu, kterému bude tento systém sloužit. Myslím si, že ovládání systému je dostatečně intuitivní, neboť zaměstnanci cykloservisu, i ti méně počítačově zdatní, neměli po krátkém zaškolení žádné problémy s jeho základním ovládáním.

Při tvorbě informačního systému mi výrazně pomohly nejen náměty a vzorové implementace z dále uvedené literatury, ale také znalosti a zkušenosti z předchozích projektů a dalších domácích prací vytvořených při studiu předmětů zaměřených na webovou technologii a databázové a informační systémy.

Jako možná rozšíření do budoucna bych uvedl například rozšíření funkcí pro správu účetnictví, zavedení dalšího typu uživatele – účetní, rozšíření redakčního systému o možnosti hlasování či známkování článků, a v neposlední řadě také zvýšení abstrakce v celé aplikaci, které by vedlo k jejímu zobecnění a snadnějšímu upravení pro jakýkoliv druh podnikání, nejen cykloservis, nebo obecně servis.

# Literatura

- [1] Williams, H.E., Lane, D. *Programujeme webové aplikace pomocí PHP a MySQL*. Přeložil David Krásenský. 1. vyd. Praha: Computer Press, 2002, ISBN 80-7226-760-4.
  
- [2] Jak psát web – o tvorbě internetových stránek. [online], [2007-05-05]  
URL: <http://www.jakpsatweb.cz>
  
- [3] Wikipedie – otevřená encyklopedie. [online], [2007-05-05]  
URL: <http://cs.wikipedia.org>

# Seznam příloh

Příloha A: Skript pro vytvoření struktury databáze

Příloha B: Skript pro ověření přihlašovacích údajů a IP adresy

Příloha C: CD se zdrojovými kódy aplikace a elektronickou podobou textu této bakalářské práce

## Příloha A – skript pro vytvoření struktury databáze

```
DROP TABLE IF EXISTS `aktuality`;
CREATE TABLE `aktuality` (
  `id` int(5) NOT NULL auto_increment,
  `datum` date NOT NULL default '0000-00-00',
  `nadpis` varchar(50) NOT NULL default '',
  `text` text NOT NULL,
  PRIMARY KEY (`id`)
) TYPE=MyISAM;

DROP TABLE IF EXISTS `clanky`;
CREATE TABLE `clanky` (
  `id` int(5) NOT NULL auto_increment,
  `datum` date NOT NULL default '0000-00-00',
  `nadpis` varchar(100) NOT NULL default '',
  `uvod` text NOT NULL,
  `text` text NOT NULL,
  `autor` varchar(100) NOT NULL default '',
  `obrazek` tinyint(1) NOT NULL default '0',
  `precteni` int(11) NOT NULL default '0',
  PRIMARY KEY (`id`)
) TYPE=MyISAM COMMENT='clanky ze sveta cyklistiky';

DROP TABLE IF EXISTS `dodavatel`;
CREATE TABLE `dodavatel` (
  `id` int(5) NOT NULL auto_increment,
  `nazev` varchar(20) NOT NULL default '',
  `adresa` varchar(50) NOT NULL default '',
  `email` varchar(50) default NULL,
  `telefon` varchar(20) NOT NULL default '',
  `telefon2` varchar(20) default NULL,
  PRIMARY KEY (`id`)
) TYPE=MyISAM COMMENT='Tabulka dodavatelů';

DROP TABLE IF EXISTS `faktury`;
CREATE TABLE `faktury` (
  `id` int(6) NOT NULL auto_increment,
  `datum` date NOT NULL default '0000-00-00',
  `klient_id` int(5) NOT NULL default '0',
  `suma` int(6) NOT NULL default '0',
  `zakazka` int(5) default NULL,
  PRIMARY KEY (`id`)
) TYPE=MyISAM PACK_KEYS=0 COMMENT='Tabulka faktur';

DROP TABLE IF EXISTS `kategorie`;
CREATE TABLE `kategorie` (
  `id` int(5) NOT NULL auto_increment,
  `kat` varchar(20) NOT NULL default '',
  PRIMARY KEY (`id`)
) TYPE=MyISAM COMMENT='kategorie zboží';

DROP TABLE IF EXISTS `klienti`;
CREATE TABLE `klienti` (
  `id` int(5) NOT NULL auto_increment,
  `jmeno` varchar(20) NOT NULL default '',
  `prijmeni` varchar(20) NOT NULL default '',
  `adresa` varchar(50) NOT NULL default '',
  `email` varchar(50) default NULL,
  `telefon` varchar(20) default NULL,
  `utratil` int(6) NOT NULL default '0',
  `sleva` int(2) default NULL,
  PRIMARY KEY (`id`)
) TYPE=MyISAM COMMENT='Tabulka klientů cykloservisů';
```

```

DROP TABLE IF EXISTS `kosik_admin`;
CREATE TABLE `kosik_admin` (
  `zbozi` varchar(50) NOT NULL default '',
  `mnozstvi` int(5) NOT NULL default '0',
  `cena` int(5) NOT NULL default '0',
  `celkem` int(6) NOT NULL default '0'
) TYPE=MyISAM COMMENT='nakupni kosik';

DROP TABLE IF EXISTS `log_udalosti`;
CREATE TABLE `log_udalosti` (
  `operace` varchar(80) NOT NULL default '',
  `datum` datetime NOT NULL default '0000-00-00 00:00:00',
  `user` varchar(20) NOT NULL default '',
  `ip` varchar(16) NOT NULL default ''
) TYPE=MyISAM;

DROP TABLE IF EXISTS `nakoupene`;
CREATE TABLE `nakoupene` (
  `datum` date NOT NULL default '0000-00-00',
  `nazev` varchar(50) NOT NULL default '',
  `mnoz` varchar(5) NOT NULL default '',
  `suma` int(6) NOT NULL default '0',
  `dodavatel` varchar(20) NOT NULL default '0',
  `vlozil` varchar(20) NOT NULL default '',
  `ip` varchar(16) NOT NULL default ''
) TYPE=MyISAM COMMENT='seznam prodaného zboží';

DROP TABLE IF EXISTS `pristupy`;
CREATE TABLE `pristupy` (
  `pocet` int(10) NOT NULL default '0'
) TYPE=MyISAM COMMENT='pocitadlo pristupu k index.php';

DROP TABLE IF EXISTS `prodane`;
CREATE TABLE `prodane` (
  `datum` date NOT NULL default '0000-00-00',
  `nazev` varchar(50) NOT NULL default '',
  `mnoz` varchar(5) NOT NULL default '',
  `cena` int(6) NOT NULL default '0',
  `klient` int(5) NOT NULL default '0',
  `sleva` int(2) NOT NULL default '0',
  `vysl_cena` int(6) NOT NULL default '0',
  `prodavac` varchar(20) NOT NULL default '',
  `ip` varchar(16) NOT NULL default '',
  `faktura` int(6) NOT NULL default '0'
) TYPE=MyISAM COMMENT='seznam prodaného zboží';

DROP TABLE IF EXISTS `users`;
CREATE TABLE `users` (
  `user_name` varchar(20) NOT NULL default '',
  `password` varchar(20) NOT NULL default '',
  `rights` int(1) NOT NULL default '5',
  PRIMARY KEY (`user_name`)
) TYPE=MyISAM COMMENT='tabulka uzivatelu';
INSERT INTO `users` VALUES ('admin', 'adLOVfC6BVOeQ', 1);
-- admin/a

DROP TABLE IF EXISTS `zakazky`;
CREATE TABLE `zakazky` (
  `id` int(5) NOT NULL auto_increment,
  `prijata` date default NULL,
  `zakaznik` int(5) NOT NULL default '0',
  `popis` text NOT NULL,
  `stav` varchar(100) default NULL,
  `hotova` date default NULL,
  `cena` varchar(6) default NULL,
  `vyfakturovana` date default NULL,
  PRIMARY KEY (`id`)
) TYPE=MyISAM PACK_KEYS=0 COMMENT='tabulka zakázek';

```

```
DROP TABLE IF EXISTS `zbozi`;
CREATE TABLE `zbozi` (
  `id` int(5) NOT NULL auto_increment,
  `nazev` varchar(50) NOT NULL default '',
  `kat` varchar(20) NOT NULL default '',
  `subkat` varchar(20) default NULL,
  `dod` varchar(20) NOT NULL default '',
  `mnoz` int(5) NOT NULL default '0',
  `min_mnoz` int(5) NOT NULL default '0',
  `voc` int(5) NOT NULL default '0',
  `moc` int(5) NOT NULL default '0',
  `text` varchar(120) default NULL,
  `prodano` int(5) NOT NULL default '0',
  PRIMARY KEY (`id`)
) TYPE=MyISAM COMMENT='tabulka zbozi';
```



## Příloha B – skript pro ověření přihlašovacích údajů a IP adresy

```
<?php
function authenticateUser($connection, $username, $password, $ip)
{
    //kontrola IP adresy - povoleny rozsah 192.168.0.100-109
    if(strncmp("192.168.0.10", $ip, 12) != 0) return false;
    //kontrola přihlasovacich udaju
    if (!isset($username)||!isset($password)) return false;
    $key = substr($username, 0, 2);
    $crypted_password = crypt($password, $key);
    $authentication_query = "SELECT password FROM users WHERE user_name='$username'
AND password='$crypted_password'";
    $result = @ mysql_query($authentication_query, $connection) or showerror();
    if (mysql_num_rows($result)!=1) return false;
    else return true;
}
?>
```