

UNIVERZITA PARDUBICE

Fakulta elektrotechniky a informatiky

WWW aplikace s využitím relační databáze pro správu  
golfového klubu

Evžen Mynka

Bakalářská práce

2011

Univerzita Pardubice  
Fakulta elektrotechniky a informatiky  
Akademický rok: 2010/2011

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Evžen MYNKA**  
Osobní číslo: **I08120**  
Studijní program: **B2646 Informační technologie**  
Studijní obor: **Informační technologie**  
Název tématu: **Webový portál golfového klubu**  
Zadávající katedra: **Katedra informačních technologií**

### Z á s a d y p r o v y p r a c o v á n í :

Cílem práce je návrh a vytvoření webového portálu golfového klubu.

Obsah práce:

Analýza navrhovaného řešení, která bude obsahovat popis použitých technologií, návrh databáze a aplikačního řešení.

Popis problematiky transakčního zpracování a izolace transakcí v databázovém systému MySQL.

Návrh a realizace webového informačního systému pro usnadnění administrativních činností golfového klubu.

Pro realizaci aplikace bude použito: MySQL, PHP a HTML.

Rozsah grafických prací:

Rozsah pracovní zprávy:

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

Seznam odborné literatury:

**\*Castagnetto, J. a kol. Programujeme PHP profesionálně. Computer Press, 2004.**

**\*Oppel, A. Databáze bez předchozích znalostí. Computer Press, 2006.**

**\*Kofle, M. Mistrovství v MySQL 5. Computer Press, 2007.**

Vedoucí bakalářské práce:

**RNDr. Miroslav Benedikovič**  
Katedra softwarových technologií

Datum zadání bakalářské práce: **17. prosince 2010**

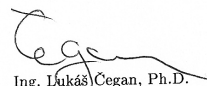
Termín odevzdání bakalářské práce: **13. května 2011**



prof. Ing. Simeon Karamazov, Dr.  
děkan



L.S.



Ing. Lukáš Čegan, Ph.D.  
vedoucí katedry

V Pardubicích dne 31. března 2011

## **Prohlášení autora**

Prohlašuji, že tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1. autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 12.5.2011

Evžen Mynka

## **Anotace**

Tato práce se zabývá tvorbou webové aplikace, která klade důraz na správně navržené a normalizované relační schéma databáze. Aplikace je zaměřená na správu golfového klubu a implementaci internetového obchodu. Tento obchod je zaměřen na golfové vybavení a potřeby. Samotná aplikace využívá databázový server MySQL a je napsána v jazyce PHP. Dále jsou zde využity technologie HTML, CSS, XML a databázovou vrstvu Dibi.

## **Klíčová slova**

Databáze, transakce, PHP, MySQL, Dibi, internetový obchod, golf

## **Title**

Creation of WWW application with usage relational database for administration of a golf club

## **Anotation**

This work deals with the composition of a web application which emphasises a correctly stated and normalised relational scheme of the database. The application is aimed for the administration of a golf club and implemetation of an e-shop. This shop focuses on a golf equipment and fittings. The applicationitself uses the database system MySQL and is written in the PHP language. Technologies HTML, CSS, XML and database layer Dibi are used as well.

## **Keywords**

Database, transaction, PHP, MySQL, Dibi, internet shop, golf

## **Poděkování**

Chtěl bych poděkovat RNDr. Miroslavu Benedikoviči za odborné vedení a pomoc při tvorbě této bakalářské práce.

# Obsah

<b>Seznam zkratk</b> .....	<b>4</b>
<b>Seznam obrázků</b> .....	<b>5</b>
<b>1 Úvod</b> .....	<b>6</b>
<b>2 Porovnání databázových systémů MySQL a Oracle</b> .....	<b>7</b>
2.1 MySQL .....	7
2.2 Oracle.....	7
2.3 Transakce a transakční zpracování .....	8
2.3.1 Řízení transakce.....	8
2.3.2 Izolace transakcí .....	9
2.4 Omezený počet záznamů.....	9
2.4.1 Limit v MySQL .....	9
2.4.2 Limit v Oracle.....	10
2.5 Auto increment .....	10
2.5.1 Auto increment v MySQL .....	10
2.5.2 Auto increment v Oracle.....	11
2.6 Závěrečné hodnocení .....	11
<b>3 Návrh projektu</b> .....	<b>12</b>
3.1 Role.....	12
3.2 Vzhled aplikace .....	12
3.3 UML activiti diagram .....	12
3.4 UML use case diagram .....	13
3.5 Rich picture diagram .....	13
<b>4 Databáze</b> .....	<b>14</b>
4.1 Tabulky.....	14
4.1.1 Uživatel.....	15
4.1.2 Adresa.....	15
4.1.3 Nástěnka .....	16
4.1.4 Role.....	16
4.1.5 Účet.....	16
4.1.6 Pm.....	17
4.1.7 Objednávka.....	17

4.1.8	Doprava .....	18
4.1.9	Turnaje.....	19
4.1.10	Koš_turnaje.....	19
4.1.11	Služby .....	19
4.1.12	Koš_služby .....	20
4.1.13	Produkty .....	20
4.1.14	Kategorie .....	21
4.1.15	Koš_produkty .....	21
4.1.16	Log.....	21
4.2	Indexy.....	22
4.3	E-R daigram.....	22
<b>5</b>	<b>Aplikace.....</b>	<b>23</b>
5.1	Použitý software .....	23
5.1.1	Toad Data Modeler 3.3.....	23
5.1.2	XAMPP .....	23
5.1.3	NetBeans IDE 6.9.....	24
5.1.4	HTTP klient .....	24
5.2	Použité technologie .....	25
5.2.1	HTML.....	25
5.2.2	CSS .....	25
5.2.3	PHP.....	25
5.2.4	Dibi.....	26
5.2.5	XML .....	26
5.3	Systém přístupu ke stránkám.....	26
5.4	Funkce .....	27
5.4.1	Registrace .....	27
5.4.2	Moje údaje .....	28
5.4.3	Turnaje.....	28
5.4.4	Služby .....	29
5.4.5	Produkty .....	29
5.4.6	Košík.....	30
5.4.7	Nástěnka .....	30
5.4.8	Zprávy.....	30



5.4.9	Účty .....	31
5.4.10	Objednávky.....	31
5.4.11	Logování do systému.....	31
5.4.12	Export do XML .....	31
5.5	Adresářová struktura.....	32
<b>6</b>	<b>Závěr .....</b>	<b>33</b>
	<b>Seznam použité literatury .....</b>	<b>34</b>
	<b>Příloha A – Vzhled webové stránky po přihlášení administrátora.....</b>	<b>35</b>
	<b>Příloha B – UML activity diagram .....</b>	<b>36</b>
	<b>Příloha C – UML use case diagram .....</b>	<b>37</b>
	<b>Příloha D – Rich Picture .....</b>	<b>38</b>
	<b>Příloha E – E-R diagram.....</b>	<b>39</b>
	<b>Příloha F – Adresářová struktura.....</b>	<b>40</b>

## Seznam zkratek

<b>CSS</b>	Cascade Style Sheets (Tabulky kaskádových stylů)
<b>E-R diagram</b>	Entity-Relationship diagram
<b>HTML</b>	HyperText Markup Language (Hypertextový značkovací jazyk)
<b>HTTP</b>	HyperText Transfer Protocol (Protokol pro výměnu hypertextových odkazů)
<b>MD5</b>	Message-Digest algorithm (Algoritmus šifrování dat)
<b>PHP</b>	Hypertext Preprocessor (Skriptovací jazyk pro tvorbu webových stránek)
<b>SGML</b>	Standard Generalized Markup Language (Univerzální značkovací jazyk)
<b>SHA1</b>	Secure Hash Algorithm (Algoritmus šifrování dat)
<b>SQL</b>	Structured Query Language (strukturovaný dotazovací jazyk)
<b>UML diagram</b>	Unified Modeling Language
<b>WYSIWYG</b>	What You See, Is What You Get (Způsob editace dokumentů)

## Seznam obrázků

Obrázek 1 – Limit v MySQL.....	10
Obrázek 2 – Limit v Oracle .....	10
Obrázek 3 – Auto_increment v MySQL .....	10
Obrázek 4 – Vytvoření sekvence.....	11
Obrázek 5 – Vytvoření trigru.....	11
Obrázek 6 – Tabulka Uživatel .....	15
Obrázek 7 – Tabulka Adresa .....	15
Obrázek 8 – Tabulka Nástěnka.....	16
Obrázek 9 – Tabulka Role .....	16
Obrázek 10 – Tabulka Účet .....	17
Obrázek 11 – Tabulka Pm .....	17
Obrázek 12 – Tabulka Objednávka .....	18
Obrázek 13 – Tabulka Doprava.....	18
Obrázek 14 – Tabulka Turnaje .....	19
Obrázek 15 – Tabulka Koš_turnaje .....	19
Obrázek 16 – Tabulka Služby .....	20
Obrázek 17 – Tabulka Koš_služby.....	20
Obrázek 18 – Tabulka Produkty .....	21
Obrázek 19 – Tabulka Kategorie.....	21
Obrázek 20 – Tabulka Koš_služby.....	21
Obrázek 21 – Tabulka Log .....	22
Obrázek 22 – Procentuální poměry využití webových serverů k březnu 2011 .....	24
Obrázek 23 – Procentuální poměry využití webových prohlížečů v březnu 2011 .....	24
Obrázek 24 – Systém přístupu ke stránkám .....	27
Obrázek 25 – Registrační formulář .....	28
Obrázek 26 – Formulář pro změnu osobních údajů .....	28
Obrázek 27 – Ukázka produktu v elektronickém obchodě.....	30
Obrázek 28 – Ukázka výpisu logů do systému.....	31

# 1 Úvod

Cílem této bakalářské práce je vytvořit internetovou aplikaci využívající relační databázi. Tato aplikace by měla být určena všem lidem, kteří se zajímají o golf. Měla by poskytnout základní informace o pravidlech hry, seznámit uživatele s herní etiketou, umožnit koupi potřebného vybavení a možnost pronajmutí některých služeb. To vše za pomoci jednoduchého a intuitivního ovládání. Dále by měla poskytnout možnost registrace do systému se správou osobních údajů a možnost komunikace s ostatními uživateli pomocí zpráv.

V první části této práce bude provedeno porovnání různých databázových systémů a bude popsán způsob transakčního zpracování.

V druhé části se provede analýza, na jejímž základě bude proveden návrh databázového schématu. Budou zde popsány tabulky, vztahy a další databázové objekty. Následně bude popsána implementace tohoto schématu do databáze MySQL.

Dalším krokem bude popsání použitých technologií při vývoji webové aplikace, výhody jejich implementace a budou popsány základní funkce aplikace.

V poslední části bude proveden popis konečného stavu aplikace a zhodnoceny výsledky práce.

## 2 Porovnání databázových systémů MySQL a Oracle

Před samotným začátkem jakékoliv práce na vývoji webové aplikace si musí každý určit, co vlastně od výsledné aplikace očekává a jaké nároky na ni bude mít. S tímto rozhodnutím je úzce spjat i výběr příslušného databázového systému. K dispozici je zde velká škála různých databázových systémů. Já se zaměřím jen na dva z nich, a to na MySQL a Oracle. Následně popíši jak společné vlastnosti těchto databázových systémů, tak i odlišnosti v řešení určitých problémů, které jsou spojeny s některými funkcemi a možnostmi, které nám tyto databáze poskytují.

### 2.1 MySQL

MySQL je databázový systém, který vytvořila švédská firma MySQL AB. Nyní je tento systém vlastněn firmou Sun Microsystems, dceřinou společností Oracle Corporation.

MySQL je nejpopulárnější open source<sup>1</sup> databázový systém na světě, a to zvláště díky jeho velkému výkonu, spolehlivosti a snadnému použití. Jedná se také o databázi používající novou generaci aplikací, stavěných na technologii LAMP, což zahrnuje Linux, Apache, MySQL a PHP

MySQL je multiplatformní databáze, tudíž umožňuje chod databáze na více platformách. V tomto konkrétním případě je jich více než dvacet a mezi nejrozšířenější a nejznámější patří například Linux, Windows, Mac OS a Solaris. Také zabezpečuje rozsáhlou možnost servisu a technické podpory. (2)

### 2.2 Oracle

Oracle je databázový systém firmy Oracle Corporation a oficiální název zní Oracle Database. Tato databáze je považována za světový top a poskytuje velmi robustní a propracovanou databázi. Aktuální verzí je Oracle Database 11g. Tento systém podporuje SQL, PL/SQL, objektové databáze atd. Oracle je poskytován jak pod komerční licenci, tak i pod licenci nekomerční, která je označována XE. Tato verze je ovšem částečně omezená jak hardwarově, tak i službami serveru na kterém běží.

Oracle se vyznačuje schopností zpracovat extrémně velký tok dat, zpracováním velkého počtu transakcí, velkou bezpečností a Multi-Version Read Consistency mechanismem, který zajišťuje, že nedochází ke vzájemnému blokování zápisových a čtecích operací.(4)

---

<sup>1</sup> Program s otevřeným zdrojovým kódem

## 2.3 Transakce a transakční zpracování

Transakce je ve světě databází velice důležitým pojmem, kterým se v dnešní době chlubí téměř každá databáze. Co to ale transakce jsou? "Transakční zpracování je obecný koncept, jehož cílem je zajistit integritu jakéhokoliv dynamického systému v rámci přechodu z jednoho konzistentního stavu do druhého. Konzistentním stavem z hlediska databázového systému může být stav, kdy máme v pořádku data."<sup>(4)</sup> Jinými slovy nám transakce umožňují provést více databázových příkazů jako jeden celek. Ukázkovým příkladem transakce je bankovní převod peněz, kdy se musí peníze odečíst z konta odesílatele a přičíst na konto příjemce. Tyto dvě operace se musí provést buď jako celek nebo se nesmí provést vůbec.

Transakce nám také hlídají konzistenci dat na úrovni čtení a zápisu. Konzistence čtení znamená, že uživatel získá data, která jsou konzistentní před spuštěním transakce, a dále s těmito daty v rámci dané transakce pracuje nezávisle na tom, zda jiná transakce tato data mezitím upravila. Jinak řečeno jsou pro danou transakci důležitá jen data platná před jejím začátkem. Tato ideologie ale s sebou přináší určitá úskalí, jako například, že dotaz vidí jen data platná před začátkem transakce, ale nevidí data, která byla změněná jím samotným. Tento problém může nastat při opakovaném volání jednoho příkazu `select`, ve kterém mezi tím došlo ke změně dat a s touto změnou by už bylo počítáno.<sup>(2)</sup>

### 2.3.1 Řízení transakce

Transakce se mohou řídit příkazy `commit` a `rollback`. Příkaz `commit` potvrdí danou transakci jako celek a příkaz `rollback` vrátí všechny změny, která daná transakce měla provést.

V jazyce MySQL dále ke startu transakce slouží příkaz *start transaction*. Pro vypnutí automatického potvrzování transakce slouží příkaz *set autocommit=0*. Toto vypnutí, ale není volba databáze, a tak platí jen v rámci daného spojení. Dále je potřeba dodat, že u MySQL jsou transakce podporovány pouze u tabulek typu InnoDB a BDB a při zadání jakéhokoliv příkazu je spuštěna transakce.<sup>(5)</sup>

U Oraclu se pro řízení transakce používají *save pionty*, které nám rozdělují transakci do více menších dílů a v případě chyby při vykonávání transakce se daná transakce odroluje na daný *save point*. Oracle nevyžaduje příslušný typ tabulky a podobně jako v MySQL je transakce opět spuštěna po zavolání jakéhokoliv příkazu nad databázovým objektem.

Některé příkazy ovšem nevyžadují příkaz `commit` pro potvrzení transakce a jsou potvrzovány automaticky. Neboli po provedení těchto příkazů nelze provedené změny odrolovat zpět. Mezi takové příkazy patří *create*, *alter*, *drop* a *truncate*.<sup>(2)</sup>

### 2.3.2 Izolace transakcí

Izolace transakcí má uživateli navodit představu, že na daném systému pracuje sám. To ovšem ve většině případů není pravda. Existují tři situace, které mohou nastat, když na systému pracují alespoň dva uživatelé:

- Dirty Reads(Špinavé čtení) – transakce čte data, která byla změněna jinou ještě nepotvrzenou transakcí.
- Nonrepeatable Reads(Neopakovatelné čtení) – Transakce opětovně čte čtená data a shledává, že tato data byla změněná jinou transakcí, která byla potvrzena během probíhání dané transakce
- Phantom Reads(Výskyt fantomů) – Transakce opakovaně spouští stejný dotaz vracející množinu výsledků a shledává, že jiná transakce, která byla během běhu dané transakce potvrzena, upravila danou množinu výsledků

Proto nám norma SQL92 udává čtyři úrovně izolace transakcí:

- Read uncommitted – může nastat špinavé čtení, neopakovatelné čtení, a výskyt fantomů.
- Read committed – nemůže nastat špinavé čtení, ale může nastat neopakovatelné čtení a výskyt fantomů.
- Repeatable read – nemůže nastat špinavé čtení a neopakovatelné čtení, ale mohou se vyskytnout fantomy.
- Serializable – nemůže nastat špinavé čtení, neopakovatelné čtení ani výskyt fantomů (2)

## 2.4 Omezený počet záznamů

Velmi často nastane situace, kdy nechceme z databáze vypsát všechny položky, ale jen určitý počet. Tento počet je vyjádřen limitem neboli počtem položek, které chceme vypsát. V praxi to může vypadat tak, že místo toho, aby se nám na stránku vypsalo 100 záznamů, jich budeme chtít vypsát pouze 10 a ostatní by měli být zobrazeny na dalších stránkách.

### 2.4.1 Limit v MySQL

MySQL umožňuje v syntaxi příkazu využít klauzuli LIMIT, která může mít buď jeden nebo dva parametry. Pokud obsahuje jen jeden parametr, tak nám tato hodnota říká, kolik prvků se má zobrazit na stránku. Pokud obsahuje dva parametry, tak první z nich udává, od kterého prvku se mají záznamy zobrazovat a druhý určuje kolik jich bud kolik

prvků se má zobrazit na stránku. Obecně platí, že při použití klauzule limit je potřeba použít i klauzuli ORDER BY, protože databázový server si hodnoty řadí podle sebe.(3)

```
SELECT * FROM tbl LIMIT 5;
```

Obrázek 1 – Limit v MySQL

#### 2.4.2 Limit v Oracle

Stejně jako v případě automatického incrementování ani zde Oracle nenabízí tak jednoduché a elegantní řešení. Oracle nemá klauzuli LIMIT, ale využívá sloupec ROWNUM, který nám udává číslo každého řádku. Konečný výpis je tvořen dvěma vnořenými selecty a podmínkou, která udává čísla ROWNUM a ty nám udávají hodnoty řádků, jenž se mají vypsát. Následující příklad ukazuje, jak je možné vypsát položky 51 až 60.

```
SELECT * from (SELECT rownum r_ ,id_aktuality,nadpis,datum,nick,obsah from a_aktuality) WHERE  
r_ BETWEEN (50) AND (60)
```

Obrázek 2 – Limit v Oracle

### 2.5 Auto increment

Velmi často nastane situace, kdy jako primární klíč v některé z tabulek používáme číselnou hodnotu. V tomto případě musíme docílit toho, aby každá tato hodnota byla jedinečná, což vyplývá z definice primárního klíče. Různé databáze nám poskytují odlišné možnosti jak tento problém řešit. Nejjednodušším způsobem je pravděpodobně automatické inkrementování této číselné hodnoty při každém vložení dat do databáze.

#### 2.5.1 Auto increment v MySQL

Databáze MySQL nám nabízí velmi jednoduchou funkci, která jak již bylo výše popsáno, nám při každém vložení dat do tabulky automaticky zvýší hodnotu primárního klíče. Tuto funkci si zapneme, když u sloupce, který má mít unikátní hodnotu uvedeme modifikátor AUTO\_INCREMENT. Například si vytvoříme tabulku Animals kde hodnota primárního klíče bude id. (3)

```
CREATE TABLE animals (  
    id MEDIUMINT NOT NULL AUTO_INCREMENT,  
    name CHAR(30) NOT NULL,  
    PRIMARY KEY (id)  
) type = InnoDB
```

Obrázek 3 – Auto\_increment v MySQL



## 2.5.2 Auto increment v Oracle

Oracle nám bohužel nenabízí tak jednoduché a rychlé řešení jako MySQL. Zde se tento problém musí řešit o něco komplikovaněji a zdlouhavěji, a to pomocí trigru a sekvencí.

Sekvence je databázový objekt, který se stará o automatické generování čísel, které jsou v určité relaci jako například inkrementace o jedničku. Tuto sekvenci je potřeba vytvořit pro každou hodnotu primárního klíče zvlášť. Dá se použít i jediná sekvence, ale tím se výrazně sníží přehlednost databáze. Mezi vlastnosti sekvence patří pamatování poslední vygenerované hodnoty a pamatování následující hodnoty. Tyto hodnoty získáme příkazy `NázevSekvence.CURRVAL` a `NázevSekvence.NEXTVAL`. Příklad vytvoření sekvence obj s minimální hodnotou rovnou 100 a inkrementací o jedna.

```
create sequence obj minvalue 100 increment by 1;
```

Obrázek 4 – Vytvoření sekvence

Trigger je databázový objekt, který se spustí za určité, předem stanovené podmínky. V tomto případě tvoří trigger nutnou součást, která je potřeba k vložení hodnoty ze sekvence do požadované tabulky jako hodnoty primárního klíče.

```
create or replace
trigger trigg_obj
before insert on objednavka for each row
begin
select obj.nextval into :new.id_objednavky from dual;
end;
```

Obrázek 5 – Vytvoření trigru

## 2.6 Závěrečné hodnocení

Základním cílem tohoto porovnání bylo zjistit výhody a nevýhody jednotlivých databází a rozhodnout o lepší z nich z hlediska implementace pro tuto práci. Oba systémy se ukázali jako velice podobné a práce v nich také. Obě databáze by se daly použít u většiny implementací webových aplikací, které mají zajišťovat rychlý a zabezpečený přístup. Z toho vyplývá, že pro tuto práci jsou oba systémy dostatečně vhodné a je jedno, který z nich se použije.

Ačkoliv se na Univerzitě Pardubice dva semestry vyučoval databázový systém Oracle, rozhodl jsem se pro MySQL. Tento databázový systém pro mě představoval výzvu naučit se něco nového o jiném databázovém systému a také poukázal na některé jednodušší syntaxe v zápisu jednotlivých příkazů.

### 3 Návrh projektu

Cílem tohoto projektu je vytvoření webové aplikace, která umožní správu golfového klubu. Základními vlastnostmi aplikace by mělo být informovat uživatele o tom co to golf je, jaká jsou pravidla a seznámit je s herní etiketou. Dále by měla poskytnout možnost registrace a rozlišení uživatelů dle jejich rolí a dle toho jim přiřadit práva. Aplikace by měla umožnit komunikaci uživatelů mezi sebou, možnost nákupu potřebného zboží a administraci databáze.

Vývoj proběhne pomocí jazyka PHP. Samotné testování vzhledu a funkčnosti aplikace pomocí internetových prohlížečů Opera a Mozilla Firefox.

#### 3.1 Role

V aplikaci by měli být tři základní role rozlišující pravomoci jednotlivých uživatelů:

- Neregistrovaný – Tento uživatel by měl být schopen pouze zjistit základní informace o klubu, hře, měl by mít možnost prohlídky nabízeného zboží a v neposlední řadě také možnost registrace
- Registrovaný – Tento uživatel by měl být schopen zjistit informace o hře, měl by mít možnost si upravit vlastní údaje, přihlašovat se na turnaje, objednávat služby a produkty a komunikovat s ostatními registrovanými uživateli.
- Administrátor – Toto by měl být hlavní uživatel, který by měl mít stejná práva jako registrovaný uživatel a navíc by měl mít na starost veškerou administraci a editaci vzkazů. Dále také práva na vypsání a rušení turnajů, zálohování databáze pomocí XML a úpravu uživatelských rolí a práv.

#### 3.2 Vzhled aplikace

Vzhled aplikace je přiložen v příloze A

#### 3.3 UML aktiviti diagram

UML aktiviti diagram je přiložen v příloze B

### **3.4 UML use case diagram**

UML use case diagram je přiložen v příloze C

### **3.5 Rich picture diagram**

Rich picture je přiložen v příloze D

## 4 Databáze

Toto je jedna z klíčových částí celého projektu, protože při špatném návrhu databáze mohou nastat obrovské problémy, a to ať v databázové nebo v aplikační části.

Dobrý návrh samotné databáze je základ každé aplikace. Při špatném návrhu se mohou vyskytnout dříve či později problémy, které si mohou následně vyžádat smazání celé databáze a vynutit si vytvoření zcela nového databázového návrhu a následné přepracování celé aplikační části.

Tato databáze by měla být navržena tak, aby splňovala třetí normální formu.

### 4.1 Tabulky

Tabulka je základním databázovým prvkem, který slouží k uložení informací do relační databáze. Každá databázová tabulka obsahuje pevně daný počet sloupců s předem definovaným významem. Teoreticky může každá tabulka obsahovat nekonečně mnoho řádků. V každé tabulce lze vytvořit řadu omezení, například si můžeme vybrat sloupec, který bude představovat primární klíč, vzdálený klíč, sloupec, který musí být vyplněn, typ záznamů nebo jeho velikost a řadu dalších. Aby mezi tabulkami mohly vzniknout nějaké vztahy, musíme mezi nimi vytvořit určité vazby a omezení. Tato omezení se nazývají kardinalita a parcialita.




Kardinalita nám udává, množstevní vztahy mezi prvky jednotlivých tabulek, neboli nám říká, že X řádků zdrojové tabulky nám může vstoupit do vztahu s Y řádky cílové tabulky. Obecně se rozlišují tři druhy kardinality:

- 1:1 - vztah, ve kterém na obou stranách vystupuje pouze jeden objekt dané entity (např. vztah manželé mezi entitou muž a entitou žena)
- 1:N – na jedné straně je jeden objekt, který může vstoupit do vztahu s jedním nebo více objekty na straně druhé (např. obchod a zboží)
- M:N – vztah, kde na obou stranách vstupuje více objektů (např. zákazník a zboží, kde jeden zákazník si může koupit více druhů zboží a jeden druh zboží si může koupit více zákazníků)

Parcialita nám udává povinnost respektive nepovinnost existence role příslušné entity vztahu. Jinak řečeno nám říká, zda záznam v tabulce A musí nebo jen může mít příslušný záznam v tabulce B.

#### 4.1.1 Uživatel


Tabulka uživatel je jednou ze základních tabulek v tomto schématu. Jako primární klíč je zde použita hodnota Nick, která představuje uživatelské jméno, pomocí kterého se uživatel přihlašuje do systému. Z toho vyplývá, že v systému nesmí být dva uživatelé se stejným nickem. Hodnotami cizích klíčů zde jsou Id\_role, které odkazuje do tabulky Role a Id\_adresy, které odkazuje do tabulky adresa. Dalšími hodnotami v tabulce jsou heslo, stav účtu, jméno a příjmení uživatele. Všechny tyto hodnoty mají omezení not null, čili nesmí být prázdné. Tato tabulka obsahuje vazby na tabulky Nástěnka, Role, Adresa, Objednávka, Účet a dvě vazby na tabulku Pm.

Uživatel		
 Nick	Varchar(20)	NN (PK)
Jmeno	Varchar(25)	NN
Prijmeni	Varchar(35)	NN
Heslo	Varchar(50)	NN
Stav_uctu	Int	NN
 Id_role	Int	NN (FK)
 Id_adresy	Int	NN (FK)

Obrázek 6 – Tabulka Uživatel

#### 4.1.2 Adresa

Jak už název tabulky napovídá, tak zde jsou uloženy informace o bydlišti a kontaktu registrovaného uživatele. Primárním klíčem je zde Id\_adresy, které je databází před každým vložením automaticky generován a inkrementován o jedničku. Jediným povinným údajem v této tabulce je e-mail, který uživatel musí vyplnit v registračním formuláři. Mezi ostatní nepovinné údaje patří město, ulice, PSČ, číslo popisné a telefon. Tato tabulka je vázaná na tabulku Uživatelé. Tato vazba umožňuje aby jeden záznam v tabulce Uživatelé odpovídal pouze jednomu záznamu v tabulce Adresa a jeden záznam v tabulce adresa odpovídal více záznamům v tabulce Uživatel.

Adresa		
 Id_adresy	Int	NN (PK)
Mesto	Varchar(20)	
PSC	Int(5)	
Ulice	Varchar(30)	
CP	Int(6)	
email	Varchar(40)	NN
Telefon	Int	

Obrázek 7 – Tabulka Adresa

### 4.1.3 Nástěnka

Tato tabulka slouží jako uložení vzkazů ostatním uživatelům. Vztah mezi tabulkou Uživatel a tabulkou Nástěnka je takový, že jeden řádek tabulky uživatel odpovídá více řádkům tabulky nástěnka a právě jeden řádek tabulky Nástěnka odpovídá právě jednomu záznamu v tabulce uživatel. Jako primární klíč zde slouží hodnota Id\_vzkazu. Tato hodnota je před každým vložením automaticky inkrementována. Jako vzdálený klíč je zde hodnota Nick, pomocí níž je vytvořena vazba na tabulku Uživatel. Další hodnoty v tabulce jsou text, čas a datum vložení vzkazu do tabulky. Žádný z těchto záznamů nesmí nabývat hodnoty null.<sup>2</sup>

Nastenka		
Id_vzkazu	Int	NN (PK)
Text	Varchar(1000)	NN
Cas	Time	NN
Datum	Date	NN
Nick	Varchar(20)	NN (FK)

Obrázek 8 – Tabulka Nástěnka

### 4.1.4 Role

Jedná se o tabulku, která rozděluje uživatele do dvou různých rolí. Těmi jsou buď registrovaný uživatel nebo administrátor. Primárním klíčem je zde použita hodnota Id\_role a opět je automaticky generována a inkrementována samotnou databází. Dalším záznamem je zde název role, který je omezen délkou dvaceti znaků. Vazba mezi touto tabulkou a tabulkou Uživatelé udává, že právě jeden řádek v tabulce Role odpovídá více řádkům v tabulce Uživatelé, a jeden řádek v tabulce uživatelé odpovídá jednomu záznamu v tabulce Role.

Role		
Id_role	Int	NN (PK)
Nazev	Varchar(20)	

Obrázek 9 – Tabulka Role

### 4.1.5 Účet

V této tabulce jsou uloženy informace o veškerých placených transakcích, které byly uskutečněny mezi uživatelem a databází. Jako primární klíč je zde použita hodnota Záznam, která je celočíselná a generuje ji databáze. Cizím klíčem je zde hodnota Nick, která se odkazuje na tabulku Uživatel. Ostatní položky tabulky jsou Datum vložení záznamu do tabulky, Částka představující cenu objednávky nebo výši vkladu a popis.

<sup>2</sup> Nesmí obsahovat prázdnou hodnotu.

Omezení v této tabulce zamezují, aby některá z hodnot obsahovala hodnotu null. Vazba mezi tabulkami Uživatel a Účet je typu 1:1. To znamená, že jeden uživatel vlastní jen jeden účet a jeden účet patří právě jednomu uživateli.

Účet		
🔑	Zaznam	Int NN (PK)
	Datum	Date NN
	Castka	Int NN
	Popis	Varchar(30) NN
🔑	Nick	Varchar(20) NN (FK)

Obrázek 10 – Tabulka Účet

#### 4.1.6 Pm

Pm je zkratka z anglického Personal Message, což znamená osobní zpráva. Tato tabulka poskytuje možnost posílat vzkazy mezi jednotlivými uživateli a komunikovat tedy jinak, než jen pomocí nástěnky. Primárním klíčem je zde Id\_pm a jak už je zvykem, je i tento řádek generován samotnou databází. Jsou zde dva cizí klíče a to Nick\_od a Nick\_pr. Oba tyto klíče poskytují vazbu na tabulku Uživatel. Důvod pro použití dvou totožných vazeb je jednoduchý a to abychom mohli identifikovat jak odesílatele, tak i příjemce dané zprávy. Jak už bylo zmíněno, tak obě tyto vazby jsou totožné a jsou ve vztahu 1:N. Každá z vazeb tedy umožňuje aby jeden uživatel měl více vzkazů, avšak jeden vzkaz měl pouze jednoho odesílatele a příjemce. Dále jsou zde další povinné hodnoty, které musí být vyplněny ještě před tím, než se vzkaz odešle. Hodnota Datum a čas značí datum a čas, kdy byl vzkaz vložen do databáze. Obě hodnoty se vygenerují z databáze a automaticky vloží, tudíž se o ně uživatel nemusí starat. Hodnota Přčteno je automaticky nastavována na hodnotu 0, což značí, že zpráva nebyla přečtená. Samotný text zprávy je pak omezen na 500 znaků a nesmí to být prázdný řetězec.

Pm		
🔑	Id_pm	Int NN (PK)
	Datum	Date NN
	Cas	Time NN
	Text	Varchar(500) NN
	Přčteno	Bool NN
🔑	Nick_od	Varchar(20) NN (FK)
🔑	Nick_pr	Varchar(20) NN (FK)

Obrázek 11 – Tabulka Pm

#### 4.1.7 Objednávka

Toto je tabulka, která slouží k evidenci objednávek, a to ať turnajů, služeb nebo produktů. Primárním klíčem je Id\_objednávky. Je to celočíselná hodnota generovaná

databázi. Jsou zde dva cizí klíče, které představují vazby na tabulky Uživatel a Doprava. Vztah mezi tabulkami Uživatel a Objednávka umožňuje, aby jeden uživatel měl více objednávek, a aby jednu objednávku mohl vlastnit pouze jeden uživatel. Vztah vůči tabulce Doprava umožňuje, aby jedna objednávka musela mít jeden typ dopravy a jeden druh dopravy mohlo mít více objednávek. Oba tyto vztahy jsou tudíž 1:N s rozdílem v kardinalitě. Zbývající záznamy v tabulce jsou Datum vzniku objednávky a Cena celkem, která se určí na základě vybraných položek ve všech třech košících, které budou popsány níže.

Objednávka		
Id_obj	Int	NN (PK)
Datum_vzniku_obj	Date	NN
Cena_celkem	Int	
Id_dopravy	Int(1)	NN (FK)
Nick	Varchar(20)	NN (FK)

Obrázek 12 – Tabulka Objednávka

#### 4.1.8 Doprava

Tato tabulka v sobě zahrnuje informace ohledně způsobu, kterým si uživatel přeje doručit jím požadované zboží. V základu jsou zde tři druhy dopravy, ze kterých si může uživatel vybrat. První z nich je osobní odběr, který je pochopitelně zcela zdarma. Druhou možností je poslání zásilky poštou, za což je účtován poplatek 100Kč. Posledním způsobem je doručení balíku pomocí kurýrní společnosti PPL. Tato možnost je zpoplatněna. Výše tohoto poplatku je 150Kč. Každý z těchto záznamů je označen primárním klíčem Id dopravy, který odkazuje na tabulku Objednávka. Vazba mezi tabulkami je ve vztahu 1:N kde jedna objednávka musí mít pouze jeden druh dopravy, kdežto jeden druh dopravy může být použit na více objednávek. Dalšími položkami v této tabulce jsou Cena dopravy a Popis způsobu dopravy. Pokud by někdo chtěl přidat do tabulky další položky, tak musí počítat s omezeními, které se vztahují na tyto sloupce a to, že cena musí být celočíselná a kratší než čtyři cifry a název dopravy musí být kratší než dvacet znaků.

Doprava		
Id_dopravy	Int(1)	NN (PK)
Nazev	Varchar(20)	
Cena	Int(3)	

Obrázek 13 – Tabulka Doprava



#### 4.1.9 Turnaje

V této tabulce jsou uchována všechna data ohledně plánovaných a konaných turnajů. Jako první zde najdeme hodnotu primárního klíče Id\_turnaje. Tato hodnota je opět generována databází. Mezi další položky zde patří Název turnaje, Popis turnaje, Cenu za startovné a Datum konání turnaje. Položky Název a Popis jsou opět omezeny maximální délkou 50 a 500 znaků. Tato tabulka má nepřímou vazbu s tabulkou Objednávky. Vazba je nepřímá z důvodu vazby M:N. V těchto případech se využívá pomocné tabulky. V tomto případě se jedná o tabulku Kos\_turnaje, která je popsána níže.

Turnaje			
Id_turnaje	Int	NN	(PK)
Nazev	Varchar(50)		
Popis	Varchar(500)		
Startovne	Int		
Datum	Date		

Obrázek 14 – Tabulka Turnaje

#### 4.1.10 Koš\_turnaje

Jak už bylo zmíněno výše, jedná se o pomocnou tabulku ve vazbě typu M:N mezi tabulkami Objednávka a Turnaje. Atributy této tabulky jsou pouze dvě hodnoty, které představují primární a zároveň cizí klíče.

Koš_turnaje			
Id_obj	Int	NN	(PFK)
Id_turnaje	Int	NN	(PFK)

Obrázek 15 – Tabulka Koš\_turnaje

#### 4.1.11 Služby

Zde jsou uvedené informace o poskytovaných službách na tomto portálu. Každá služba obsahuje svoje id, název, popis služby a cenu. Id\_služby zde představuje primární klíč generovaný databází. Hodnoty Název a Text jsou textové řetězce, které jsou omezeny počtem znaků a to 50 a 300. Tato tabulka je nepřímo napojena na tabulku Objednávky a tvoří vazbu M:N, takže je pro jejich vzájemné propojení opět použita pomocná tabulka.

Sluzby			
Id_sluzby	Int	NN (PK)	
Nazev	Varchar(50)		
Text	Varchar(300)		
Cena	Int		

Obrázek 16 – Tabulka Služby

#### 4.1.12 Koš\_služby

Obdobně jako tabulka Koš\_turnaje je i toto pomocná tabulka ve vazbě M:N a to mezi tabulkami Objednávka a Služby. Tato tabulka má ovšem ještě další atributy jako jsou například datum a čas, na kdy byla daná služba rezervována a počet, který určuje buď počet kusů dané služby nebo počet hodin rezervace této služby. Hodnoty těchto atributů nesmějí nabývat hodnoty null. Primární a vzdálené klíče zde představují hodnoty Id\_obj a Id\_služby.



Kos_sluzby			
Id_obj	Int	NN (PFK)	
Id_sluzby	Int	NN (PFK)	
Datum	Date	NN	
Cas	Time	NN	
Pocet	Int(2)	NN	

Obrázek 17 – Tabulka Koš\_služby

#### 4.1.13 Produkty

Tato tabulka slouží pro evidenci veškerých produktů, které jsou nabízeny v e-shopu<sup>3</sup>. Produkty jsou rozděleny do několika kategorií, které jsou blíže popsány v tabulce Kategorie a vazba mezi těmito tabulkami je typu 1:N, kde jeden výrobek může být pouze v jedné kategorii a v jedné kategorii může být více výrobků. Spojení mezi těmito tabulkami je provedeno pomocí hodnoty cizího klíče Id\_kategorie. Primární klíč je tvořen sloupcem Id\_produkta. V této tabulce dále najdeme hodnoty Název, který reprezentuje název produktu, Text, který blíže popisuje daný produkt, Cenu produktu a URL adresu, která udává umístění obrázku k příslušnému produktu. Vazba této tabulky s tabulkou Objednávka je typu M:N a je opět provedená pomocí pomocné tabulky. Veškeré další omezení hodnot této tabulky najdete na obrázku, který je uveden níže.


<sup>3</sup> Do češtiny se překládá jako internetový obchod.

Produkty		
 Id_produkту	Int	NN (PK)
Nazev	Varchar(20)	
Text	Varchar(500)	
Cena	Int(5)	
URL	Varchar(200)	
 Id_kategorie	Int	NN (FK)

Obrázek 18 – Tabulka Produkty

#### 4.1.14 Kategorie



Tato tabulka popisuje jednotlivé kategorie, pod které spadají dané produkty. Primárním klíčem je zde sloupec Id\_kategorie, který je generován databází. Dále je zde hodnota název, která odpovídá názvu jednotlivých kategorií. Vazba mezi touto tabulkou a tabulkou Produkty byla popsána výše v podkapitole Produkty.

Kategorie		
 Id_kategorie	Int	NN (PK)
Nazev	Varchar(30)	

Obrázek 19 – Tabulka Kategorie

#### 4.1.15 Koš\_produkty

Jak již bylo výše uvedeno, tak tato tabulka slouží jako pomocná tabulka ve vztahu M:N mezi tabulkami Objednávka a Produkty. Jako primární a zároveň cizí klíče zde jsou použity hodnoty Id\_obj a Id\_produkty. Poslední hodnotou v této tabulce je hodnota Počet, která udává počet výrobků, které si zákazník objednal.

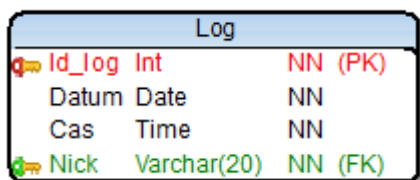
Koš_produkty		
 Id_obj	Int	NN (PFK)
 Id_produkту	Int	NN (PFK)
Pocet	Int(2)	NN

Obrázek 20 – Tabulka Koš\_služby

#### 4.1.16 Log

Tato tabulka je dostupná jen pro administrátora a slouží pro sledování aktivity uživatelů na webu. Pokaždé když se některý z uživatelů přihlásí, respektive odhlásí ze systému, je do této tabulky uložen záznam o datu a čase této aktivity a nick daného uživatele. Primárním klíčem je zde hodnota Id\_log, kterou generuje databáze a jako

vzdálený klíč je použitá hodnota Nick. Další hodnoty jsou Datum a čas a nesmí obsahovat hodnotu null. Vazba mezi touto tabulkou a tabulkou Uživatele je typu 1:N, kde jeden uživatel může mít více záznamů v této tabulce a jeden záznam musí odpovídat pouze jednomu uživateli.



Log		
Id_log	Int	NN (PK)
Datum	Date	NN
Cas	Time	NN
Nick	Varchar(20)	NN (FK)

Obrázek 21 – Tabulka Log

## 4.2 Indexy

Index je databázová konstrukce, která pomáhá zrychlit vyhledávání. Indexy jsou buď implicitní nebo explicitní. Implicitní indexy se vytvoří automaticky nad sloupci, které představují primární nebo cizí klíč. Explicitní indexy si člověk musí vytvořit sám a vytvářejí se nad sloupci, podle nichž se bude často vyhledávat. Mohlo by se zdát, že čím více indexů se použije, tím je to pro databázi lepší, ale opak je pravdou. Každý index totiž zabírá v paměti místo, které je vyhrazeno pro databázi. Navíc každý index zpomaluje operace, které mění obsah indexovaných sloupců. Proto si před vytvořením každého indexu musíme říct, zda je tento index opravdu potřeba a zda nebude páchat více škody než užitku.

V této aplikaci jsem se většinou spokojil s implicitními indexy nad primárními a cizími klíči, které mi vytvořila databáze. Výjimkou je jeden index, který jsem vytvořil u tabulky PRODUKTY nad sloupcem CENA. Tento index jsem vytvořil proto, že se produkty v aplikaci řadí dle ceny a to vzestupně od nejnižší až po nejvyšší a také je zde předpoklad k velkému počtu položek v tabulce.

## 4.3 E-R diagram

E-R diagram je přiložen v příloze E.

## 5 Aplikace

Tato část práce se bude zabývat samotnou implementační částí celé aplikace. Budou zde vysvětleny hlavní funkce aplikace, popsán software a technologie, které byly použity při vývoji. Také bude zmíněn systém přístupu k jednotlivým stránkám.

### 5.1 Použitý software

Veškerý software, který byl při vývoji použit spadá do skupiny open source. Jinými slovy se dá říct, že na pořízení nebo následné provozování tohoto softwaru nejsou potřeba žádné finanční náklady. Jedinou výjimku tvoří operační systém, kterým v tomto případě byl 32bitový Windows 7 Professional.

#### 5.1.1 Toad Data Modeler 3.3

Samotný návrh struktury databáze proběhl v programu Toad Data Modeler 3.3. Jedná se o profesionální program pro návrh databázových modelů. Tento program také umožňuje následné vygenerování SQL kódu pro vytvoření navržených objektů v samotné databázi. Samotná práce s tímto programem je velice intuitivní a jednoduchá, takže si uživatel na tento program velice rychle zvykne.

#### 5.1.2 XAMPP

Dalším nezbytným programem byl XAMPP. Jedná se o programový balíček, který vám velmi rychle a snadno nainstaluje současně Apache, MySQL a PHP. Tímto způsobem se lehce vyhnete zdlouhavému instalování a nastavování každého tohoto produktu zvlášť.

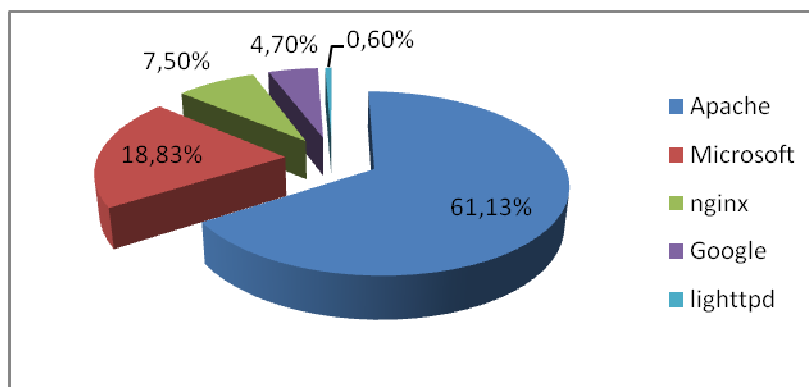
#### MySQL

Pro správu databáze bylo použito prostředí phpMyAdmin, které je součástí nainstalovaného balíčku XAMPP. Jedná se o grafické prostředí pro správu MySQL databáze, které administrátorovi poskytuje rozsáhlé možnosti pro správu databáze.

#### HTTP server

Jak už bylo zmíněno výše, tak za http server neboli webový server byl zvolen server Apache. Jedná se o celosvětově nejrozšířenější webový server. Pro potřeby této aplikace nemusela být defaultní konfigurace serveru nijak upravována. O Apachi obecně platí, že je to velice stabilní, rychlý a jednoduše konfigurovatelný webový server, jehož hlavní předností je to, že je jeho používání zcela zdarma.

Poměr využití webových serverů dle news.netcraft.com k březnu 2011 můžeme vidět na obrázku č. 21. (7)



Obrázek 22 – Procentuální poměry využití webových serverů k březnu 2011

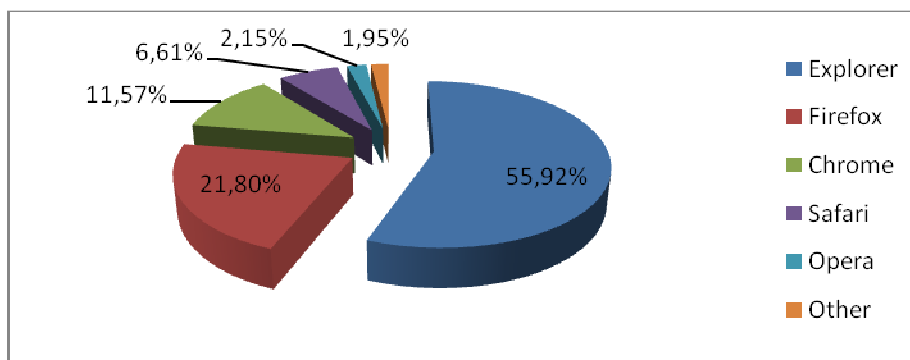
### 5.1.3 NetBeans IDE 6.9

Samotné psaní kódu proběhlo v programu NetBeans IDE 6.9. Toto vývojové prostředí podstatně ulehčuje psaní kódu díky jeho barevnému zvýraznění, označování chybné syntaxe, doplňování syntaxe, snadnému formátování kódu a spoustě dalších užitečných vlastností. Další důvod pro volbu tohoto softwaru byla skutečnost, že s tímto programem už mám určité zkušenosti a jsem zvyklý na jeho grafické prostředí a poskytované funkce.

### 5.1.4 HTTP klient

Funkčnost a design byly ověřovány v internetových prohlížečích Opera a to ve verzích 10 a 11.01 a Mozilla Firefox ve verzích 3.6 a 4. Použití dvou internetových prohlížečů bylo zvoleno z důvodu sjednocení designu aplikace v různých internetových prohlížečích.

Následující obrázek ukazuje podíl tržního rozdělení internetových prohlížečů dle techspot.com/news k dubnu 2011 (8)



Obrázek 23 – Procentuální poměry využití webových prohlížečů v březnu 2011

## 5.2 Použité technologie

V této kapitole si přiblížíme technologie, které byly použity při vývoji aplikace. Budou popsány jejich základní vlastnosti, principy a výhody použití, popřípadě jejich technologické slabiny.

### 5.2.1 HTML

Základním kamenem celé aplikace je značkovací jazyk HTML (HyperText Markup Language). Jedná se o jazyk, který slouží pro vývoj webových stránek a jeho kořeny sahají k jazyku SGML. Historie jazyka HTML sahá až do roku 1991 a verze 0.9. Od té doby tento jazyk prošel řadou změn a poslední významná změna byla provedena v roce 1999, kdy vznikla verze 4.1. Nyní se pracuje na verzi 5, avšak její dokončení odhaduje až na rok 2022.

Samotný jazyk obsahuje základní grafické prvky, jako jsou tabulky, odkazy, seznamy atd. Jazyk pracuje na straně klienta, a tak je v něm možno vytvořit pouze statické stránky.

### 5.2.2 CSS

Jedná se o kaskádové styly, které pomáhají k tvorbě designu webových stránek. Tyto styly nám dovolují uchovávat předem definované styly v externím souboru a používat je všude tam, kde je potřeba formátovat vzhled stránek. Důvodů pro použití CSS (Cascading Style Sheets) je celá řada, od rychlejšího zobrazování stránek, přehlednosti kódu, odpadnutí nutnosti zasahovat do HTML po možnosti spravovat vše z jednoho místa a mnoho dalších. CSS nám vlastně umožňuje nadefinovat jeden styl a použít ho všude tam, kde je potřeba, bez nutnosti tento styl definovat při každém použití. Tímto nám vzniká přehlednější a strukturovanější kód.

### 5.2.3 PHP

PHP (Hypertext Preprocessor) je skriptovací programovací jazyk, který je určen především pro programování dynamických internetových stránek. Velmi často se začleňuje přímo do jazyka HTML, což umožňuje vznik webových aplikací. Tento jazyk pracuje oproti jazyku HTML na straně serveru. (1)

Při začlenění tohoto jazyka do jazyka HTML je potřeba jej oddělit. K tomu slouží sekvence `<?php` a `?>`. Pokud chceme tvořit webovou stránku pomocí php, musíme mít vytvořený soubor s příponou `.php`. Tento soubor ovšem neumožňuje přímé zobrazení této stránky jako v případě HTML a musí být umístěn v příslušném adresáři webového serveru s aktivním PHP modulem.

Díky výše popsaným vlastnostem jazyka PHP bude funkční část aplikace správa golfového klubu popsána právě v něm.

#### 5.2.4 Dibi

Dibi je abstraktní databázová knihovna pro PHP 5. Mezi základní výhody Dibi patří eliminace výskytu chyb, maximální ulehčení práce programátorům, přenosnost mezi databázovými systémy a hlavně snaha o maximální jednoduchost. Další skvělou vlastností Dibi je imunita proti MySQL injection. Toho je možné dosáhnout pomocí správného používání modifikátorů typu %s, %i, %d atd. Modifikátor %s označuje, že na dané místo bude vložena proměnná typu string. %i udává číslo a %d datum.

Dibi dále disponuje řadou užitečných funkcí. K těm nejvýznamnějším patří `fetchPairs()` a `fetchAssoc()`.

První ze zmiňovaných vrátí výsledek v podobě asociativního pole ve tvaru klíč => hodnota. Pomocí parametrů dále můžeme specifikovat, který ze sloupců bude klíčem a který hodnotou.

Vrcholem je však funkce `fetchAssoc()`, která při spojení vícero tabulek s různými typy vazeb vrátí výsledné tabulce její přirozený tvar. (6)

#### 5.2.5 XML

Jedná se o jazyk, který je považován za nástupce HTML. Výhoda XML spočívá v možnosti tvorby vlastních tagů. Tyto vlastní tagy umožní podstatně lépe označit význam dané informace. A protože je dnešní internet informacemi přehlcen, má jazyk XML výhodu při vyhledávání těchto informací. Jinak řečeno jazyk XML nepopisuje jak se má daná informace zobrazit, ale její význam. Samotné zobrazování je pak řešeno pomocí kaskádových stylů. (9)

### 5.3 Systém přístupu ke stránkám

Systém přístupu na jednotlivé stránky je řešen v hlavním souboru *index.php* a konfiguračním souborem *config.php*, který je v adresáři *Funkce*. Celé zabezpečení přístupu na stránky je řešeno pomocí jednoduchého pole, které je uloženo v souboru *config.php*. Jedná se o dvourozměrné pole, které je složeno z asociovaných hodnot typu klíč => odkaz, kde klíčem je celé číslo a odkazem je myšlen odkaz na konkrétní stránku. Při každém požadavku na zobrazení konkrétní stránky je v souboru *index.php* zjištěno, zda konkrétní požadavek se vyskytuje ve výše uvedeném poli a pokud ano, je uživateli tato stránka zpřístupněná. V opačném případě se jedná o požadavek neexistující stránky a je generován informační výpis.

Pokud je uživateli zpřístupněná konkrétní stránka, je ještě před jejím samotným zobrazením kontrolováno, zda daný uživatel má práva na zobrazení dané stránky nebo její dílčí části. Pokud daná práva nemá, je uživatel o této skutečnosti opět upozorněn příslušným výpisem. V opačném případě se danému uživateli zobrazí buď požadovaná stránka nebo její dílčí část, což záleží na právech konkrétního uživatele.



```

if(isset ($_GET['page'])){
    $str = $_GET['page'];
    if(array_key_exists($str, $pag))
        include $pag[$str];
    else
        echo '<div id="obsah">Snažiš se mě hacknout? </div>';
}else{include 'Funkce/Obsah.php';}

```

Obrázek 24 – Systém přístupu ke stránkám

## 5.4 Funkce

Tato aplikace obsahuje řadu funkcí, které jsou uživatelům přístupny, dle jejich práv a uživatelských rolí. Tyto uživatelské role se dělí na tři části a to: uživatel, registrovaný uživatel a administrátor.

### 5.4.1 Registrace

Jedná se o základní funkci, která je přístupná každému uživateli a je nutné ji podstoupit, pokud se daný uživatel chce stát členem klubu a mít rozšířené možnosti práce na tomto webu. Tato volba se nachází pod logem v pravé horní části.

Samotná registrace je realizována pomocí jednoho krátkého formuláře, který je potřeba celý vyplnit a není zde možno některou informaci vynechat. Tento formulář byl tvořen za účelem neodradit potenciálního uživatele a zjistit o něm pouze nezbytné informace potřebné pro registraci.

Maximální délka řetězce v jednotlivých kolonkách je kontrolována nastavením maximálního možného počtu znaků přímo ve vstupním poli. Další omezení jako unikátnost, absence nebezpečných znaků nebo nutnost shody hesel je kontrolována v php skriptu a po úspěšném splnění všech podmínek je proveden zápis tohoto uživatele do databáze. V opačném případě je vypsána příslušná chyba.

Po úspěšném vyplnění a odeslání formuláře je uživateli automaticky vytvořen fiktivní účet, ze kterého mu je jako registrační poplatek strhnuta částka 1000Kč. Dále je následně uživateli v sekci *Moje údaje* poskytnutá možnost doplnit či změnit své údaje s výjimkou uživatelského jména.

Uživatelské jméno:	<input type="text"/>
Jméno:	<input type="text"/>
Příjmení:	<input type="text"/>
Heslo:	<input type="text"/>
Ověření hesla:	<input type="text"/>
Email:	<input type="text"/>

Pro úspěšnou registraci musíte vyplnit všechny údaje

Obrázek 25 – Registrační formulář

#### 5.4.2 Moje údaje

V této části si může každý uživatel zkontrolovat a následně i upravit vlastní údaje. Je zde uvedená role, která byla danému uživateli přidělená, informace o uživateli, včetně nicku a také údaje o místě bydliště. Tyto údaje si může každý uživatel libovolně měnit. Výjimku tvoří nick a přidělená role a to z důvodů jak oprávnění dané osoby, tak i z důvodů zachování jedinečnosti nicku v databázi.

Nick:	<input type="text" value="admin"/>
Role:	<input type="text" value="Admin"/>
Jméno:	<input type="text" value="Evžen"/>
Příjmení:	<input type="text" value="Mynka"/>
Město:	<input type="text" value="Pardubice"/>
PSC:	<input type="text" value="53002"/>
Ulice:	<input type="text" value="Ulice první"/>
Číslo popisné:	<input type="text" value="114"/>
Telefon:	<input type="text" value="123456789"/>
Email:	<input type="text" value="mynka@seznam.cz"/>

Obrázek 26 – Formulář pro změnu osobních údajů

#### 5.4.3 Turnaje

Tato část se dá rozdělit na dvě části a to část uživatelskou a část administrátorskou.

Do uživatelské části má přístup jak uživatel, tak i administrátor, který má všechna práva uživatele plus práva na administraci. V této části si uživatel může zjistit, které turnaje se mají konat a všechny potřebné informace k těmto turnajům. Dále si může zjistit

na které z nich je přihlášen, může se na jím vybraný turnaj přihlásit nebo se eventuelně z některého přihlášeného turnaje odhlásit. Design je zaměřen především na přehlednost a snadnou ovladatelnost.

V administrační části, která je přístupná jen pro administrátora, jsou připraveny funkce pro správu jednotlivých turnajů. Mezi tyto funkce patří jak vypisování nových turnajů, tak i administrace stávajících turnajů. Mezi další možnosti administrátora patří možnost odepsání určitého účastníka z jím zvoleného turnaje. Všechny vstupní formulářové položky jsou ověřovány jak na straně samotného vstupního pole, tak i na straně php skriptu.

#### **5.4.4 Služby**

Tato sekce je opět rozdělená do dvou částí a to na administrátorskou a uživatelskou.

V uživatelské sekci si může registrovaný uživatel objednat jednu či více služeb, které tento web poskytuje. Design je navržen pro možnost rychlé a jednoduché objednávky přičemž je kladen důraz na minimalizaci chyby uživatele při tvorbě této objednávky. Korektnost požadovaných parametrů je hlídána jak na straně vstupního formuláře, tak i na straně php skriptu.

Dále je uživateli nabídnuta možnost administrace vlastních objednávek, kde si může prohlédnout své objednané služby a nežádoucí služby zrušit. Tato možnost je přístupná přes sekci *Moje\_služby*.

V administrační části je administrátoru nabídnut přehled veškerých objednávek a to jak služeb a produktů, tak i přihlášení na konkrétní turnaj.

#### **5.4.5 Produkty**

Tento web nabízí uživateli možnost zakoupit si golfové pomůcky a vybavení. Tyto produkty jsou rozděleny do pěti kategorií a jsou dostupné v sekci *Zboží*. Zboží si může prohlédnout i neregistrovaný uživatel, avšak pro jeho objednání je nutnou podmínkou přihlášení. Zboží v uživatelem zvolené kategorii je řazeno podle ceny a to vzestupně. Po vybrání druhu zboží a požadovaného počtu je tento produkt automaticky přidán do košíku a uživatel je o této skutečnosti informován výpisem na obrazovce.

Tento košík je pro přihlášeného uživatele zobrazen v pravém horním rohu hned pod logem. Jeho hlavní funkcí je podstatně ulehčit danému uživateli přehled o jím objednaných produktech. Dále tento košík disponuje několika funkcemi potřebnými k modifikaci objednávky uživatele.

<b>Nazev: Cobra Baffler Rail H Hybrid Lady</b>	
	Cena u konkurence: 4692.8Kč
	Naše cena: <b>4190Kč</b>
	U nás ušetříte: <b>502.8Kč</b>
	Počet kusů: 1 ▼
<input type="button" value="Přidat do košíku"/>	
<p>Novinka od společnosti Cobra. Pokračovatel velice populárního hybridu Baffler. Ještě lepší odpouštěcí vlastnosti a ještě delší odpaly. Upravený tvar hlavy hole umožňuje plynuleji procházet rafem. Umístění těžiště podporuje vysoký odpal a dlouhou carry vzdálenost.</p>	

Obrázek 27 – Ukázka produktu v elektronickém obchodě

#### 5.4.6 Košík

Jednou z hlavních částí internetového obchodu je nákupní košík. Tento košík je umístěn v pravé horní části webu, hned nad informacemi o pobočce. Účelem tohoto košíku je plnit dvě základní funkce a to:

- Přehledný výpis a jednoduchou správu produktů, které si zákazník přeje objednat
- Výběr požadované dopravy a potvrzení požadované objednávky

Vzhled samotné stránky byl navržen s ohledem na přehlednost, jednoduchost a intuitivnost při úpravě objednávky.

#### 5.4.7 Nástěnka

Tato možnost umožňuje uživateli umisťovat vzkazy na společnou webovou nástěnku. Pro lepší přehlednost je zde možnost filtrace zobrazovaných vzkazů. Pro lepší formátování textu je zde použit skript TinyMCE.

V administrační části je administrátoru umožněno jak zobrazovat požadované vzkazy, tak i jejich následné odstraňování.

#### 5.4.8 Zprávy

Tato funkce původně nebyla plánována. Její přidání do systému vynutila potřeba komunikace jednotlivých uživatelů jak mezi sebou, tak i s administrátorem webu. Tato komunikace probíhá pomocí textových zpráv, jejichž maximální délka je omezena na 500 znaků. Je zde umožněno formátovat daný text a to jednoduše pomocí TinyMCE skriptu. Typ samotné zprávy se skládá z nicku odesílatel, respektive příjemce, textu samotné zprávy, data a času, kdy byla zpráva odeslána. Tyto zprávy jsou posílány pouze jednomu konkrétnímu uživateli. Dále je zde použit filtr, pomocí kterého si daný uživatel jednoduše vybere, které zprávy si přeje zobrazit.

V administrační části je správci umožněno zobrazovat zprávy dle toho, kdo a komu danou zprávu poslal. Tato možnost zobrazování byla zvolena na základě předpokladu, který počítá s velkým počtem těchto zpráv v systému. Další funkce administrátoru umožňuje mazání jím vybraných zpráv.

#### 5.4.9 Účty

Tato část obsahuje rozsáhlé administrační možnosti ohledně účtů uživatelů. Jsou zde funkce umožňující vložení určité částky na účet daného uživatele, možnost odstranění vybrané položky z databáze účtů, výpis požadovaných hodnot z databáze účtů a nakonec přehledný výpis koncových stavů účtů všech uživatelů.

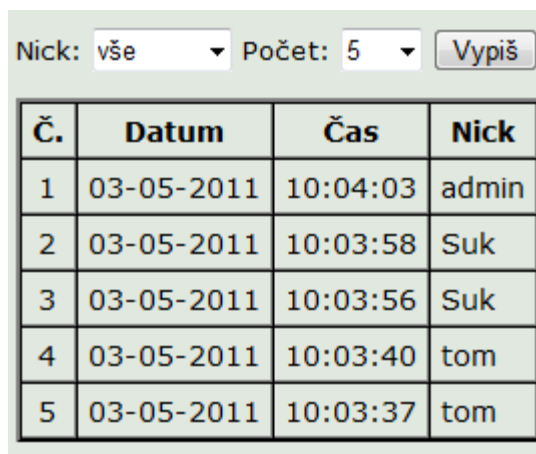
#### 5.4.10 Objednávky

Tato funkce umožňuje administrátoru rychlý a snadný přehled objednávek, a to dle kritérií zadaných ve filtru, který se snaží maximálně zpřehlednit jednotlivé výpisy. V samotném výpisu jsou uvedeny informace o datu objednání, typu zboží či služby, která byla objednána, celkové částce konkrétní objednávky a mnoho dalších informací.

#### 5.4.11 Logování do systému

Pro zkvalitnění zabezpečení stránek byla vytvořena funkce, která automaticky zaznamenává informace o uživateli a čase jeho přihlášení respektive odhlášení ze systému. Hlavním účelem této funkce je umožnit administrátoru sledovat množství přihlášených uživatelů a možnost vytvoření historie přístupu jednotlivých uživatelů.

Pro výběr zobrazovaných informací zde jsou použity filtrační prvky. Informace jsou zobrazovány v tabulkové formě a jsou seřazeny sestupně dle data a času.



Č.	Datum	Čas	Nick
1	03-05-2011	10:04:03	admin
2	03-05-2011	10:03:58	Suk
3	03-05-2011	10:03:56	Suk
4	03-05-2011	10:03:40	tom
5	03-05-2011	10:03:37	tom

Obrázek 28 – Ukázka výpisu logů do systému

#### 5.4.12 Export do XML

Tato funkce umožňuje administrátoru zálohovat obsah nejdůležitějších tabulek do xml dokumentu. V tomto případě se jedná o tabulky Uživatel, Účet a Objednávka. To vše

je zprostředkováno pomocí jednoduchého kliknutí na název tabulky, kterou si administrátor přeje zálohovat.

## 5.5 Adresářová struktura

Adresářová struktura aplikace je navržena tak, aby bylo kdykoliv možné jednoduše dohledat požadované zdrojové kódy. Názvy jednotlivých souborů jsou zvoleny tak, aby co nejlépe vystihovaly akci, kterou mají vykonávat.

Kořenový adresář obsahuje indexovou stránku a kaskádové styly jak pro zobrazení, tak i pro tisk samotné stránky.

Adresář Funkce obsahuje hlavní funkce jako jsou hlavička, menu, patka atd. Dále je zde soubor *config.php*, který je zodpovědný za připojení k samotné databázi a také zabezpečuje přístup uživatelů na jednotlivé stránky s ohledem na jejich práva.

V adresáři Images jsou uloženy veškeré obrázky, které jsou použity jak ke grafickému zobrazení webu, tak i na ukázkou produktů nabízených v internetovém obchodě.

Adresář Pages obsahuje další podadresáře (Admin, Uživatel a Zboží) a několik souborů php, ke kterým mají přístup všichni uživatelé. Další soubory jsou umístěny do složek s ohledem na potřebná práva, která jsou nutná pro přístup k těmto souborům.

Posledními dvěma adresáři zde jsou adresář dibi a tinymce, které obsahují soubory potřebné pro chod databázové vrstvy Dibi a WYSIWYG editoru TinyMCE.

Grafická ukáзка adresářové struktury je přiložena v sekci přílohy (F).

## 6 Závěr

Cílem práce bylo vytvoření webových stránek pro správu golfového klubu. Vývoj aplikace byl zaměřen především na přehlednost stránek a jejich jednoduché a intuitivní ovládání. Cíla aplikace byla napsána pomocí jazyka PHP, který obstarává veškerou logiku, komunikaci s databázovým serverem a dynamičnost stránek.

Při samotném vývoji jsem vycházel ze znalostí získaných během studia a to hlavně v oblasti výběru skriptovacího jazyka PHP. Databázový server MySQL byl vybrán jako snaha o prohloubení mých znalostí v oblasti databází a jako snaha porovnat tento server se serverem Oracle. Také jsem narazil na několik technologií, se kterými jsem se setkal poprvé. Mezi tyto technologie patří hlavně databázová vrstva Dibi. Pomocí této technologie jsem se snažil vytvořit aplikaci, která by se dala použít na různých databázových serverech a vyžadovala by minimální změny kódu. Také mi tato technologie výrazně zjednodušila a zpřehlednila syntaxi databázových dotazů.

Nakonec byly splněny všechny požadavky kladené v zadání, tzn., že byla vytvořena aplikace, která poskytuje všechny standardní funkce pro správu golfového klubu. Snažil jsem se rozpracovat aplikaci do co největších detailů. To se mi nakonec částečně podařilo, protože žádná aplikace není zcela dokonalá a na každé aplikaci by se dalo něco vylepšit.

Mezi možná rozšíření aplikace bych zahrnul širší využití javascriptu na straně klienta, designově přijatelnější a přehlednější tabulky a rozšířenější možnosti administrace webu. Mezi další způsoby zkvalitnění aplikace by patřilo použití některého z frameworků a rozdělení aplikace do nezávislých celků pomocí technologie MVC.


Nad rámec zadání byla vytvořena služba pro komunikaci mezi klienty, se kterou v zadání není počítáno. Další přídatkem je jednoduchý logovací systém, který uchovává v databázi informace o přístupu uživatelů do, respektive ze systému.

## Seznam použité literatury

1. JESUS, Castagnetto, et al. Programujeme PHP profesionálně. Brno : Computer Press, 2004. 656 s. ISBN 80-7226-310-2.
2. OPPEL, Andy. Databáze bez předchozích znalostí. Brno : Computer Press, 2006. 320 s. ISBN 80-251-1199-7.
3. KOFLER, Michael. Mistrovství v MySQL 5. Brno : Computer Press, 2007. 808 s. ISBN 978-80-251-1502-2.
4. Oracle Corporatio. MySQL [online]. 2010 [cit. 2011-04-01]. MySQL::Why MySQL?. Dostupné z WWW: <<http://www.mysql.com/why-mysql/>>.
5. Transakce. In MySQL [online]. Praha : Pavel Kysilka, 2005 [cit. 2011-04-05]. Dostupné z WWW: <[http://www.linuxsoft.cz/article.php?id\\_article=989](http://www.linuxsoft.cz/article.php?id_article=989)>.
6. GRUDL, David. Dibi [online]. 2008 [cit. 2011-04-11]. Dibi. Dostupné z WWW: <<http://dibiphp.com/cs/>>.
7. Netcraft [online]. Aprile 2011 [cit. 2011-04-11]. April 2011 Web Server Survey. Dostupné z WWW: <<http://news.netcraft.com/>>.
8. Techspot [online]. 2011-3-1 [cit. 2011-04-11]. IE back to losing share, Firefox gains a little, Chrome still surging. Dostupné z WWW: <<http://www.techspot.com/news/43096-ie-back-to-losing-share-firefox-gains-a-little-chrome-still-surging.html>>.
9. XML [online]. 28. květen 1999 [cit. 2011-04-29]. XML. Dostupné z WWW: <<http://www.kosek.cz/clanky/xml/xml-uvod.html>>.




## Příloha A – Vzhled webové stránky po přihlášení administrátora



**Vítej admin**

[Odhlásit](#)

**Můj košík**



**Naše pobočka**  
Město : Pardubice  
Ulice : 17. Listopadu  
PSČ : 53002  
Telefon : 607123456  
IČO : 123456543

**Odkazy**  
[Golf.cz](#)  
[GolfPortal](#)  
[Golfové komponenty](#)

---

**o golfu**

**Základní informace**

Golf je venkovní sport, který se hraje golfovým míčkem do jamky a používá přitom různé hole. Golf byl hrán na Britských ostrovech po několik století. Původně byl považován za hru pro elitu, ale nyní získává všeobecnou popularitu a počet hráčů golfu celosvětově roste. Golf patří mezi olympijské sporty, dříve byl zařazen do programu LOH 1900 a 1904. Golf bude opět na programu LOH v roce 2016 (Rio de Janeiro).

[více na: Základní informace](#)

**Jak začít hrát golf**

Pokud se rozhodujete začít hrát golf, je několik skutečností, které musíte zvážit. První je dostupnost golfu ve vašem okolí. Dnes už je síť golfových hřišť a tréninkových areálů poměrně hustá, z většíny míst v naší republice na některé narazíte do půl hodiny jízdy.

[více na: Jak začít hrát golf](#)

---

**Pravidla golfu**

Zde najdete pravidla golfu - nejprve ve zkrácené verzi, která postačuje ve většině případů (ale není možné ji použít při žádné soutěži) a níže kompletní pravidla platná od 1. 1. 2004.

[více na: Pravidla golfu](#)



---

**Vítej admin**

**Hlavní stránka**

**Služby**

- Kurzy s trenérem
- Pronájem hřiště

**Zboží**

- Oblečení
- Hole
- Míčky
- Batohy
- Další

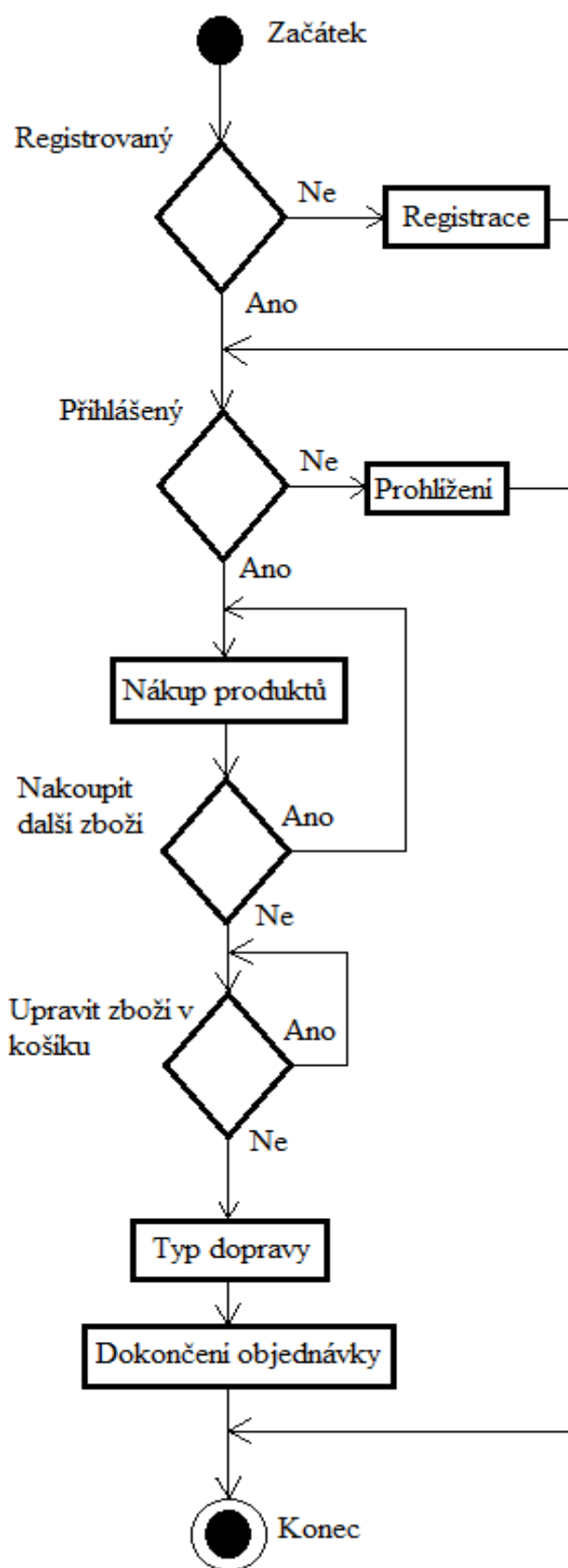
**Možnosti**

- Moje údaje
- Můj účet
- Turnaje
- Objednat služby
- Moje služby
- Nástěnka
- Zprávy

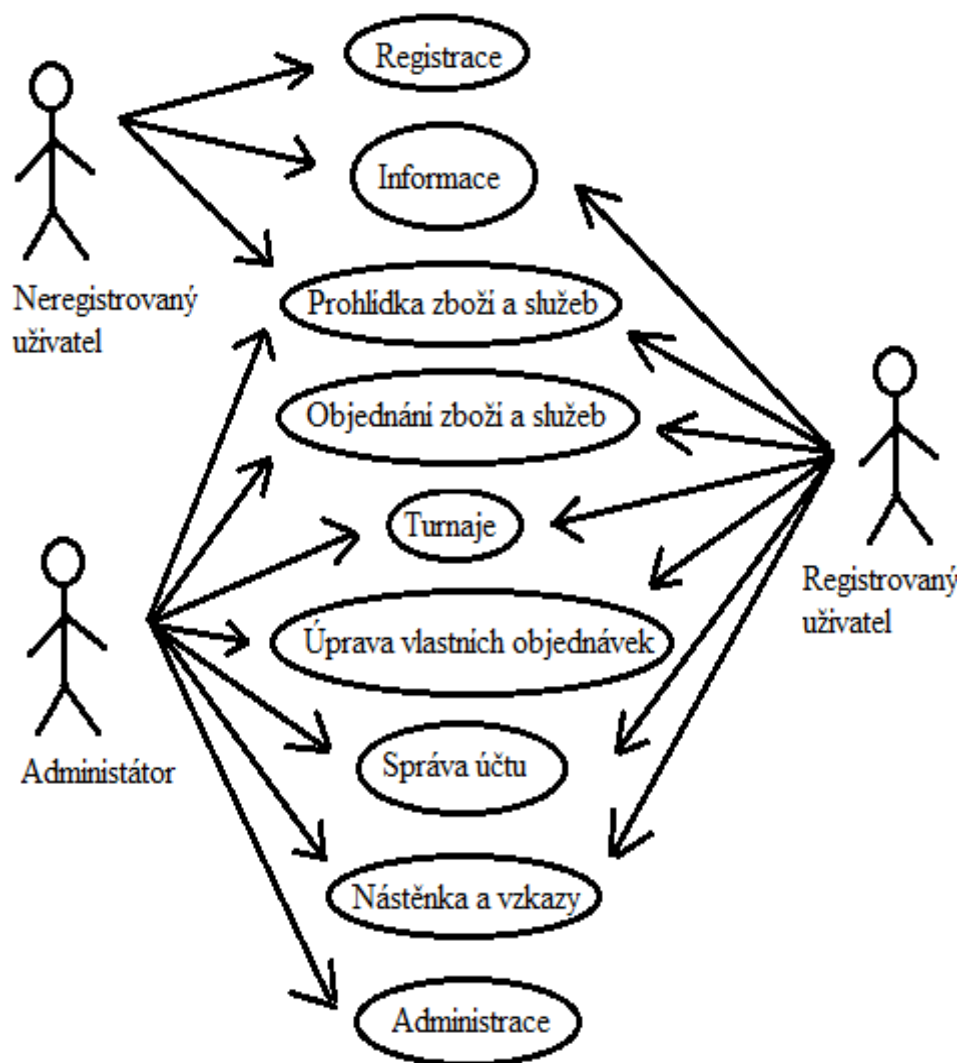
**Administrace**

- Práva
- Účty
- Turnaje
- Objednávky
- Nástěnka
- Uživatelé

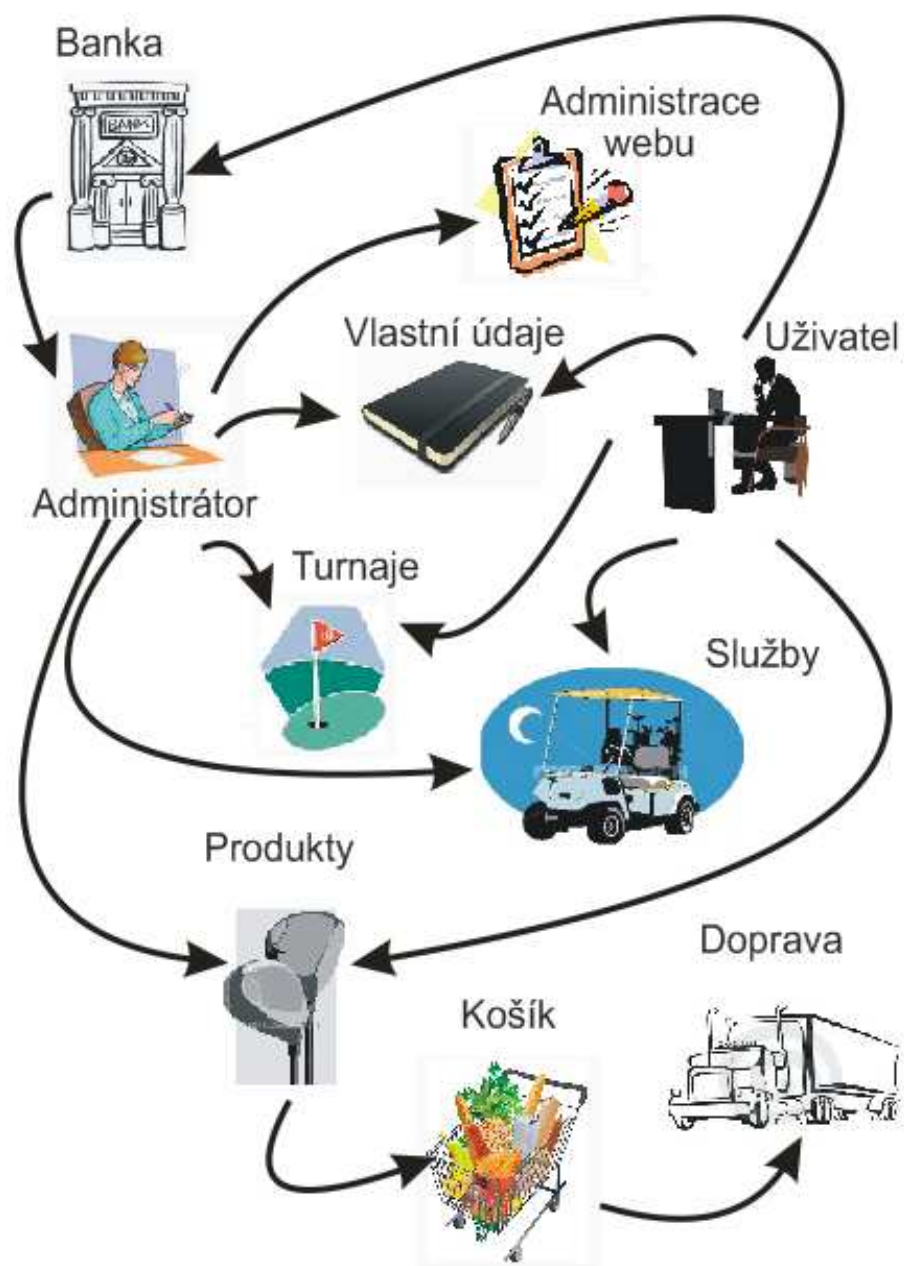
## Příloha B – UML activity diagram



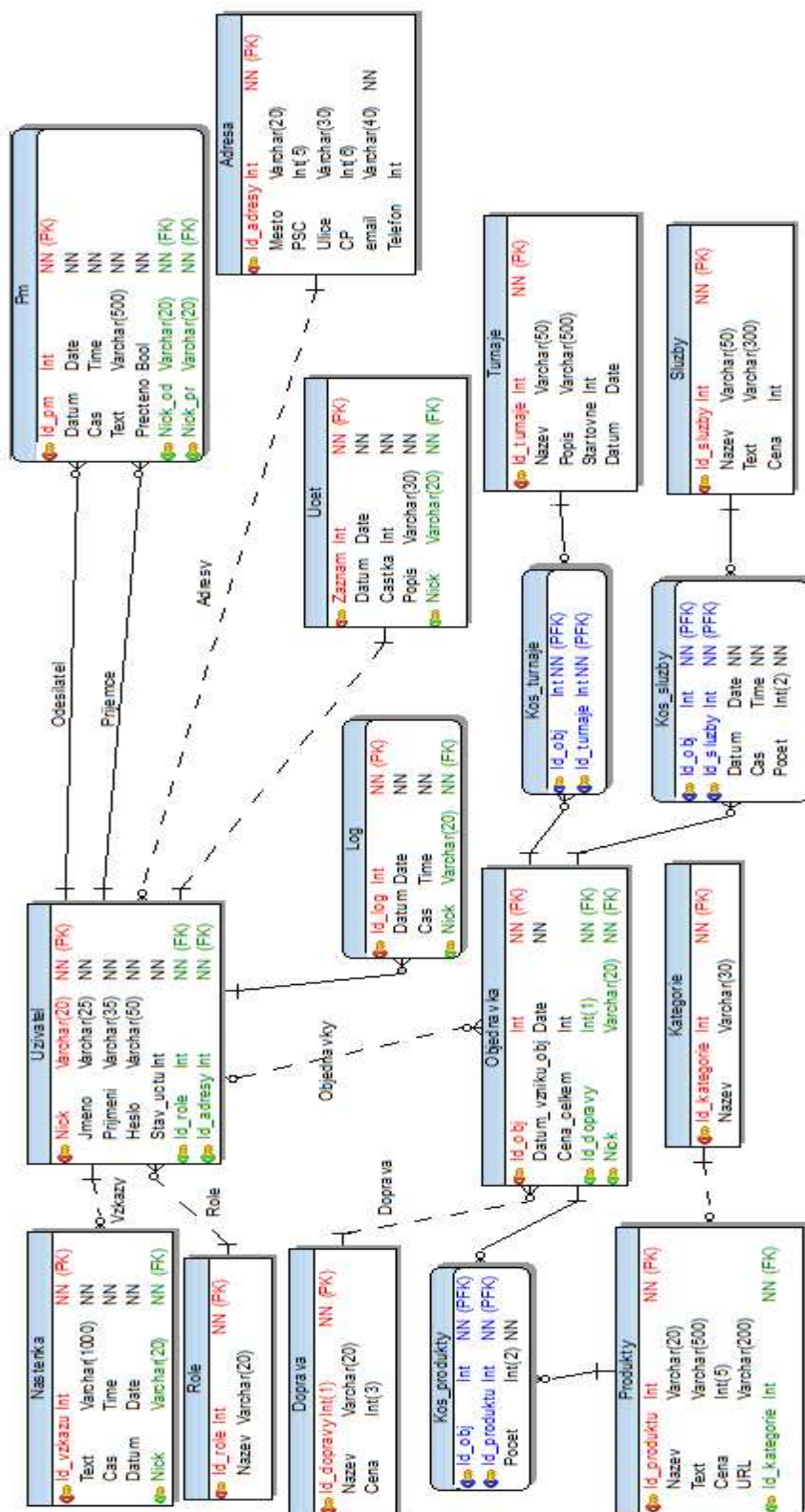
## Příloha C – UML use case diagram



## Příloha D – Rich Picture



## Příloha E – E-R diagram



## Příloha F – Adresářová struktura

