

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 FITCENTRA RELAX

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

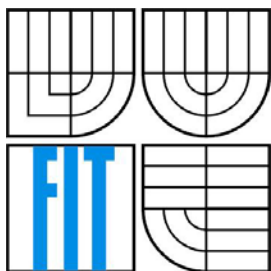
AUTOR PRÁCE  
AUTHOR

JAN HODAŇ

BRNO 2008



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 FITCENTRA RELAX

INFORMATION SYSTEM OF RELAX HEALTH FARM

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

AUTOR PRÁCE  
AUTHOR

JAN HODAŇ

VEDOUCÍ PRÁCE  
SUPERVISOR

ING. ŠÁRKA KVĚTOŇOVÁ

BRNO 2008

## Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav informačních systémů

Akademický rok 2007/2008

### Zadání bakalářské práce

Řešitel: **Hodaň Jan**

Obor: Informační technologie

Téma: **Informační systém fitcentra RELAX**

Kategorie: Databáze

Pokyny:

1. Seznamte se s dostupnými nástroji a jazyky pro tvorbu webových aplikací, zejména s databází MySQL, HTML, PHP apod.
2. Proveďte podrobnou analýzu požadavků na informační systém fitcentra na základě konzultace se zadavatelem projektu. Vytvořte vhodné modely tohoto systému.
3. Zvolte vhodné implementační prostředí a realizujte prototyp navrženého systému.
4. Na vhodně zvoleném vzorku dat demonstруйте použití a funkčnost vytvořené aplikace.
5. Zhodnoťte dosažené výsledky a diskutujte možnosti dalšího rozšíření.

Literatura:

- Williams, H. E., Lane, D.: PHP a MySQL - Vytváříme webové databázové aplikace. Computer Press, 2002, 552 s. ISBN 8072267604
- Kosek, J.: HTML, tvorba dokonalých www stránek. Praha: Grada Publishing, 1998, 291 s. ISBN 80-7169-608-0
- PHP: Hypertext Preprocessor. Dostupné na: [www.php.net](http://www.php.net)
- DeLisle, M.: PHPMyAdmin - efektivní správa MySQL. Brno: Zoner Press, 270 s. ISBN 8086815099
- Ullman, L.: PHP a MySQL. Computer Press, 2004, 536 s. ISBN 8025100634

Při obhajobě semestrální části projektu je požadováno:

- Bez požadavků.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese <http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Květoňová Šárka, Ing.**, UIFS FIT VUT

Datum zadání: 1. listopadu 2007

Datum odevzdání: 14. května 2008

**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
Fakulta informačních technologií  
Ústav informačních systémů  
612 66 Brno, Božetěchova 2



---

doc. Ing. Jaroslav Zendulka, CSc.  
vedoucí ústavu

**LICENČNÍ SMLOUVA**  
**POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO**

uzavřená mezi smluvními stranami:

**1. Pan**

Jméno a příjmení: **Jan Hodaň**

Id studenta: 84371

Bytem: Družební 1176, 742 21 Kopřivnice

Narozen: 21. 06. 1985, Nový Jičín

(dále jen “autor”)

a

**2. Vysoké učení technické v Brně**

Fakulta informačních technologií

se sídlem Božetěchova 2/1, 612 66 Brno, IČO 00216305

jejímž jménem jedná na základě písemného pověření děkanem fakulty:

.....

(dále jen “nabyvatel”)

**Článek 1**  
**Specifikace školního díla**

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):  
bakalářská práce

Název VŠKP: Informační systém fitcentra RELAX

Vedoucí/ školitel VŠKP: Květoňová Šárka, Ing.

Ústav: Ústav informačních systémů

Datum obhajoby VŠKP: .....

VŠKP odevzdal autor nabyvateli v:

tištěné formě                      počet exemplářů: 1

elektronické formě              počet exemplářů: 2 (1 ve skladu dokumentů, 1 na CD)

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

## **Článek 2**

### **Udělení licenčního oprávnění**

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti:
  - ihned po uzavření této smlouvy
  - 1 rok po uzavření této smlouvy
  - 3 roky po uzavření této smlouvy
  - 5 let po uzavření této smlouvy
  - 10 let po uzavření této smlouvy(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/ 1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

## **Článek 3**

### **Závěrečná ustanovení**

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne: .....

.....  
Nabyvatel

.....  
Autor

## **Abstrakt**

Tato bakalářská práce se zabývá vytvářením informačního systému fitcentra RELAX v Kopřivnici. Hlavním cílem je zpřístupnění informací o fitcentru na internetových stránkách za účelem získání potenciálních nových zákazníků. Těm bude výsledný systém následně nabízet celou řadu členských výhod a zvýhodněných cen vybraných akcí či produktů. Díky informačnímu systému je zjednodušená organizace ve fitcentru, za pomoci zautomatizování různých procedur a snadné úpravy poskytovaných údajů.

## **Klíčová slova**

informační systém, WWW stránky, fitcentrum, PHP, MySQL, HTML, CSS, JavaScript

## **Abstract**

The essay deals with creation of information system for RELAX health farm in Koprivnice. The main aim of this project is to make available information about fitness club on its website pages in order to gain customers. The system will then offer various special offers to new and current customers. Thank to the information system, various tasks in the fitness club are simplified by using automated procedures and easy access to customers' data.

## **Keywords**

information system, WWW sides, health farm, PHP, MySQL, HTML, CSS, JavaScript

## **Citace**

Hodaň Jan: Informační systém fitcentra RELAX. Brno, 2008, bakalářská práce, FIT VUT v Brně.

# Informační systém fitcentra RELAX

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Šárky Květoňové.

Další informace mi poskytli Radka Žlebková, Martina Jeličová a Aleš Polesz.

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

.....  
Jan Hodaň  
29. 4. 2008

## Poděkování

Chtěl bych poděkovat slečně Ing. Šárce Květoňové za trpělivost a vedení mých kroků k úspěšnému vypracování bakalářské práce. Dále bych chtěl poděkovat majiteli fitcentra RELAX panu Aleši Poleszovi, který mi umožnil přístup ke všem údajům a organizaci fitcentra a zaměstnankyním Radce Žlebkové a Martině Jeličové za postřehy a připomínky k tomuto projektu.

© Jan Hodaň, 2008.

*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 .....	7
1 Úvod.....	9
2 Informační systémy.....	10
2.1 Co je informační systém.....	10
2.2 Historie IS .....	11
2.3 Vytváření IS .....	12
2.3.1 Formulace požadavků .....	13
2.3.2 Datová a funkční analýza.....	13
2.3.3 Návrh IS.....	13
2.3.4 Implementace IS .....	13
2.4 Životní cyklus IS .....	14
2.5 Základní modely životního cyklu IS .....	14
2.5.1 Vodopádový model.....	14
2.5.2 Model výzkumník .....	15
2.5.3 Prototypový model.....	16
2.5.4 Spirálový model.....	17
2.5.5 Evoluční model.....	19
3 Přehled použitých technologií.....	20
3.1 Unifikovaný modelovací jazyk UML.....	20
3.2 Databáze MySQL.....	21
3.3 Značkovací jazyk HTML .....	22
3.4 Kaskádové styly CSS .....	23
3.5 Skriptovací jazyk JavaScript .....	24
3.6 Skriptovací jazyk PHP .....	25
4 Analýza a návrh IS.....	26
4.1 Popis požadavků.....	26
4.2 Jednotlivé funkce systému .....	27
4.3 Modely systému .....	28
4.3.1 Use - Case diagram .....	29
4.3.2 ER diagram .....	29
5 Implementace.....	32
5.1 Vytvoření databáze a tabulek .....	32
5.1.1 Datové typy v databázi.....	32
5.1.2 Klíče a integritní omezení.....	32



5.1.3	Hodnoty NULL.....	33
5.2	Implementované části systému.....	33
5.2.1	Vzhled IS .....	33
5.2.2	Registrace.....	34
5.2.3	Přihlášení a odhlášení .....	35
5.2.4	Aktuality, tělocvična, cvičitelé, kontakty.....	36
5.2.5	Ceník.....	37
5.2.6	Vyživovací doplňky – objednávky .....	38
5.2.7	Váš účet.....	40
5.2.8	Permanentky .....	40
5.2.9	Tréninkový plán.....	42
5.2.10	Správa aktualit.....	43
5.2.11	Zaměstnanci a cvičitelé .....	44
5.2.12	Aktualizace ceníku a produktů .....	44
5.2.13	Rezervace tělocvičny.....	45
5.2.14	Chatbox.....	45
6	Závěr .....	47
	Literatura .....	48
	Seznam příloh.....	50

# 1 Úvod

V dnešní době se vše točí kolem Internetu. Každá reklama, každá informace obsahuje sdělení typu „objednávejte na WWW, více informací na WWW“ apod. V současné době jsme dospěli do stádia, kdy můžeme říci, že kdo není na internetu, jako by nebyl. A právě proto se již informační systémy netýkají pouze velkých a bohatých firem, ale začínají se jejich prostřednictvím prezentovat i menší firmy, úzce zaměřené na vybrané druhy poskytovaných služeb v lokálních oblastech. Na svých webových stránkách nabízejí produkty a poskytované služby, kontaktní informace na majitele a provozovny a samozřejmě také aktuality okolo dění ve firmě. Cílem této bakalářské práce bylo vytvořit informační systém pro fitcentrum RELAX v Kopřivnici.

Informační systém fitcentra je především zaměřen na lidi, kteří fitcentrum navštěvují a nabízí jim různé členské výhody, aby si fitcentrum své zákazníky udrželo. A také pro lidi, kteří mají o dané téma zájem a chtějí se na něj blíže informovat.

Co je vlastně informační systém se dozvíme ve druhé kapitole. Je tady popsána historie a architektura informačních systémů. Dále je zde podrobně popsáno, jak se informační systém vytváří a jaký je jeho životní cyklus. Jaké máme základní modely životního cyklu, se podrobněji dozvíme na konci téhle kapitoly.

Tématem třetí kapitoly je seznámení se s použitými technologiemi, které jsem využíval při vytváření této bakalářské práce. Blíže se zde podíváme na jazyk UML, který jsem použil při návrhu informačního systému. Dále je zde zmínka o tom, jak se vytváří databáze s použitím jazyka MySQL, která nám prezentuje datovou vrstvu. Pro komunikaci s uživatelem byl použit jazyk HTML společně s kaskádovými styly CSS. V závěru kapitoly nesmíme zapomenout na skriptovací jazyky PHP a JavaScript, které zajišťují funkčnost a také dynamický vzhled aplikace.

Čtvrtá kapitola je věnována analýze a návrhu daného systému. V první části jsou shrnuty požadavky, co by měl systém umět. Tyhle požadavky jsou specifikovány zadavatelem práce, v našem případě majitelem fitcentra RELAX. Dále se kapitola zabývá jednotlivými funkcemi a modely systému. Ukážeme si zde jak vypadá Use-Case diagram (diagram případu užití) a ER diagram.

V páté kapitole si popíšeme jak probíhala vlastní implementace daného systému. Nejprve se podíváme na vytvoření databáze a tabulek a jejich vlastnosti jako jsou primární a cizí klíče a vlastnost null/not null. Poté si popíšeme jednotlivé implementované části systému, které byly požadovány zadavatelem práce.

Poslední dvě kapitoly jsou věnovány dalšímu rozvoji tohoto systému, možným vylepšením už stávajících funkcí a zhodnocení dosažených výsledků.

## 2 Informační systémy

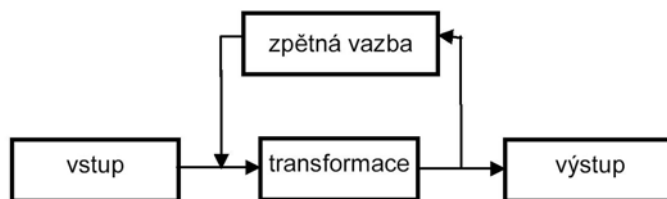
Tato kapitola nás blíže seznámí s pojmem informační systém, jeho historií a architekturou. V dalších podkapitolách je podrobněji uvedeno, jak se informační systém vytváří a podrobněji popsane jednotlivé etapy vývoje. Nakonec jsou zde zmíněny informace o životním cyklu IS a jeho základní modely.

### 2.1 Co je informační systém

Informační systémy (IS) jsou systémy pro sběr, udržování, zpracování a poskytování informací a dat. Takovým příkladem IS může být například kniha příchodů do zaměstnání, telefonní seznam, katalog s různými nabídkami, jídelní lístek atd. Není tedy nutné, aby systém pracoval vždy na počítači, může být i v papírové formě. My se dále budeme zabírat IS pracujícími na počítači.

Na začátku jsme si řekli, že IS pracuje s informacemi a daty, co to tedy je? Informace je nějaké sdělení, které nám odstraňuje jistou dávku neznalosti nebo nevědomosti. Za data považujeme jakékoliv zaznamenané poznatky či fakta. Informaci pak můžeme chápat jako spojení dat s nějakým významem. [13]

Informační systém se skládá ze vstupní části, kde zadáváme informace, a výstupní části, ze které informace naopak získáváme. Aby výstupní data nebyla totožná se vstupními, musí mezi nimi probíhat jejich transformace. V transformační části probíhají procesy, které realizují transformace většinou ve formě transakcí. Nesdílou součástí IS je také zpětná vazba, reagující na vnitřní stavy systému. Z čeho se tedy IS skládá je vidět na obrázku 2.1:



Obrázek 2.1: Schéma informačního systému [7]

Informační systém se rozděluje na tzn. vrstvy. To jsou části systému, které mají rozdílné funkce a implementují se různými prostředky. V současnosti se nejvíce využívá 3-vrstvá architektura:

- Klientská (presentační) vrstva – slouží ke komunikaci s uživatelem.
- Aplikační (funkční) vrstva – obsahuje vlastní aplikace (uložení dat v databázi, operace nad daty) a zajišťuje komunikaci mezi uživatelským rozhraním a databázovým systémem. Také se stará o bezpečnost a propojení s ostatními sítěmi.
- Datová vrstva – řeší samotný přístup k datům, které jsou uložena v databázi.

## 2.2 Historie IS

Historie počítačové podoby IS sahá do poloviny 50. let minulého století. Tohle období se nazývá prehistorické a jeho hlavními rysy byly:

- Definice dat byla součástí definice procesů.
- Data byla součástí procesů.
- Data nemohou být sdílena více procesy.

V tomto období pojmy jako soubor či databáze neexistovaly. Neexistovaly ani počítačové sítě či jiná vzájemná komunikace dvou počítačů. Nebyly k dispozici dnes již standardní počítače s klávesnicí a monitorem jako vstupními a výstupními zařízeními, ale byly řízeny dávkovým zpracováním v podobě děroštitkových balíčků. V tomto období byl uživatel nucen zahrnout vše do svého programu: popis dat, data samotná, algoritmy a uživatelské rozhraní. Používaly se různé typy zpracování, které se nazývaly modely.

Od poloviny 60. let nastala v IS změna. Došlo zde k oddělení dat od procesů, ostatní vlastnosti však zůstávají stejné. Definice dat byla stále součástí definice procesů a data nemohla být sdílena více procesy. Výsledkem tohoto oddělení dat od procesů ale bylo to, že definice programu umožňovala pracovat s více databázemi a jeden systém mohl spravovat více sad údajů. V tomto období se také začaly objevovat systémy pro zpracování souborů v operačních systémech a samostatné systémy pro zpracování souborů, které umožnily zpracovávat data více procesy současně. Výhodou bylo oddělení dat od procesů a tím jejich zpřístupnění více procesům. Nevýhodou byla neexistující unifikovaná definice dat, dále jejich případná nekonzistentnost (upravována více uživateli, nezaveden pojem transakce) a redundance (uložení stejných dat ve více souborech).

Klíčovou dobou v oblasti IS byla 70. léta. Byly zde zavedeny pojmy databáze a systém řízení báze dat. Hlavními rysy tohoto období byly:

- Definice dat nejsou součástí definice procesů.
- Data nejsou součástí procesů.
- Data mohou být sdílena více procesy pomocí transakcí.

Procesy již nemají přímý přístup k datům, které jsou uloženy v databázi, ten se odehrává pomocí systému pro řízení báze dat. Jelikož už není popis dat součástí definice procesů, je dosaženo nezávislosti dat na uživatelských programech a hlavně nezávislosti uživatelských programů na datech. Zavádí se pojem transakce, který zvyšuje konzistenci dat a zároveň snižuje jejich redundanci, čímž umožňuje atomizaci procesů. Dalšími výhodami je strukturovaný a srozumitelný přehled o datech, možnost snadno vyvíjet IS v aplikacích vyžadujících více zdrojů dat a jednotný přístup k datům, s možností realizace vysokého stupně abstrakce. I když se databázové technologie využívají dodnes, mění se způsob ukládání dat v databázi. V dnešní době se nejvíce využívá relační model dat.

V 80. letech se již manipulace s daty nemění. Změny však dosahuje uživatelské rozhraní, jehož definice už není součástí definice procesů a vznikají aplikace se standardizovaným rozhraním vzhledem ke komunikační vrstvě. Díky tomu je umožněn přístup k aplikacím z různých typů rozhraní, např. z Internetu, tiskárny, displeje apod. Tímhle dosáhneme rozvoje i druhé důležité části IS a to komunikaci s uživatelem.

Workflow je pojem, který se objevuje v 90. letech. Jde o programové vybavení definující procesy mimo aplikace. Stav informačního systému je měněn procesem, který je obvykle dán stavem databáze a popsán jako transformační funkce typicky mimo databázi. Definice procesů se do 90. let nejčastěji prováděla procedurálním způsobem.

Posledním pojmem, který se nám v historii IS objevuje je OLAP, což je zkratka okamžitého zpracování dat. Je to technologie pro uložení dat v databázi, umožňující uspořádání velkých objemů dat tak, aby byla pro uživatele srozumitelná a přístupná. Mezi její hlavní rysy patří, že nemění stav IS, neprovádí žádné transakce a předpokládá předem nespecifikované množství dotazů pracujících s agregovanými daty. Největší rozdíl u technologie OLAP od předešlých technologií je v tom, že se jedná o jednorázové nahrání dat a následné provádění složitých dotazů nad nimi, zatímco předešlé technologie přidávaly a modifikovaly data často, průběžně a to obvykle mnoha uživateli současně.

V dnešní době se vytváří informační systémy především kombinací jazyků MySQL, PHP, HTML/XHTML a Javascript. Informační systémy se staly nedílnou součástí našeho života a obrovským zdrojem informací. [7, 12, 13]

## 2.3 Vytváření IS

Postup vytváření IS má několik fází, které se dají rozdělit následovně:

- formulace požadavků,
- datová a funkční analýza,
- návrh IS,
- implementace IS.

V následujících podkapitolách si jednotlivé fáze více rozebereme.

### **2.3.1 Formulace požadavků**

V této první fázi je důležité specifikovat, co má a nemá daný systém umět, jak bude fungovat a jak by mělo vypadat komunikační prostředí. Tyto informace se získávají na základě komunikace se zadavatelem projektu. Nejdůležitějším výstupem této úrovně vzniku IS je zpráva, definující základní problémy a jejich způsoby řešení. Tato studie by měla obsahovat několik základních bodů: [8]

- Detailní a precizní specifikaci elementárních problémů.
- Formulaci požadavků pro řešení problémů.
- Postup realizace projektu.
- Odhad, jak bude projekt finančně a časově náročný.

Po vypracování a schválení této zprávy je možno přejít na další fázi vývoje IS.

### **2.3.2 Datová a funkční analýza**

Fáze datové a funkční analýzy se zabývá analýzou požadavků a vytvořením analytického konceptuálního modelu. Předpokládá se zde úplné pochopení řešených problémů a specifikuje se zde jejich definitivní řešení. Opět zde probíhá komunikace se zadavatelem pro získání podrobných a přesných informací o požadavcích na informační systém a vazbách na již existující a používané IS. Provádí se zde sestavení modelů existujících IS a výstupem této fáze je analytický návrh nového IS. Tento návrh musí být opět konzultován a schválen zadavatelem.

### **2.3.3 Návrh IS**

Tato fáze obsahuje transformaci analytického konceptuálního modelu na databázový model a definici jeho procesů. Musí se vytvořit celková architektura systému. Volí se zde software, ve kterém se bude systém vytvářet, navrhuje se datové soubory, struktura programu, formuláře a sestavy. Také se připravuje sada testovacích úloh. Úlohou návrhu informačního systému je tedy sestavit vhodný datový model, tzn. jakým způsobem se budou v systému uchovávat informace a pomocí jakých pravidel budou zpracovány a poskytovány k dalšímu užití. Výsledkem této fáze je model informačního systému se všemi funkčními celky, jejich vzájemná komunikace a propojení s daty.

### **2.3.4 Implementace IS**

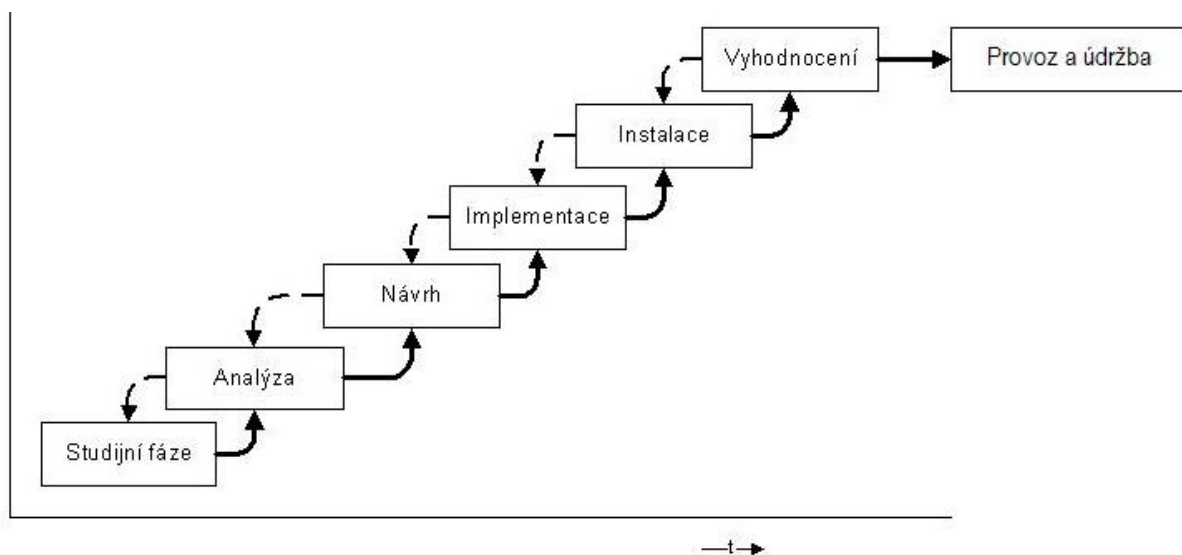
Podstatou implementace informačního systému je tvorba a testování jednotlivých programových modulů, na základě návrhů z analýzy IS, pomocí vhodných implementačních prostředků, které byly zvoleny taktéž při návrhu. Dále je tato fáze zaměřena na tvorbu testovacích dat a testovacích programů. Je zde také vytvářena dokumentace k programům a provozní dokumentace spojená se zaškolením pracovníků zadavatele projektu. Výsledkem implementace je funkční informační systém, který obsahuje všechny zadané požadavky.

## 2.4 Životní cyklus IS

Do životního cyklu informačního systému patří kromě již zmíněných fází vytváření IS také instalace, vyhodnocení a provoz a údržba IS. Instalace je poslední fází před spuštěním zkušebního provozu IS. Doplňují se zde potřebná data a zpracovávají se zkušební úlohy.

Předposlední fází celého životního cyklu je vyhodnocení IS. Doba trvání této fáze je úměrná rozsáhlosti projektu. Jejím výstupem jsou mimo jiné i připomínky zadavatele a výstupy testovacích programů. Výsledkem je protokol o zkušebním provozu, kde se zmiňujeme i o případných závadách a jejich odstranění. Po všech fázích nakonec přichází samotný provoz a údržba informačního systému.

Jednotlivé fáze životního cyklu nemusí být postupné, často se některé činnosti provádí paralelně. Také tvorba informačního systému není přímočará. Po vyhodnocení jednotlivých fází se přechází k další činnosti, ale v případě nesouhlasu s požadavky se musíme vrátit k předchozí fázi a předělat ji. Jak tedy vypadá obecný životní cyklus IS vidíme na obrázku 2.2:



Obrázek 2.2: Životní cyklus IS [8]

## 2.5 Základní modely životního cyklu IS

Pro tvorbu informačních systémů je možno použít různé modely životního cyklu. Jsou to formalizované popisy činností, které mají danou časovou posloupnost. Z hlediska toho, jak navazují jednotlivé fáze životního cyklu na sebe, rozlišujeme:

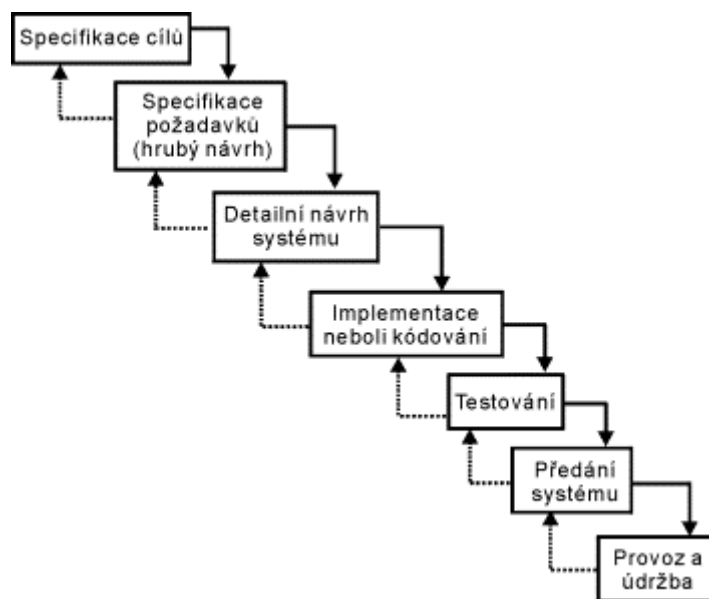
### 2.5.1 Vodopádový model

Vodopádový model je nejstarším klasickým modelem životního cyklu. Začal se používat již v 70. letech minulého století pro výstavbu automatizovaných systémů řízení. Cílem vodopádového modelu bylo zavedení jednotného řádu do vývoje informačních systémů. Díky jeho hierarchické dekompozici

bylo umožněno řešení komplexnějších problémů a díky kontrole výstupů jednotlivých etap se snížilo množství chyb.

Základní charakteristikou je tzv. vrstvená forma. Jedná se o to, že při návrhu informačního systému se jednotlivé etapy životního cyklu provádí postupně, navazují na sebe, ale neprotínají se. Jednotlivé fáze postupují podle daného plánu realizace, výstup předcházející fáze je zároveň vstupem následující a zpětně se už k fázím nevrací.

Příložený obrázek 2.3 vyjadřuje postupný plán realizace modelu vodopád.



Obrázek 2.3: Vodopádový model [9]

Výhodami tohoto modelu je poměrně rychlý a levný postup řešení, ale to pouze v případě, že se nevyskytnou problémy. Proto je vhodný na návrhy systému, kde předem známe problém i způsob jeho řešení.

Nevýhody a problémy s tohoto modelu jsou především v tom, že reálné projekty zřídka sledují jednotlivé fáze modelu v předem daném pořadí. Také zadavatelé nejsou schopni sdělit všechny potřebné požadavky na systém hned při specifikaci. A jelikož provozuschopnost systému vidí zadavatel až při závěrečné fázi, je následná úprava požadavků náročná a drahá.

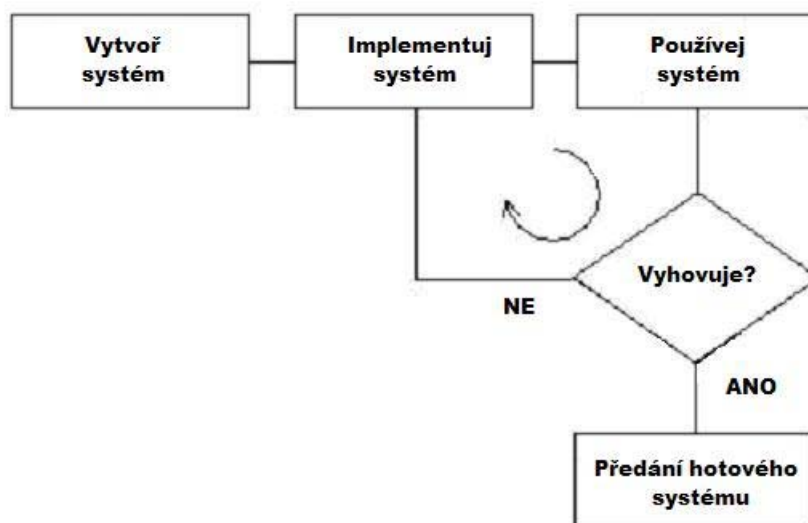
I přes všechny nevýhody je toto řešení lepší než náhodný postup řešení informačních systémů.

## 2.5.2 Model výzkumník

Během vývoje informačního systému pomocí tohoto modelu se řešitelé při získávání nových poznatků často vrací k implementační fázi. To je také základní charakteristikou modelu. Díky tomuhle způsobu vývoje IS nemůžeme plánovat časové ani finanční nároky celého projektu.

Příložený obrázek 2.4 popisuje vývoj IS pomocí modelu výzkumník.





Obrázek 2.4: Model výzkumník [10]

Výhodou modelu výzkumník je velmi dobrá flexibilita požadavků při jejich dodatečné změně zadavatelem.

Hlavní nevýhodou tohoto modelu je vysoká manažerská náročnost, protože neznáme časové, finanční ani personální vytížení projektu a také dokumentace. Pokud dokumentace nevzniká průběžně, je odrazem hotového díla a my při údržbě nevíme, zda dané požadavky byly ustanoveny při zadání projektu nebo jsou nové.

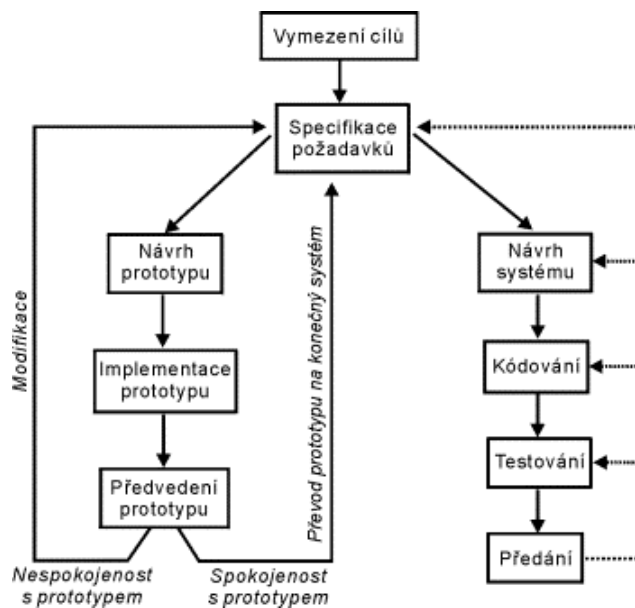
Model výzkumník se používá především v situacích, kdy nemáme zkušenosti s daným typem problému a až postupem času, na základě implementace, získáváme znalosti pro lepší specifikaci zadání a návrhu systému.

### 2.5.3 Prototypový model

Prototypový model se začal prosazovat v 80. letech. Cílem prototypového modelu bylo v co nejkratší době seznámit zadavatele s prvními verzemi systému a využívání prototypů. Prototyp chápeme jako část implementovaného systému, která může mít logickou nebo fyzickou formu, má všechna vnější rozhraní a umožňuje reakci zadavatele na výsledky.

Základní charakteristikou prototypového modelu je umožnění změn výchozích požadavků zadavatele a jednoduchá reakce na tyto změny. Tím se výrazně liší od vodopádového modelu. Po prezentaci prototypu se zadavatel rozhodne, zda mu takové řešení vyhovuje nebo nevyhovuje. V prvním případě je prototyp dále zpracováván, je udělána podrobná specifikace a systém se implementuje v této podobě. Při neschválení prototypu je tento prototyp buď zahozen nebo přepracován tak, ať je zadavatelem schválen.

Schéma postupu u prototypového modelu je znázorněno na obrázku 2.5.



Obrázek 2.5: Prototypový model [9]

Prototypový model má největší výhodu v tom, že při společné komunikaci zadavatele a řešitele dokáže co nejlépe vyjasnit požadavky i jejich implementaci, protože zadavatel vidí a testuje prototypy od brzkého stádia vývoje celého informačního systému.

Nevýhodou tohoto modelu může být náročnost na vedení projektu v případě, že různé skupiny řešitelů dělají různé prototypy. U rozsáhlých systémů je také nutnost zvolit si předem dané množství opakování prototypu a stanovit datum jejich dokončení. Při nestanovení těchto podmínek by se mohla jejich příprava značně prodloužit a časově protáhnout.

Prototypový model se používá v případech, kde není předem jasné zadání projektu informačního systému nebo očekáváme jeho změny v průběhu vývoje.

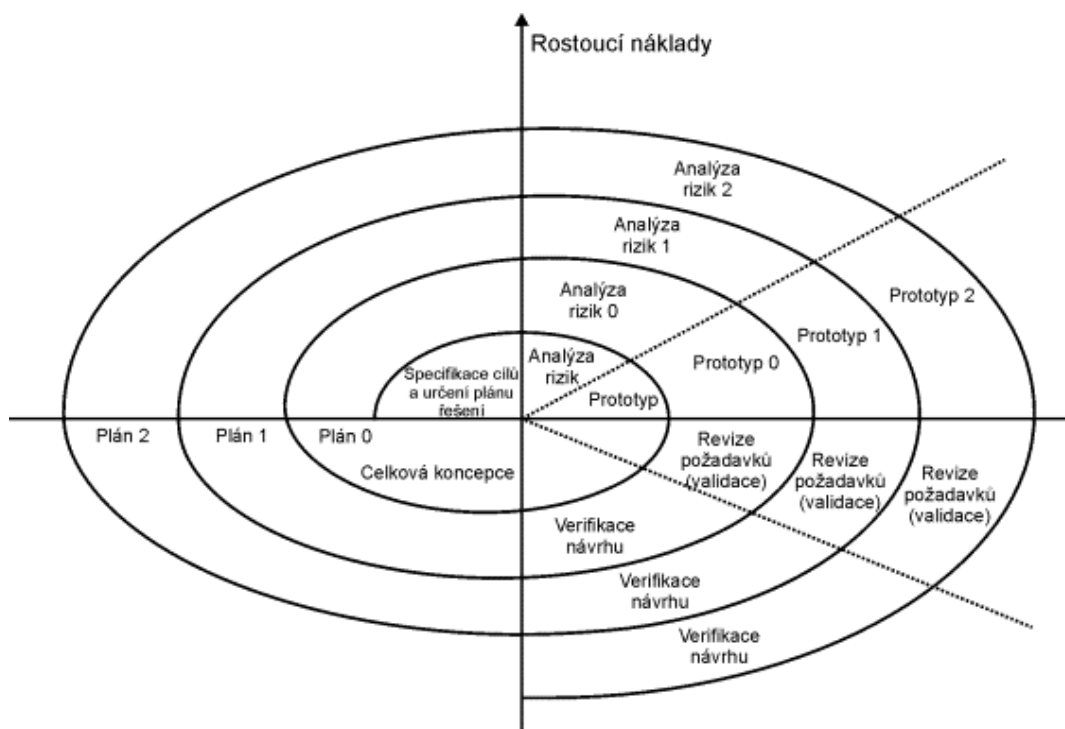
## 2.5.4 Spirálový model

V roce 1988 vytvořil B. W. Boehm, spojením prototypovaného přístupu a analýzy rizik, spirálový model. Základem tohoto modelu je podmínka, aby se jednotlivé kroky vývoje systému opakovaly ve spirále a zároveň měl každý následující krok vyšší úroveň zvládnuté problematiky. S původním vodopádovým modelem má společné vývoje v jednotlivých krocích. Tyto kroky se skládají z několika částí:

- Určení řešené oblasti a možností na její řešení.
- Vyhodnocení jednotlivých možností řešení a analýza rizik spjatých s řešením.
- Tvorba prototypu dané úrovně, jeho odzkoušení a zhodnocení.
- Validace prototypu – správnost jeho funkce.

- Verifikace celého kroku – ověření výstupu s požadavky.
- Plánování dalšího postupu.

Schéma na obrázku 2.6 zobrazuje napojení jednotlivých fází spirálového modelu.



Obrázek 2.6: Spirálový model [9]

Výhodou spirálového modelu je brzká možnost sledování a hodnocení informačního systému. Tím pádem je možno po jednotlivých úsecích konzultovat vývoj se zadavatelem a upravovat IS podle jeho nových požadavků. Při tomto způsobu modelování se používají již ověřené stupně vývoje a díky analýze rizik je zde možnost předcházet chybám.

Jsou zde také nevýhody. Složitost a malá členitost modelu vyžaduje zkušené programátory. Také je při každé iteraci nutná neustálá konfrontace se zadavatelem, takže je tento model nevhodný na vyvíjení systému bez jeho účasti. Z důvodu složitosti taktéž neumožňuje přesné určení časové vytíženosti a finančních nákladů.

U spirálového modelu je klíčovým bodem celého vývoje informačního systému určení a analýza rizik. V případě, že jsou chybně zvoleny komponenty s vysokým rizikem, může jejich pozdní odhalení ohrozit celý vývoj IS.

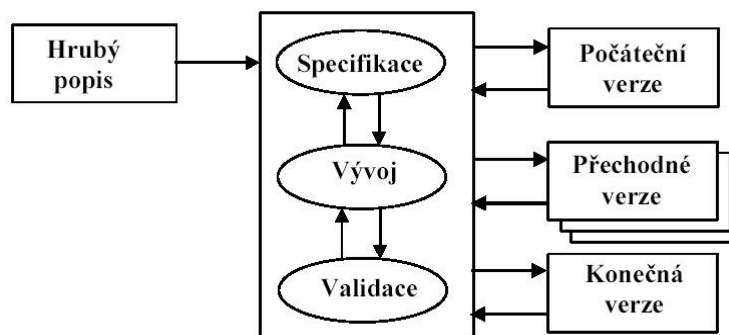
## 2.5.5 Evoluční model

Jedná se model životního cyklu, který vyvíjí informační systém postupně po malých částech. Měněním a upřesňováním požadavků během vývoje pak vzniká celý systém.

Kromě vývoje celého informačního systému se používá také ve dvou dalších oblastech vývoje:

- Při plánovaném vývoji nových verzí původního produktu – upravují se ty části, ve kterých si zadavatel přeje nějaké změny, měl by být zachován původní průběhový model (např. pokud byla první verze systému vytvořena vodopádovým modelem, měly by tak být vytvořeny i verze následující).
- Při údržbě stávajícího produktu – odstraňování chybného chování zjištěného při provozu systému.

Obrázek 2.7 zobrazuje vývoj IS pomocí evolučního modelu.



Obrázek 2.7: Evoluční model [11]

Hlavní výhodou tohoto modelu životního cyklu je jeho využití při nejasnostech v požadavcích, protože využíváme postupného vývoje a údržby. Další výhodou je krátký časový interval mezi řešením dílčího požadavku a jeho zadáním.

Nevýhodou je, především když se jedná o rozsáhlé projekty, velmi obtížné sledování a řízení projektu.

Tento vývoj modelu informačních systému se stal velmi oblíbeným v oblasti objektově orientovaného programování. Nejoblíbenějšími variantami tohoto přístupu jsou inkrementální programování a agilní metodiky.

V této kapitole byly popsány základní poznatky o informačních systémech, co to je informační systém jeho historie a architektura. Také tady bylo podrobněji rozebráno vytvoření informačního systému a jednotlivé fáze vytváření. Nakonec je možné, se zde dozvědět o životním cyklu IS a byly zde podrobněji popsány nejnámější modely těchto životních cyklů. [8, 9, 10, 11, 14, 24]

## 3 Přehled použitých technologií

V kapitole 2.1 byly uvedeny jednotlivé vrstvy architektury informačních systémů. Pro každou z těchto vrstev jsem musel zvolit nějaký implementační prostředek. Datová vrstva je implementována pomocí technologie MySQL. Další použitou technologií je značkovací jazyk HTML, který ve spojení s kaskádovými styly CSS a JavaScriptem tvoří klientskou vrstvu. Pro implementaci poslední aplikační vrstvy je použit skriptovací jazyk PHP.

Implementaci informačního systému musí předcházet jeho návrh. Ten jsem udělal s pomocí technologie UML, což je unifikovaný modelovací jazyk.

### 3.1 Unifikovaný modelovací jazyk UML

Jazyk UML, neboli unifikovaný modelovací jazyk, je v dnešní době jedním z nejlépe propracovaných přístupů k popisu softwarového produktu. Nabízí standardní způsob zápisu návrhu systému od konceptuálního modelu až po jeho implementaci pomocí příkazů programovacího jazyka a databázových schémat.

Historie UML se váže k roku 1994, kdy se vývojem tohoto jazyka začali zabírat Grady Booch a Jim Rumbaugh. Společně začali ve firmě Rational Software, která je nyní součástí IBM, spojovat své metodiky Booch a OMT (Object Modeling Technique). Na konci roku 1995 se k nim ještě přidal Ivar Jacobson se svou metodologií OOSE (Object-Oriented Software Engineering). Výsledkem jejich snažení byl návrh jazyka UML a metodika RUP (Rational Unified Process).

Jazyk UML má bohatou sémantiku i syntaxi. Díky tomu nám usnadňuje návrh a vizualizaci různých druhů aplikací. Protože nám jazyk UML umožňuje modelovat jednoduché i složité aplikace pomocí stejné syntaxe, můžeme své návrhy sdílet i s dalšími návrháři. V tomto jazyce rozlišujeme tři základní stavební bloky:

- **Předměty** – řadíme do nich samotné elementy zpracovávaného modelu, které jsou dále rozděleny do několika podkategorií:
  - **Strukturní předměty** – obsahují statické informace o softwarovém artefaktu a jsou zpravidla vyjádřeny podstatnými jmény (např. třída, komponenta, uzel). V diagramu je zobrazujeme jako obdélníky, čtverce, kružnice. Aby dal diagram smysl, měl by obsahovat alespoň dva tyto strukturní předměty.
  - **Dynamické předměty** – zachycují dynamické chování systému (vzájemnou komunikaci mezi objekty systému).
  - **Předměty vyjadřující seskupení** – sdružují sémanticky příbuzné předměty.
  - **Anotační předměty** – obsahují libovolné množství poznámek k jakékoliv části modelu.

- **Relace** – jsou vztahy, mezi jednotlivými předměty modelu a ukazují, jak spolu tyto předměty souvisí. V UML rozeznáváme tyto typy relací:
  - Asociace – je obecná souvislost předmětů. Za speciální případy asociace považujeme kompozici a agregaci.
  - Závislost – je vztah, při kterém změna v jednom předmětu způsobí změnu v jiném předmětu nebo mu poskytne informace, které od něj požaduje.
  - Generalizace – je druh vztahu, kdy je jeden předmět specifikací jiného předmětu, tzn. že má jisté vlastnosti stejné. Příkladem mohou být třídy nákladní auto a osobní auto, které dědí vlastnosti (jsou specifikací) z nadřazené třídy vozidlo.
  - Realizace – je vztah, při kterém jeden předmět představuje dohodu, za kterou je odpovědný jiný předmět.
- **Diagramy** – zachycují různé aspekty modelovaného systému. Jeden diagram ve vizuální formě zachycuje vždy jeden důležitý aspekt, proto se v projektu používá více různých typů diagramů. Hlavní dělení diagramů je na **statické**, které se věnují struktuře systému a **dynamické**, ty se věnují chování systému.

Pro návrh informačního systému RELAX jsem vytvořil diagram případů užití (Use-Case diagram) a diagram tříd (ER diagram), které jsou blíže popsány ve čtvrté kapitole – Analýza. [12, 15]

## 3.2 Databáze MySQL

MySQL (My Structured Query Language = systém pro řízení databází) je relační databázový systém typu DBMS (database management system). Relační databáze je tvořena jednou či více tabulkami (relacemi), které mají řádky (záznamy) a sloupce (atributy). Sloupce mají svůj název, datový typ a někdy také další specifické vlastnosti (not null, primary key atd.). Tabulky a data jsou zpracovávány pomocí dotazů, které vycházejí z deklarativního programovacího jazyka SQL (Structured Query Language).

Databázový systém MySQL byl vytvořen švédskou firmou MySQL AB. Hlavními autory tohoto systému jsou pánové Michael „Monty“ Widenius a David Axmark. Jelikož se jedná o systém s dvojí licencí (je k dispozici bezplatně pod licencí GPL nebo komerční placenou licencí), je v dnešní době nejrozšířenějším databázovým systémem. Další výhodou je to, že se jedná o multiplatformní databázi což znamená, že se dá snadno nainstalovat na různé operační systémy (MS Windows, Linux, Solaris, FreeBSD i další). Je také rychlý a výkonný a používá se v kombinaci s programovacími jazyky C/C++, Java, Perl, PHP, Python, Tcl, Visual Basic nebo .NET. Nejčastěji se využívá s programovacím jazykem PHP, který umožňuje přístup k datům v databázi, v kombinaci s webovým serverem Apache jako základní software webového serveru.

V současnosti je využívána verze 5.1, ale ke stažení je už i alpha verze 6.0. Jelikož bylo MySQL od počátku optimalizováno na rychlost, scházely zde nejrůznější vlastnosti jako pohledy, triggerů či uložené procedury. Hlavní vlastnosti podporované verzí 5.0:

- Primární klíč – je atribut (někdy i množina), jehož hodnoty jednoznačně určují řádky relace. To znamená, že v jedné relaci nemůžou existovat 2 řádky se stejným primárním klíčem.
- Cizí klíč – je to integritní omezení, pomocí něhož můžeme spojovat různé tabulky přes atributy se stejnými záznamy. Využíváme toho např. tak, že při mazání záznamů z primární tabulky se smažou záznamy také z další tabulky, kde se nám rovná cizí klíč.
- Transakce – je to skupina příkazů, která neporušuje konzistentní stav databáze. To znamená, že jestliže neproběhne jakákoliv část transakce, je celá transakce považována za nevykonanou a všechny zúčastněné zdroje jsou vráceny do svého původního stavu.
- Různé znakové sady a časová pásma.
- Poddotaz – je takový dotaz, jehož výsledky použijeme jako podmínku na pravé straně klausule WHERE. Výsledkem tohoto poddotazu může být buď jedna vrácená hodnota nebo množina hodnot.
- Uložená procedura – je pojmenované seskupení příkazů PL/SQL uložených na serveru. Provádí požadovanou akci nad daty a nevrací žádnou hodnotu.
- Funkce – má za úkol zjistit nebo vypočítat požadovanou hodnotu. Tuto hodnotu pak vrací na místo, kde je funkce volána pomocí názvu funkce.
- Trigger – je programová jednotka, která se automaticky spustí v případě předem definované operace s daty. Například při uložení nového záznamu.
- Pohled – je to uložená definice dotazu. Pohled má své jméno a pro názornost jej chápeme jako tabulku a také s ním tak pracujeme. Narozdíl od tabulky, kde jsou data uložena přímo, je pohled jen předpis, jak se mají data z tabulky vytáhnout. Pohledy používáme v případě práce s velkými tabulkami, aby jsme pracovali jen s výsekem dat, z hlediska utajení a ochrany dat chceme, aby více zaměstnanců vidělo z tabulky jen ty data, které by měli a nakonec, když pracujeme se stejnými daty více způsoby. [3, 12, 16, 25]

### 3.3 Značkovací jazyk HTML

HTML (Hyper Text Markup Language) je značkovací jazyk pro tvorbu webových stránek. Řadí se do skupiny jazyků určených k vytváření stránek v systému World Wide Web, tím umožňuje publikaci obsahu na Internetu. HTML soubor je textový dokument, který lze vytvářet v jakémkoliv textovém editoru a má koncovku .html nebo .htm.

Tento jazyk patří do podskupiny univerzálního jazyka SGML (Standard Generalized Markup Language) a jeho vznik se datuje k roku 1991. V tomto roce ho navrhl Tim Berners-Lee společně s protokolem HTTP (HyperText Transfer Protocol) pro přenos jazyka HTML v počítačové síti.

HTML je charakterizován souborem značek (tagů) a jejich vlastnostmi (atributy). Pomocí značek označujeme text a tím mu určujeme jeho význam, tzn. rozhodneme jestli text bude nadpis, podnadpis, odstavec, tabulka, seznam atd. Dále se pomocí tagů dají vkládat např. obrázky či odkazy. V souvislosti s HTML se můžeme setkat s těmito pojmy:

- Tag – je to shluk písmenek a znaků, které vymezují obsah elementu. Jsou většinou párové a zapisují se do ostrých závorek. Koncový se od počátečního liší tak, že obsahuje ještě zpětné lomítko (př. `<ul>...</ul>` - párový tag označující seznam).
- Element – tvoří základní jednotku HTML stránky. Obsahuje vždy alespoň jeden tag, který může mít upřesňující parametry, dále může obsahovat text nebo další elementy (př. `<h2>Ahoj</h2>` - element, který nám vypíše text Ahoj, podle nastavení párového tagu `<h2>`).
- Parametr – vyskytuje se u počátečního tagu a specifikuje jeho vlastnosti (př. `<table cellpadding="5">` - je to počáteční tag tabulky a jeho parametr cellpadding určuje vzdálenost mezi jednotlivými buňkami tabulky).

V současnosti se používá verze jazyka HTML 4.01, která byla vydána 24. prosince 1999. Touto verzí měl jazyk HTML skončit a být postupně nahrazen novým jazykem XHTML, který využívá univerzální jazyk XML.

7. března 2007 však konsorcium W3C (World Wide Web Consortium), které se společně s veřejností stará o vývoj webových standardů pro WWW, rozhodlo založit novou pracovní skupinu, která má za úkol do roku 2010 uvolnit novou specifikaci HTML 5.0. Jeho finální verze zbavená všech chyb se zatím odhaduje až na rok 2022. [2, 4, 12, 15]

## 3.4 Kaskádové styly CSS

Kaskádové styly CSS (Cascading Style Sheets) se objevují okolo roku 1996, kdy je poprvé vydalo konsorcium W3C jako doporučení k HTML. CSS je soubor metod pro grafickou úpravu stránek napsaných v jazycích HTML, XHTML a XML. CSS znamená česky kaskádové styly. Jazyk se nazývá kaskádový, protože se definice stylů na sebe vrství, ale platí jen ta poslední.

Kaskádové styly nám přináší spoustu výhod:

- Více možností formátování – např. určení okrajů pomocí vlastností margin a padding.
- Stejně styly – u CSS si jednou nadefinujeme vlastnosti stylu pro daný objekt a v HTML už jen používáme daný styl pro objekty na více místech. Díky tomu nemusíme např. u stejných tabulek definovat pokaždé barvu pozadí, zarovnání, odsazení atd., ale určíme jim stejnou třídu (class) stylů, kterou definujeme v souboru .css. Ten následně připojíme k souboru .html (.htm).



- Oddělení vzhledu od obsahu – každá část je psána v jiném souboru a při její změně nemusíme zasahovat do toho druhého.
- Lehčí změna stylu webu – pokud chceme změnit vzhled webu a používáme CSS, upravíme jen jeden soubor se styly. V opačném případě (bez CSS pouze HTML/XHTML) by jsme museli hledat a nahrazovat všechny značky a jejich atributy.
- Rychlejší načítání stránek – soubor .css, který využíváme pro celý web (více souborů .html), se načítá do mezipaměti a pokud není změněn, tak se nahraje pouze jednou. Tímto se běh stránek značně urychlí.

Nevýhodou kaskádových stylů je, že se v různých prohlížečích nezobrazují stejně a je velmi obtížné tento problém vyladit. Situace se zlepšila příchodem IE 7, ale i tak CSS nefungují stejně.

I přes tyto nevýhody při ladění vzhledu webových stránek nám kaskádové styly značně usnadňují a zpřehledňují práci s nimi. V současnosti jsou vydané dvě specifikace CSS1 a CSS2 (plus 2.1). Budoucnost je nejasná. Specifikace CSS3, která má řadu novinek, se stále neobjevuje a dlouho asi ještě ani nebude, o CSS5, která by měla podporovat HTML5, se také přestalo mluvit. Nejnovější zprávou tedy je, že 22. října 2007 se konsorcium W3C rozhodlo vyjádřit souhlas se vznikem specifikace CSS2.2. [4, 5, 12, 17, 18]

## 3.5 Skriptovací jazyk JavaScript

V prosinci roku 1995 byl jako doplněk k jazykům HTML a Java ohlášen společností Netscape a Sun Microsystems JavaScript. Jeho autorem je Brendan Eich. JavaScript je objektově orientovaný skriptovací jazyk, který se v současnosti nejčastěji používá jako interpretovaný programovací jazyk pro tvorbu WWW stránek a je multiplatformní.

JavaScript se používá k dynamické úpravě stránek, spouštění programů reagujících na nějakou událost či třeba na kontrolu správnosti formátu zadaných dat. Jeho kód se zapisuje buď přímo do webových stránek (.html, .php) nebo do externích souborů s příponou .js. JavaScript se nejčastěji spouští na straně klienta, tzn. až po stažení WWW stránek, narozdíl např. od PHP viz. níže.

Charakteristikou jazyka je syntaxe podobná jazykům C/C++/Java. Dále se jedná o jazyk interpretovaný – nemusí se kompilovat, objektový – využívá objektů prohlížeče a je také case sensitivní – rozlišuje v zápisu malá a velká písmena.

Tento jazyk má ale také výrazná omezení. Z důvodu bezpečnosti, protože se spouští na straně klienta, nemůže JavaScript pracovat se soubory, pracuje pouze s cookies. Další omezení je např. že běží pouze v prohlížeči a uživatel ho může zakázat. Jelikož existují různé verze jazyka a prohlížečů, dochází zde k častým chybám. Všechny tyto omezení, hlavně nemožnost přístupu k souborům, z něj dělají doplňkový programovací jazyk pro tvorbu WWW stránek pomocí HTML. [12, 18, 22, 23]

## 3.6 Skriptovací jazyk PHP

PHP (v dnešní době Hypertext Preprocessor, dříve Personal Home Page) je skriptovací jazyk, který se vykonává na straně serveru a vkládá se do HTML kódu. Protože je PHP kód vykonáván na straně serveru, vidí uživatel až jeho výsledky a nemůže si PHP skripty zobrazit, jako to jde s HTML, či je upravit. Aby byly příkazy PHP rozlišeny od HTML kódu, vkládají se do párových tagů `<? a ?>` (nebo `<?php a ?>`).

Historie jazyka se váže k roku 1994, kdy jej vytvořil Rasmus Lerdorf na zobrazení počtu návštěvníků na svých webových stránkách. Uživatelům se toto velmi zalíbilo a žádali a vytvářeli nové skripty. V roce 1998 vyšla verze PHP3, kterou roku 1997 zformovali Zeev Suraski a Andi Gutmans. Byla rychlejší, obsahovala více funkcí a běžela už pod operačním systémem MS Windows. PHP4 jsme se dočkali v květnu roku 2000 a o čtyři roky později i verze 5, která už obsahovala i propracovaný objektově orientovaný přístup. Doposud nejnovější verzí je PHP 5.2.5, která byla zveřejněna 8. listopadu 2007.

Typickými vlastnostmi jazyka je jeho dynamická typovost (tzn. datový typ proměnné je určen přiřazenou hodnotou), jeho pole i indexy jsou heterogenní (tzn. že můžou obsahovat jakékoliv hodnoty) a řetězce jde uzavírat jak do uvozovek tak do apostrofů. V prvním případě je obsah parsován, ve druhé není. Parsování je rozkódování spousty informací z řetězce do patřičných proměnných.

Jelikož je jazyk PHP velmi všestranný a nezávislý na platformě, stal se velmi oblíbeným. Jeho jednoduchost je také v tom, že kombinuje vlastnosti více programovacích jazyků a tím nechává vývojářům jistou dávku volnosti v syntaxi. V kombinaci s databází MySQL a webovým serverem Apache je často využíván k vývoji webových aplikací, jako jsou například informační systémy různých center a firem, internetové obchody atd. V jazyku PHP je například udělán server `email.cz`, `centrum.cz` nebo online encyklopedie Wikipedie. [3, 6, 12, 15, 19, 20, 21]

V této kapitole byly popsány implementační prostředky použité při vývoji informačního systému fitcentra RELAX. Byl zde popsán jazyk UML, pomocí kterého tvoříme návrh IS. Dále jsem popsal technologii MySQL, která vytváří datovou vrstvu. Následující podkapitoly hovořily o jazyku HTML, technologii CSS a JavaScriptu, které vzájemně tvoří klientskou vrstvu. Pro implementaci poslední aplikační vrstvy je použit skriptovací jazyk PHP, který byl popsán v závěru téhle kapitoly.

## 4 Analýza a návrh IS

Na začátku každého informačního systému musí být fáze, zvaná analýza. V této fázi si zadavatel projektu a člověk (nebo firma), který bude projekt vytvářet, ujasní všechny požadavky na daný systém. Tyto požadavky slouží k tomu, aby výsledný funkční IS fungoval tak, jak chce zadavatel.

Před samotnou analýzou a pohovorem se zadavatelem projektu jsem se na internetu podíval na internetové stránky jiných fitcenter, aby jsem získal představu, co všechno by měl takový informační systém obsahovat. Následně jsem svoje poznatky konzultoval se zadavatelem projektu a vypracovali jsme konkrétní požadavky na IS fitcentra RELAX.

### 4.1 Popis požadavků

Z výše zmíněné konzultace se zadavatelem vyplynulo, že chceme jednoduchý přehledný systém, ve kterém se bude uživatel lehce orientovat. Dalším požadavkem bylo vytvoření nějakého uživatelského rozhraní pro zaměstnance, aby mohli lehce přidávat a editovat informace zveřejněné na stránkách bez znalosti HTML kódu či jiného programovacího jazyka.

Seznam základních požadavků:

- oddělení uživatelské sekce od zaměstnanecké z důvodu bezpečnosti,
- registrace uživatelů,
- přihlášení/odhlášení,
- informace o svém účtu,
- základní informace o fitcentru,
- aktuality,
- rozpis cvičení v tělocvičně,
- seznam cvičitelů,
- ceník produktů, které se dají koupit přímo ve fitcentru,
- objednání doplňující výživy od dodavatelů,
- fotogalerie,
- elektronická evidence permanentek,
- sestavování tréninkových plánů,
- chatbox,
- evidence zaměstnanců.

## 4.2 Jednotlivé funkce systému

Po sepsání základních požadavků je nutné shrnout jednotlivé funkce informačního systému. Tyto funkce budou rozděleny podle práv jednotlivých uživatelů systému. Uživatelé systému mohou být návštěvník (neregistrovaný uživatel nebo registrovaný, ale nepřihlášený), přihlášený uživatel, provozní, trenér a majitel.

### Návštěvník může:

- Prohlížet obsah IS:
  - aktuality,
  - chatbox,
  - rozpis tělocvičny,
  - seznam cvičitelů,
  - ceník produktů k dostání ve fitcentru,
  - fotogalerii.
- Registrovat se.
- Přihlásit se.

### Přihlášený uživatel může:

- Prohlížet stejný obsah IS jako návštěvník.
- Spravovat svůj účet.
- Objednat permanentky.
- Spravovat tréninkový plán:
  - zobrazit hotové tréninkové plány a cviky sestavené trenérem,
  - vytvořit vlastní tréninkový plán a zobrazit si ho ve formě pro tisk.
- Objednat vyživovací doplňky a sledovat stav těchto objednávek.
- Prohlížet chatbox a přispívat do něj.
- Odhlásit se.

### Provozní může:

- Prohlížet stejný obsah IS jako návštěvník.
- Spravovat svůj účet.
- Spravovat permanenty:
  - přidávat permanentky registrovaným uživatelům,
  - aktivovat permanentky,
  - upravovat a mazat permanentky.

- Aktualizovat ceník na prodejně – přidávat, upravovat a mazat produkty.
- Aktualizovat nabídku produktů na objednávku – přidávat, upravovat a mazat produkty.
- Spravovat objednávky – měnit stav objednávek a mazat staré.
- Rezervovat tělocvičnu – přidávat, upravovat a mazat rezervace.
- Spravovat aktuality – přidávat, upravovat a mazat aktuality.
- Spravovat chatbox – přidávat příspěvky a mazat nevhodné.
- Odhlásit se.

**Trenér může:**

- Prohlížet stejný obsah IS jako návštěvník.
- Spravovat svůj účet.
- Spravovat cviky – přidávat, upravovat a mazat cviky.
- Sestavovat tréninkový plán pro muže a ženy.
- Prohlížet chatbox a přispívat do něj.
- Odhlásit se.

**Majitel může:**

- Prohlížet stejný obsah IS jako návštěvník.
- Spravovat svůj účet.
- Spravovat aktuality – přidávat, upravovat a mazat aktuality.
- Spravovat chatbox – přidávat příspěvky a mazat nevhodné.
- Spravovat seznam zaměstnanců a cvičitelů – přidávat, upravovat a mazat.
- Odhlásit se.

## 4.3 Modely systému

Diagramy modelů informačních systémů nám pomáhají správně implementovat daný systém na základě jeho požadavků z analýzy. Z diagramů přehledně zjistíme, jaké role zde budou jednotliví aktéři systému mít a jak budou vypadat tabulky a jejich propojení v databázi. Tyto modely jsou velmi důležité pro návrh a následnou implementaci systému.

V současnosti je nejvyužívanějším jazykem pro návrh informačního systému jazyk UML popsáný v kapitole 3.1. Pro návrh projektu IS fitcentra RELAX jsem použil dva diagramy v jazyce UML s jejichž pomocí jsem vytvořil tabulky v databázi a implementoval požadované funkce systému.

Použité diagramy:

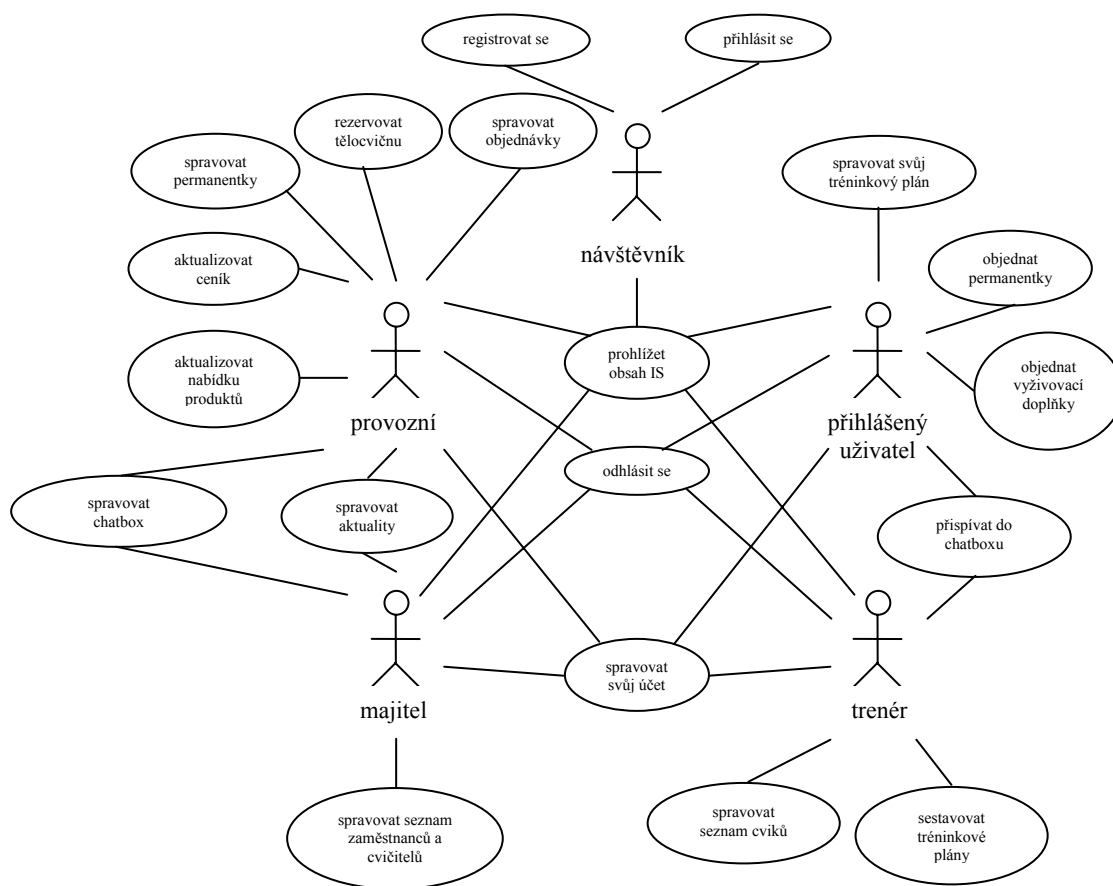
- Use - Case diagram.
- ER diagram.

### 4.3.1 Use - Case diagram

Use - Case diagram, nazývaný také diagram případů užití, je jedním z modelů, který zobrazuje dynamickou strukturu systému z pohledu uživatelů. Zobrazuje chování jednotlivých uživatelů v systému, ale nedefinuje jeho vnitřní strukturu. Diagram případu užití je grafické zobrazení specifikovaných požadavků a funkcí.

Na obrázku 4.1 je zobrazen Use - Case diagram IS fitcentra RELAX. V systému máme 5 aktérů, každý s jinými právy.

Těmito aktéry jsou návštěvník, přihlášený uživatel, provozní, trenér a majitel. Co může daný aktér v systému provádět je podrobně popsáno v kapitole 4.2.

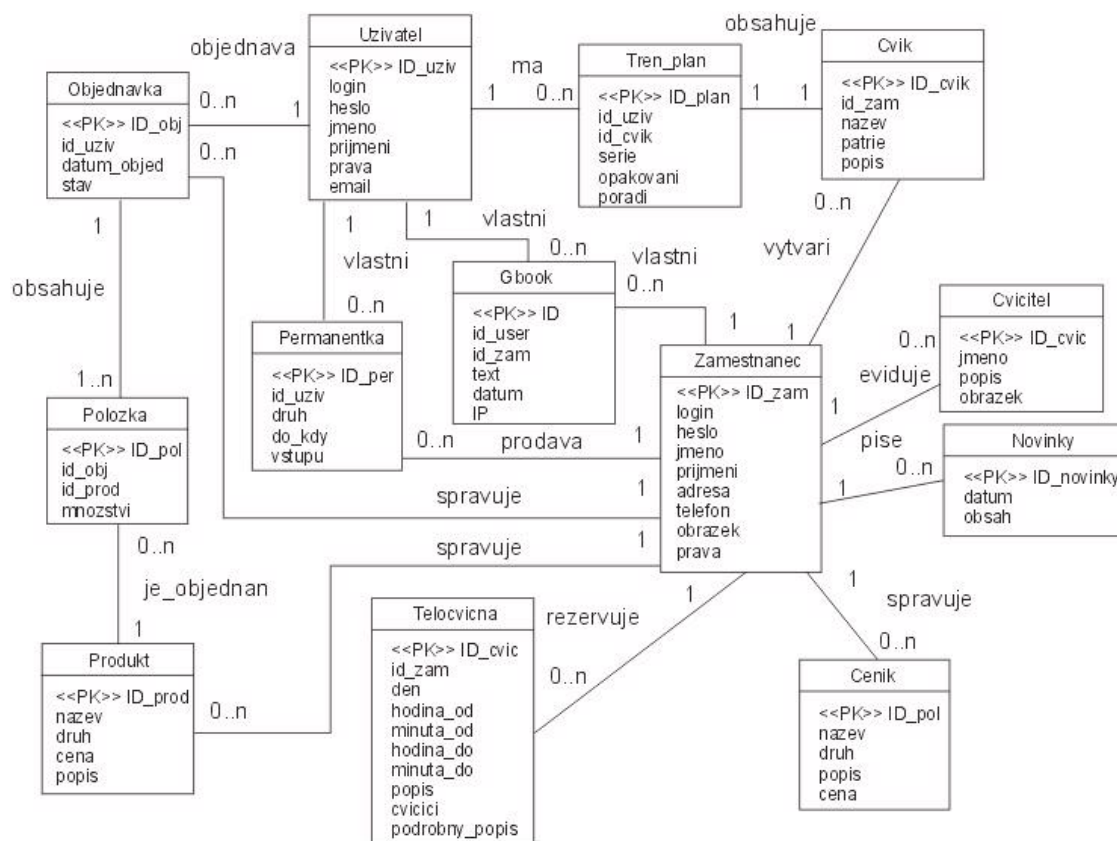


Obrázek 4.1: Use - Case diagram

### 4.3.2 ER diagram

ER diagram, nazývaný také diagram tříd, je model, který reprezentuje datovou strukturu. ER diagram používáme k návrhu databáze a ke komunikaci se záznamníkem. Z ER diagramu je možné jednoduše vyčíst, jaké údaje budeme v databázi uchovávat a jaké budou jednotlivé vztahy mezi nimi. ER diagram obsahuje entity (tabulky), vztahy mezi nimi (kardinalita) a atributy.

Obrázek 4.2 zobrazuje ER diagram pro IS fitcentra RELAX.



Obrázek 4.2: ER diagram

Stručný popis jednotlivých entit:

- Zamestnanec – v tabulce je uložen seznam všech zaměstnanců fitcentra. Každý zaměstnanec má jedinečné ID\_zam, které je primárním klíčem tabulky. K ostatním entitám má vztah s kardinalitou 1:0..n. Tenhle vztah říká, že zaměstnanec může např. vytvořit až n cviků nebo také žádný.
- Uživatel – tato entita obsahuje seznam registrovaných uživatelů. Jejím primárním klíčem je ID\_uziv. S entitami Objednavka, Permanentka, Gbook a Tren\_plan má vztah s kardinalitou 1:0..n.
- Permanentka – reprezentuje tabulku, ve které jsou uloženy záznamy o prodaných permanentkách. Má primární klíč ID\_perm a cizí klíč ID\_uziv podle kterého zjistíme, který uživatel vlastní jaké permanentky.
- Cvicitel – obsahuje seznam cvičitelů, kteří mají pravidelné cvičení v tělocvičně a základní informace k nim. Jsou evidováni příslušnými zaměstnanci.
- Novinky – v tabulce jsou uloženy texty aktualit a datum jejich vložení. Novinky jsou zapisovány zaměstnanci.

- Telocvicna – kromě primárního klíče ID\_cvic má tato tabulka obsahující informace o rezervacích v tělocvičně také cizí klíč ID\_zam. Tento atribut uvádí, který zaměstnanec jakou rezervaci udělil.
- Ceník – jednoduchá tabulka která zaznamenává, jaké produkty jsou k dostání přímo ve fitcentru. Pracovat s touto tabulkou můžou zaměstnanci s vybranými právy.
- Cvik – s tabulkou Zamestnanec je spojena cizím klíčem ID\_zam pro evidenci, který trenér cvik do databáze přidal. Daný zaměstnanec pak může podle tohoto atributu upravovat a mazat jen své cviky. Primárním klíčem tabulky je ID\_cvik.
- Tren\_plan – tato entita má jeden primární klíč ID\_plan a dva cizí klíče. ID\_uziv, který slouží k určení, která položka Tren\_plan patří danému uživateli. Druhý cizí klíč ID\_cvik spojuje entitu Tren\_plan s entitou Cvik. Mezi entitami je vztah s kardinalitou 1:1, tzn. že každá položka Tren\_plan obsahuje právě jednu položku z tabulky Cvik. Atributy serie a opakovani určují, kolikrát se má vybraný cvik opakovat a v kolika sériích by se měl jet. Poslední atribut pořadí určuje, na jakém místě se daný cvik v tréninkovém plánu zvoleného uživatele objeví. Tato tabulka odstraňuje nežádoucí vztah mezi tabulkami Uzivatel a Cvik s kardinalitou m:n.
- Objednavka – v této tabulce jsou uloženy objednávky od uživatelů. Ke komu které patří se pozná podle cizího klíče ID\_uziv. Primárním klíčem tabulky je ID\_obj. K tabulce Polozka má vztah s kardinalitou 1:1..n, tzn. že každá objednávka musí mít alespoň jednu položku.
- Produkt – obsahuje seznam produktů, které se dají přes fitcentrum objednat. Každý produkt má svůj jedinečný ID\_prod, což je v této tabulce primární klíč. S entitou Polozka je spojena vztahem s kardinalitou 1:0..n.
- Polozka – tato tabulka slouží k odstranění vztahu m:n mezi entitami Objednavka a Produkt. Obsahuje dva cizí klíče, které jsou zároveň primárními klíči ve zmíněných tabulkách, svůj primární klíč ID\_pol a atribut množství, jehož pomocí je určeno, jaké množství produktů jednoho typu je obsaženo v objednávce.
- Gbook – tabulka, která obsahuje zprávy v chatboxu. Má 2 cizí klíče a to id\_user a id\_zam podle kterých se určuje, kdo příspěvek vlastní.

V této kapitole jsme se seznámili s analýzou a návrhem daného projektu IS fitcentra RELAX. Byly zde sepsány základní požadavky na funkčnost IS, rozdělili se funkce pro jednotlivé uživatele systému a byly vypracovány dva diagramy, které dále poslouží při implementaci požadovaných funkcí.



# 5 Implementace

V následující kapitole bude popsán vznik datové vrstvy IS fitcentra RELAX a detailní popis implementovaných částí systému. Jsou zde popsány jednotlivé funkce systému, které budou uživatelé IS využívat.

Nejprve bude popsán vznik databázové struktury (datová vrstva), který se vytváří z ER-diagramu. Poté následuje popis klientské a aplikační vrstvy. To zahrnuje popis jak uživatelského rozhraní, tak popis operací s daty v databázi. Je zde napsáno jaké techniky a algoritmy byly použity na vyřešení problémů funkčností všech částí systému, které byly specifikovány v požadavcích na IS fitcentra RELAX.

## 5.1 Vytvoření databáze a tabulek

V IS fitcentra RELAX je databáze implementována pomocí databázového systému MySQL. Databáze je soubor dat, který popisuje část reálného světa. Skládá se z entit (tabulek), což je prvek reálného světa (uživatel, zaměstnanec, ceník atd.) a vztahů mezi nimi. Každá entita má své atributy (sloupce), což jsou sledované vlastnosti prvků reálného světa (např. jméno, příjmení atd.).

Struktura databáze IS fitcentra RELAX je tvořena dvanácti tabulkami, popsanými v návrhu IS v kapitole 4.3.2. Každá entita tohoto diagramu je samostatná tabulka v databázi a všechny vztahy mezi tabulkami, i jejich vlastnosti, jsou implementovány přesně dle tohoto diagramu.

### 5.1.1 Datové typy v databázi

Při vzniku tabulky je nutné definovat, jaké datové typy budou v jednotlivých sloupcích uloženy a jakou budou mít délku. Při vytváření tabulek IS fitcentra RELAX byly použity datové typy VARCHAR, INT, DATE a TEXT. Nejčastěji jsou používány typy VARCHAR na ukládání textu do databáze a INT, což je číselný datový typ. Pro delší text je nutné zvolit datový typ TEXT, protože do VARCHAR je možné uložit max. 255 znaků. Posledním použitým datovým typem je DATE, do kterého se ukládá datum. Délku jednotlivých sloupců jsem volil vhodně podle toho, k čemu je daný sloupec používán. Např. je zbytečné aby měl sloupec prava v tabulce Zaměstnanec při datovém typu INT větší délku než 1, protože číselné hodnoty práv v systému se pohybují od 1 do 4.

### 5.1.2 Klíče a integritní omezení

Další vlastnosti tabulek a jejich sloupců, které se dají nastavit při vytváření tabulky jsou klíče a integritní omezení. Klíče můžeme rozdělit na dva typy a to primární klíč, který jednoznačně identifikuje řádek v tabulce a cizí klíč. Ten se používá na propojení tabulek buď s pomocí jiného cizího klíče nebo přes primární klíč. Díky tomu můžeme při jedné akci pracovat s více tabulkami.

Integritní omezení jsou pravidla, která zajišťují správnost a konzistenci ukládaných dat a hlídají propojení mezi tabulkami. Příkladem integritního omezení je třeba vlastnost PRIMARY\_KEY, což je již zmíněný primární klíč tabulky. Dále to může být např. vlastnost UNIQUE, která zajišťuje unikátnost v tabulce, ale není primárním klíčem (např. login). DEFAULT je hodnota, která se použije, když není vložena do patřičného sloupce jiná hodnota.

Pro jedinečnost primárních klíčů jsem ve všech tabulkách použil jako primární klíč sloupec ID\_ , který má datový typ INT a má modifikátor AUTO\_INCREMENT. Tento modifikátor zvýší hodnotu ve sloupci pro každý nový řádek o 1.

### 5.1.3 Hodnoty NULL

Poslední důležitou vlastností sloupců je možnost definovat, jestli sloupec může obsahovat prázdnou hodnotu NULL nebo jinou implicitně nastavenou výchozí (DEFAULT) hodnotu. K tomu slouží hodnota NULL. Při nastavení atributu sloupce do NULL, nemusí daný sloupec v tabulce obsahovat žádnou hodnotu. Pak obsahuje právě prázdnou hodnotu NULL nebo výchozí hodnotu. Naopak, pokud chceme, aby sloupec musel vždy obsahovat hodnotu jinou než prázdnou, nastavíme atribut NOT\_NULL.

## 5.2 Implementované části systému

V následujících podkapitolách budou blíže popsány jednotlivé implementované části a funkce IS fitcentra RELAX. Obsahem těchto podkapitol bude jak se pracuje s daty, pomocí čeho se zobrazují a jak je řešena komunikace s uživatelem systému. Aplikační vrstva, která pracuje s daty v databázi MySQL, je implementována jazykem PHP.

### 5.2.1 Vzhled IS

Úvodní stránky všech návštěvníků IS jsou stejné. Plocha je rozdělena do třech bloků. Vlevo se vždy nachází menu s odkazy s pevně danou šířkou, napravo jsou partneři fitcentra a počítadlo přístupů. Obě tyto plochy mají světle šedé, nesouměrné pozadí, které je vytvořeno opakujícím se obrázkem v ose y. V těchto částech se nachází několik animací, které jsou udělány dvěma způsoby. Pohybující se postavičky jsou animované obrázky s příponou .gif. Měnění těchto obrázků napravo a měnění bannerů partnerů zajišťuje javascript.

Jako header stránky byl použit animovaný obrázek .gif, který se skládá s fotek fitcentra a tři barevných nápisů. Na pozadí hlavní plochy stránek byl použit světle žlutý podklad, který se opakuje ve všech osách. Jak vypadá základní vzhled IS je možno vidět na následujícím obrázku 5.1:

**Vítáme Vás na webových stránkách fitnesscentra RELAX.**

**NABÍDKA:**

- » Domů
- » Aktuality
- » Chatbox
- » Tělocvična
- » Cvičitele
- » Ceník
- » Fotogalerie
- » Kontakty

**MENU UŽIVATELE:**

nepřihlášen

- » Přihlásit
- » Registrovat

Naše fitcentrum má dvouletou historii, která je vázána k roku 1995 a nabízí široké spektrum služeb. Můžete zde navštívit posilovnu, solárium, tělocvičnu či jen tak si posedět na baru. Vybavení posilovny je v nadstandardní kvalitě, výběr posilovacích strojů byl konzultován s několikanásobným vítězem MS a ME v kulturistice Pavlem Jablonickým, který je zároveň patronem fitcentra.

Ve fitcentru je také možnost nákupu výživových doplňků od smluvních dodavatelů, dále je zde možnost zajistit si cvičení s našim trenérem nebo využít rozsáhlé nabídky cvičení s cvičitelkami v tělocvičně.

Fitcentrum RELAX se nachází v blízkosti centra města v nákupní zóně KASÁRNA na ulici Česká 318 ve 2. patře.

**Otevírací doba:**

po-pá 9:00-22:00  
so-ne+svátky 9:00-12:00 17:00-22:00

Dne	Aktualita
30.4.2008	Upozorňujeme na státní svátky ve dnech 1. a 8. května. Provoz fitcentra bude podle otevíracích hodin víkendů, t.j. 9-12 a 17-22hod. Děkujeme za pochopení. Radka
22.4.2008	Byli jsme nuceni zavést zálohu na klíče od skříněk. Vratná záloha bude 50kč. Děkujeme za pochopení. Aleš.

**PARTNERI**

AMIN:STAR  
isostar go far.

**Návštěvy**

Celkem	35
Týden	6
Dnes	1
Online	1
<b>Pořadí</b>	
Celkem	131518
V kategorii	6279
Pořadno	

Obrázek 5.1: Základní vzhled IS.

## 5.2.2 Registrace

Jednou ze dvou základních možností návštěvníka IS fitcentra je jeho možnost registrace. Tou druhou je pak přihlášení do systému, které bude popsáno v následující kapitole. Pokud není návštěvník WWW stránek přihlášen, má pouze omezené možnosti na stránkách. Může si prohlížet jejich obsah a tím získávat bližší informace o fitcentru. Pokud už je uživatel přihlášen, může si na stránkách objednávat výživové doplňky od smluvních dodavatelů, diskutovat v chatboxu, objednávat si permanentky a sestavovat a prohlížet si tréninkové plány a cviky. Aby se mohl návštěvník přihlásit a využívat tyto výhody, musí se nejprve registrovat.

Samotná registrace probíhá pomocí formuláře. Jak formulář vypadá vidíme na obrázku 5.2. Jelikož je tento IS dělán hlavně pro návštěvníky fitcentra, je při registraci povolena volnost a to, jak v délce loginu či hesla, tak ve formátu zadávaných údajů. Pokud zákazník nechce zadat pravdivé údaje nebo zadává např. místo jména číslice nebo zadá fiktivní email je to jen na něm. Jediný případ kdy je nutno zadávat pravdivé údaje je při objednávkách výživových doplňků a při nákupu permanentek přes IS. Jak se budou tyto akce provádět ve skutečnosti bude zmíněno v daných kapitolách. Majitel fitcentra ale nechtěl, ať jsou nuceni zadávat tyto údaje také návštěvníci, kteří se chtějí probírat jen tréninkovými plány nebo psát v chatboxu.

Při registraci je tedy do formuláře nutné vyplnit povinné údaje v jakémkoliv formátu. Tyto povinné údaje jsou označeny hvězdičkou. Pokud tak uživatel neučiní, je mu vypsána chybová hláška, že nejsou vyplněny všechny povinné údaje a formulář s již zadanými údaji kromě hesel. Tato kontrola

probíhá pomocí struktury if-else stejně, jako všechny následující kontroly v registraci. Poté se kontroluje duplicita loginů a to tak, že jsou pomocí dotazu vybrány již uložené loginy v tabulce databáze uživatelů a podle toho se nastaví hodnota proměnné `neni_duplikace`. Dále je nutné vyplnit v kolonkách Heslo a Kontrola hesla stejné údaje.

Pokud jsou všechny podmínky splněny, je zaslán informativní email pomocí PHP funkce `mail` na vyplněný email s loginem a heslem. Následně jsou pomocí dotazu vloženy zadané údaje do tabulky `Uzivatel` s tím, že `id_uziv` je do tabulky dopisováno automaticky a práva jsou nastavená na 1. Na obrazovce se objeví hláška potvrzující registraci a textový odkaz na přihlášení do systému.

The image shows a registration form with the following elements:

- Title: **Nová registrace**
- Section: **Vložte údaje**
- Note: údaje s hvězdičkou jsou povinné
- Fields: Login \*, Heslo \*, Kontrola hesla \*, Jméno, Příjmení \*, Email
- Button: REGISTROVAT

Obrázek 5.2: Registrační formulář.

### 5.2.3 Přihlášení a odhlášení

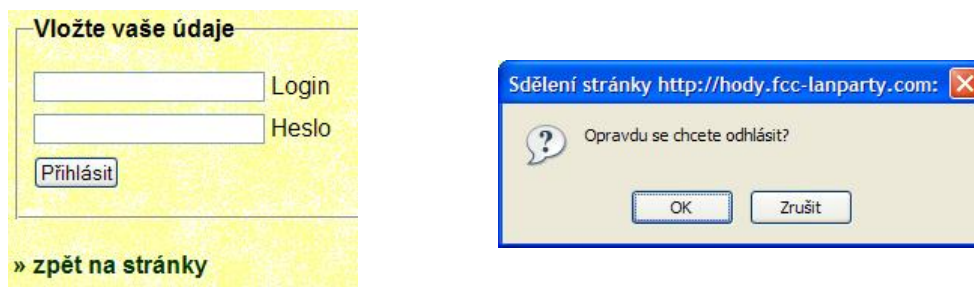
Přihlašování do IS fitcentra probíhá přes formulář, na který se dostaneme odkazem Přihlásit v menu. Přihlašování a kontrolu, jestli je stále někdo přihlášen provádí soubor `kontrola.php`, který je připojován ke každé stránce, kde je potřeba. Na začátku tohoto souboru se inicializuje používání sessions funkcí `session_start()` a zruší se případné přihlášení zaměstnance pomocí funkce `unset($_SESSION["proměnná"])`. Session v protokolu HTTP řeší jeho bezstavovost a díky nim dokážeme udržovat informace o stavu aplikace, tzn. pomocí sessions budeme mít vždy identifikovaného přihlášeného uživatele.

Poté co se smažou případné proměnné od přihlášeného zaměstnance (přihlášení zaměstnanců probíhá na jiné stránce a nechceme mít přihlášeného zároveň zaměstnance i uživatele) nastane kontrola, jestli už byly vyplněny údaje login a heslo v přihlašovacím formuláři. Pokud nastaveny byly, zkontrolují se pomocí dotazu s loginy a hesly v databázi. V případě, že se najde odpovídající záznam, jsou nastaveny superglobální proměnné `$_SESSION["proměnná"]`, které obsahují jestli je nějaký uživatel přihlášen (`$_SESSION["ses_prihlasen"] = true;`), jeho jméno, příjmení, id a práva a uživatel je přihlášen v IS fitcentra RELAX.

Pokud nejsou zadány tyto dva údaje (login a heslo) kontrolujeme, jestli už není uživatel přihlášen pomocí podmínky `if (!isset($_SESSION["ses_prihlasen"]))`. Pokud je přihlášen, dále se nic neděje. V případě že není, zkontroluje se, jestli to je první přihlášení tím způsobem, jestli už byl

odesílán nějaký login. Když se jedná o první přihlášení je vypsán pouze přihlašovací formulář, v opačném případě k tomu ještě přibývá varovná hláška o zadání správných údajů.

Odhlášení ze systému probíhá jednoduše. Po kliknutí na odkaz vyjede pomocí javascriptu potvrzovací okno, zda se chce uživatel opravdu odhlásit. Po potvrzení se provede smazání superglobálních proměnných pomocí funkce `unset($_SESSION["proměnná"])`, které jsou uloženy v souboru `odhlasit.php`. Následně je vypsána hláška, že byl uživatel odhlášen ze systému a po krátké době je automaticky přesměrován pomocí metadat na úvodní stránku. Pole pro přihlášení a odhlášení do/z systému zobrazuje obrázek 5.3:



Obrázek 5.3: Přihlášení a odhlášení do/z systému.

Přihlašování a odhlašování zaměstnanců je prováděno stejným způsobem. Přihlašování probíhá na `/admin.php` (místo `/indexuziv.php`) a loginy s hesly jsou vybírány z jiné tabulky databáze a jinak jsou i pojmenovány superglobální proměnné. Tím je zaručena vyšší bezpečnost přechodu mezi uživatelskou a zaměstnaneckou částí IS. V rootu jsou rozděleny soubory obsahující kód pro jednotlivé uživatele IS do třech složek. Složka `nepr` obsahuje php soubory pro nepřihlášené uživatele, `uziv` pro přihlášené a nakonec `zam` pro zaměstnance.

## 5.2.4 Aktuality, tělocvična, cvičitelé, kontakty

V menu fitcentra může návštěvník získat spoustu informací. V následujících čtyřech podkapitolách přiblížím jak jsou udělány tyto informativní odkazy.

### 5.2.4.1 Aktuality

V aktualitách se můžou návštěvníci stránek dozvědět nejruznější novinky o dění okolo fitcentra (např. náhlá změna provozní doby, výpadek cvičících strojů, nové cvičení v tělocvičně atd.). Jsou zobrazovány v tabulce, do níž se data vkládají pomocí dotazu `SELECT` z tabulky `Novinky` v databázi. Jelikož časem bude počet aktualit narůstat, je k nim přidáno stránkování. Stránkování je nastaveno proměnnou `$zaznamy` po 10. Pomocí `LIMIT` je v dotazu `SELECT` ošetřeno, od kterého záznamu má kolik vybrat. Proměnná `$limitZac` nabývá hodnot podle toho, na které stránce se nacházíme, `$limitKon` je nastavena z počtu záznamů zobrazovaných na stránce. Počet stránek se získává pomocí dělení celkového počtu záznamů, zobrazovanými záznamy. Toto číslo je pak pomocí funkce `ceil()` zaokrouhleno na nejbližší vyšší celé číslo. Následuje cyklus `for`, který vypíše počet stránek s tím, že



kromě aktuální stránky vypíše ostatní jako odkazy. Nakonec se kontroluje, jestli existuje nějaká předchozí nebo následující stránka. Pokud ano, jsou vypsány odkazy na ni.

#### 5.2.4.2 Tělocvična

Na odkazu tělocvična návštěvníci stránek vidí, jaký den a v jakou hodinu probíhá každý týden cvičení s cvičitelkou. Celá stránka je jedna velká tabulka, do které se vkládají údaje pomocí příkazů SELECT na jednotlivé dny. Její vzhled vidíme na obrázku 5.4. Pokud je v tabulce Teloecvicna ve sloupci popis uveden nikdo, znamená to, že v daný den není žádná stálá rezervace. Do tabulky je pak vepsán řádek s touto informací, v opačném případě je tabulka naplněna daty ze SELECT. První sloupec v tabulce je den. Je spojený na tolik řádků, kolik je vybráno záznamů ze SELECT v daný den. Při podržení kurzoru na sloupci cvičení vyjede bližší popis konkrétního cvičení.

TĚLOCVIČNA			
Den	Hodina	cvičení	cvičící
PONDĚLÍ	16:00 - 17:00	DJDM	Renča+Markéta
	18:30 - 19:30	aerobic	Karolína
	19:30 - 21:00	Břišní tance	Hanka
ÚTERÝ	16:30 - 18:00	DDM	Dáda
	19:00 - 20:00	Kalanetika + posilování	Jana
STŘEDA	18:00 - 19:00	aerobic	Karolína
Den	Hodina	cvičení	cvičící

Obrázek 5.4: Vzhled odkazu tělocvična.

#### 5.2.4.3 Cvičitelé a kontakty

Další dvě jednoduché tabulky, která zobrazují cvičitele, kteří mají ve fitcentru objednanou tělocvičnu nebo zaměstnance pracující ve fitcentru. Zvláštností je způsob vytahování fotky z databáze. V databázi je uložen název obrázku a do tabulky je napevno uložena cesta k fotce a mění se už jen její název. Fotky cvičitelů a zaměstnanců se ukládají do jiných složek, takže nemusí být ošetřena duplicita jejich názvů.

### 5.2.5 Ceník

Po kliknutí na odkaz ceník se návštěvníci dozvědí, jaké jsou ceny produktů (např. tyčinek, nápojů, permanentek) na prodej. Tyto věci je možné zakoupit přímo ve fitcentru a nemusí se objednávat. Stránka ceníku je tvořena filtrem a tabulkou, která je vidět na obrázku 5.5. Pomocí filtru je možné

zobrazovat pouze produkty vybraného typu. Druhy zboží, které je možno filtrovat jsou s přihlédnutím na rozšíření sortimentu selektovány z tabulky Cenik. Pomocí atributu GROUP BY jsou stejné druhy seskupeny a ve filtru se zobrazí jen jednou. Vybráním nějakého druhu se vyberou z tabulky pouze údaje zvoleného druhu pomocí podmínky WHERE druh = '\$\_POST[druh]', kdy druh je sloupec tabulky a proměnná \$\_POST[druh] je hodnota filtru. Potvrzení vybrání nějaké hodnoty zajišťuje v IE tlačítko, v ostatních prohlížečích je to javascript onclick ve filtru. Toto zjišťování prohlížeče probíhá pomocí údajů v hlavičce prohlížeče a proměnné \$\_SERVER["HTTP\_USER\_AGENT"]. Pokud akce není filtr nebo je hodnota filtru vše zobrazí se všechny položky.

**CENÍK**

		Druh	Popis	Cena
<input type="text" value="&lt;vše&gt;"/> <input type="text" value="&lt;vše&gt;"/> <input type="text" value="dopl. výživa"/> <input type="text" value="nápoj"/> <input type="text" value="permanentka"/> <input type="text" value="tyčinka"/>		I. výživa	tabletky creatinu	3,-
Amino	dopl. výživa	tabletky amino-star		5,-
ISO star	nápoj	energetický nápoj vhodný během či po cvičení k doplnění energie, 2dl, příchutě dle aktuální nabídky		10,-
Permanentka	permanentka	měsíční		360,-
Permanentka	permanentka	čtvrtletní		1000,-
Permanentka	permanentka	roční		3000,-
Müslityčinka	tyčinka	zdravá tyčinka s ovocnou nebo čoko chutí		15,-
Power	tyčinka	tyčinka s velkým energickým přísunem		30,-

Obrázek 5.5: Vzhled odkazu ceník.

## 5.2.6 Vyživovací doplňky – objednávky

Po přihlášení se jako uživatel je možno využít další služby IS fitcentra RELAX. Jednou z nich je i možnost odkazu na vyživovací doplňky. Při přejítí na tenhle odkaz se uživateli nabídnou tři možnosti.

### 5.2.6.1 Přehled produktů

Přehled produktů funguje stejně jako ceník s tím rozdílem, že jsou zde zobrazovány produkty, které jsou k dostání na objednávku ve fitcentru.

### 5.2.6.2 Moje objednávky

Na tomto odkazu může přihlášený uživatel kontrolovat svoje objednávky. Zjistí tady kdy si co, v jakém množství a za jakou cenu objednal a v jakém stavu jsou jeho objednávky. Ve fitcentru to bude probíhat tak, že na začátku dne shlédne provozní co nového si kdo objednal a podle toho udělá objednávku od dodavatelů. Vlastní objednávky se vybírají nadvakrát. Nejprve jsou při propojení tabulek Objednavka a Uzivatel přes id uživatele v \$\_SESSION["ses\_id"] vybrány objednávky

přihlášeného uživatele a jejich datумы. Pokud nějaká objednávka existuje je vypsána tabulka s hlavičkou, kde je zobrazeno datum objednávky. Následně se pomocí dalšího dotazu vytvoří druhá tabulka, která obsahuje jednotlivé položky dané objednávky. Tyto informace jsou vybírány z propojených tabulek Polozka a Produkt. Tyto tabulky jsou propojeny pomocí id\_prod a vybírají se jen ty položky, u kterých souhlasí id\_obj z předchozího SELECT. Na konec tabulky každé objednávky je připsána její celková cena a její aktuální stav.

### 5.2.6.3 Přidat objednávku

Poslední možností v tomto menu je přidání nové objednávky. K této akci můžeme, ale nemusíme použít filtr, který funguje na výše uvedeném principu. Při přidávání objednávky se vypíše tabulka s produkty, které je možno objednat. Objednává se tím způsobem, že se do tabulky vepíše jaké množství daného produktu chci objednat a potvrdíme to tlačítkem objednat zboží. Toto je uděláno přes formulář. Po odeslání formuláře se vytvoří nová objednávka v tabulce Objednavka s dnešním datem, stavem "byla přijata" a id přihlášeného uživatele. Následně se z databáze vytáhne id poslední objednávky a zjistí se maximální id\_prod v tabulce Produkt. Poté se pomocí cyklu for kontroluje u každého produktu, jestli u něj bylo zapsáno ve formuláři nějaké množství. Pokud ano, je tohle množství, id\_prod a id\_obj zapsáno do tabulky Polozka. Následně se zkontroluje jestli nebyla zapsána prázdná objednávka. V případě že ano, vypíše se varovná hláška a objednávka se z tabulky smaže. V opačném případě je vypsána právně ukončená objednávka podle výše zmíněných dvou SELECT s přidáním LIMIT na jednu objednávku.

O tyto objednávky od přihlášených uživatelů se starají provozní v menu objednávky, které je vidět na obrázku 5.6 a je popsáno v následující kapitole.

**OBJEDNÁVKY**

<všichni>

- <všichni>
- Bolom Jirka
- Hamernik Michal
- Machová Jana
- Petrof Filip

Majitel objednávky <b>Michal Hamernik</b>				
Objednávka z 16.4.2008 <b>se vyřizuje</b> a obsahuje:				
Název	Druh	Cena/ks	Mnozství	Celk. cena
ISOSTAR drink	nápoj	500,-	1	500,-
Celková cena objednávky je: 500,-				
Změnit stav objednávky na: je k převzetí				

Majitel objednávky <b>Michal Hamernik</b>				
Objednávka z 14.4.2008 <b>byla převzata</b> a obsahuje:				
Název	Druh	Cena/ks	Mnozství	Celk. cena
Creatin	tablety	600,-	5	3000,-
Celková cena objednávky je: 3000,-				
Změnit stav objednávky na: smazat				

Majitel objednávky <b>Jana Machová</b>				
Objednávka z 20.2.2008 <b>je k převzetí ve fitcentru</b> a obsahuje:				
Název	Druh	Cena/ks	Mnozství	Celk. cena
Creatin	tablety	600,-	1	600,-
Celková cena objednávky je: 600,-				
Změnit stav objednávky na: byla převzata				

Obrázek 5.6: Vzhled administrátorského menu objednávek.



#### 5.2.6.4 Objednávky

Po kliknutí na toto menu se zobrazí všechny stávající objednávky se stránkováním a filtr pro uživatele, kteří mají nějakou objednávku. V tabulce může provozní měnit stav objednávky z byla přijata na vyřizuje se. Jak se dostanou požadované doplňky do fitcentra změní provozní stav na je k převzetí. Poté co je objednávka uživatelem převzata je změněn na byla převzata. Na konci zúčtovacího období, t.j. když se změní provozní ve fitcentru, si provozní své prodané objednávky přepíše do své účetní knihy a objednávky ze systému vymaže. Pomocí struktury if-else je provoznímu vždy nabídnuta jen jedna změna stavu objednávky podle toho, jakého stavu v daný okamžik nabývá.

#### 5.2.7 Váš účet

Na tomto odkazu je možné upravovat své údaje. Mají k tomu přístup jak přihlášení uživatelé, tak zaměstnanci. Rozdíl je pouze v množství upravovaných údajů. Upravovat se dají všechny údaje vložené do databáze kromě práv. Po kliknutí na odkaz se zobrazí tabulka s údaji přihlášeného uživatele. V posledním sloupci tabulky je možnost upravit. Takto se dá rychle změnit jen jedna hodnota. Vyřešeno je to tak, že při kliknutí na upravit u dané hodnoty se nepřenáší jen akce na upravování záznamu, ale také proměnná \$co, které je naplněna podle měněné hodnoty. Podle ní se pak do formuláře vybere právě ta hodnota, kterou chceme změnit. V případě úpravy loginu se před jeho UPDATE kontroluje jeho duplicita, při zadání prázdného pole se kontroluje, jestli se může jako prázdné uložit. Druhou možností je změna více údajů najednou. Po kliknutí vyjede formulář se všemi možnými změnami a je hvězdičkou označeno, které jsou povinné. V případě nezadání povinného údaje nebo zadání duplicitního loginu je vypsána chybová hláška a zobrazen původní formulář, v opačném případě jsou data úspěšně upravena.

#### 5.2.8 Permanentky

V menu permanentky se přihlášený uživatel dozví, jaké si objednal permanentky, do kdy mají platnost a kolikrát permanentku využil. Uživatel má taky možnost si předem permanentku objednat. Tím ušetří ve fitcentru čas vypisováním permanentky a její zapsání provozní ve fitcentru. Menu permanentek je vidět na následujícím obrázku 5.7.



VAŠE PERMANENTKY		
Druh	Vyprší	Počet vstupů
čtvrtletní	neaktivní	neaktivní
čtvrtletní	17.7.2008	3
měsíční	6.4.2008	15

**OBJEDNAT NOVOU PERMANENTKU**

<vyber typ permanentky>

Obrázek 5.7: Vzhled menu permanentek.

Do permanentek mají přístup také zaměstnanci s právem provozního. Ti po příchodu do tohoto menu vidí tabulku všech permanentek seřazenou podle platnosti a filtr, kde si můžou zobrazit permanentky pouze vybraného uživatele. Provozní můžou permanentku aktivovat, smazat, upravit, přidat novou nebo přidat vstup.

### 5.2.8.1 Aktivovat permanentku

Po kliknutí na aktivaci permanentky je k vybrané permanentce automaticky podle jejího typu vypsána její platnost. Tato doba se spočítá pomocí funkce `now() + INTERVAL`. U již aktivované permanentky (tzn. má už vypsanou dobu platnosti) se vypíše varovná hláška.

### 5.2.8.2 Smazat permanentku

Při mazání permanenty vyskočí ještě potvrzovací okno pomocí javascriptu. Po jeho potvrzení je permanentka smazána z databáze podle id, které je přenášeno pomocí proměnné `$_GET["id"]`.

### 5.2.8.3 Upravit permanentku

Upravovat se dají pouze aktivní permanentky. Při úpravě je možné změnit pomocí formuláře datum platnosti permanentky a je naznačeno, v jakém formátu by se mělo datum uložit.

### 5.2.8.4 Přidat permanentku

Další možností provozního je přidávání permanentek. po kliknutí na odkaz vyjede jednoduchý formulář kde zaměstnanec vybere jednoho z registrovaných uživatelů seřazených podle příjmení a přiřadí mu permanentku požadovaného druhu. Kontroluje se zde, jestli byl někdo a nějaký druh vybrán.

### 5.2.8.5 Přidat vstup

Poslední možností provozního je přidávání vstupů na permanentku. Po kliknutí na ikonu přidání se zjistí, jestli je permanentka aktivní. Pokud není, je vypsána varovná hláška. V opačném případě je zvýšen počet vstupů o 1. Administrátorské menu permanentek je na obrázku 5.8.

Jméno	Příjmení	Druh	Platnost do	Počet vstupů	Možnosti
Michal	Hamerník	čtvrtletní	17.7.2008	3	[edit] [delete] [add]
Michal	Hamerník	měsíční	6.4.2008	15	[edit] [delete] [add]
Michal	Hamerník	čtvrtletní	neaktivní	neaktivní	[edit] [delete] [add]

Obrázek 5.8: Vzhled administrátorského menu permanentek.

Ve skutečnosti by měl systém permanentek fungovat následovně. Každý kdo si bude chtít koupit permanentku, bude muset být registrovaný na webu s platným jménem a příjmením (bude kontrolovat provozní). Zákazníci si budou moci objednávat permanentky sami nebo ve fitcentru přes provozního. Neplatné permanentky budou provozní vždy ke konci týdne mazat. Aktivovat se budou permanentky po zaplacení ve fitcentru. Do současné doby se toto dělalo zapisováním a škrkáním v papírovém bloku.

## **5.2.9 Tréninkový plán**

Tohle menu slouží přihlášeným uživatelům jako možnost bezplatného využití rady trenéra. Můžou se v něm podívat na předem připravený všeobecný tréninkový plán pro muže nebo ženy, projít si seznam cviků i s jejich popisem nebo si z tohoto seznamu cviků, který je vytvořen trenérem sestavit vlastní plán.

Do tréninkového plánu má přístup i zaměstnanec s právy trenéra. Stará se v něm o seznam cviků a o to, jak vypadají obecné plány pro muže a ženy.

### **5.2.9.1 Seznam cviků**

Ať je přihlášen uživatel nebo trenér, vypadá tenhle odkaz podobně. Při přejítí na tento odkaz je vypsána tabulka společně s filtrem. Filtr slouží ke zobrazování cviků podle partií. Pokud je přihlášen trenér, objeví se mu v tabulce se cviky navíc možnosti upravit či smazat cvik. Upravování probíhá přes formulář, kde je nutné zadat název a partii, na kterou je cvik zaměřen. Při volbě smazat se zobrazí potvrzovací okno. Jak u upravování cviku, tak u jeho mazání je kontrola, jestli s cvikem pracuje jeho autor, pokud ne, není akce povolena. Toto se kontroluje pomocí superglobální proměnné přihlášeného zaměstnance a id cviku, se kterým je akce prováděna.

### **5.2.9.2 Tréninkový plán pro muže a ženy, vlastní tréninkový plán**

Pokud přistupuje přihlášený uživatel do tréninkového plánu pro muže či ženy, je mu pouze zobrazena tabulka cviků. Tato tabulka je vybírána z tabulek `Tren_plan` a `Cvik`. Je to provedeno tak, že byl vytvořen uživatel muž a žena, kterým tento tréninkový plán patří. Tuto tabulku vytváří trenér. Trenér může do této tabulky přidávat cvik ze seznamu cviků, upravovat či mazat cviky v plánu a měnit jejich pořadí.

Cvik se do plánu přidává přes formulář, který obsahuje filtr na cviky podle partií a pak list box s cviky ze zvolené partie. Dále může, ale nemusí vyplnit počet sérií a opakování a pozici v plánu. Pozice v plánu je ošetřována tak, aby při nezadání byla vybrána nejvyšší pozice v tabulce a k ní pak přičtena 1. V případě, že je pořadí zadané, tak se následně po uložení do databáze zvýší všechny hodnoty od této nové pozice o 1. Je to uděláno pomocí dotazu, kde se vyberou všechny cviky v `Tren_plan` od vlastníka plánu, použije se hodnota nové pozice a všem cvikům, které mají stejný či vyšší pořadí vyjma toho, který byl právě uložen, se zvýší hodnota pořadí o 1.

Při mazání se zase hodnoty pořadí podle stejného způsobu zmenší o 1. Při úpravě pozice cviku se musí pořadí přepočítat. Nejdříve se zjistí, jestli je nové pořadí větší než staré. Pokud ano, je cvikům v tréninkovém plánu daného uživatele, které mají pořadí větší než bylo staré pořadí a zároveň menší nebo rovno novému pořadí a není to upravovaný cvik, snížena hodnota pořadí o 1. V opačném případě je tomu naopak.

Poslední akcí je možnost posouvat cviky nahoru či dolů tréninkovým plánem. Je tu ošetřena horní a spodní hranice pořadí v plánu. Výměna se děje tak, že je jednomu cviku pořadí o 1 zvětšeno a druhému zmenšeno a naopak.

Při úpravě vlastního tréninkového plánu platí stejné zásady, jen není id uživatele nastaveno na muže nebo ženu, ale je nastaveno přes \$\_SESSION["ses\_id"]. Vlastní tréninkový plán si může uživatel zobrazit i bez grafiky pro případný tisk.

Tréninkový plán pro ženy je vidět na obrázku 5.9.

Název	Partie	Sérii	Opakování	Možnosti	Posunout
Stepper	zahřátí/uvolnění svalů			 	 
Sklopovačky	břicho	3	8	 	  dolů
Tlaky vleže s jednoručkami	hrudník	3	8	 	 
Veslování	záda	3	8	 	 
Činky	ruce	3	8	 	 
Leg-press	nohy	3	8	 	 
Multipress za hlavou	ramena	3	8	 	 
Stepper	zahřátí/uvolnění svalů			 	 

Obrázek 5.9: Menu trenéra na tréninkový plán pro ženy.

### 5.2.9.3 Přidat cvik

Tuhle možnost smějí používat pouze trenéři. Slouží k přidávání cviků, které si pak uživatelé zobrazují nebo vkládají do svých tréninkových plánů. Povinnými údaji jsou název a partie. Při uložení do databáze se automaticky do tabulky ke cviku přidá, kdo je jeho autorem, což se děje opět pomocí \$\_SESSION["ses\_id"].

## 5.2.10 Správa aktualit

K této části menu mají přístup zaměstnanci s právy provozního nebo majitele. Po kliknutí se zobrazí deset posledních aktualit se stránkováním (jako u kap. 5.2.4.1.). V tabulce jsou možnosti upravit a smazat vybranou aktualitu. Při upravování se dá pouze změnit obsah, nedá se měnit datum a je nutné, aby aktualita vždy nějaký obsah měla. U mazání se zobrazuje hláška potvrzující mazání aktuality. Poslední možností v tomto menu je přidání aktuality. Po kliknutí na tenhle odkaz se zobrazí formulář,

do kterého se vepíše obsah novinky. V případě, že je obsah prázdný, je vypsána varovná hláška a zobrazí se aktuality. Při vyplnění a potvrzení tlačítkem je aktualita zapsána do tabulky a je jí pomocí funkce now() přiděleno stávající datum.

### 5.2.11 Zaměstnanci a cvičitelé

Při přihlášení do systému jako zaměstnanec s právy majitele má přihlášený možnost spravovat seznam zaměstnanců a stálých cvičitelů. Oba odkazy fungují velice podobně. Při přejítí na ně se zobrazí tabulka vyplněná buď zaměstnanci nebo cvičiteli. U zaměstnanců je navíc přidán filtr, který má napevno dané hodnoty podle práv. Přes odkazy v tabulce se dají záznamy buď upravovat, při nutnosti zadání všech povinných údajů do formuláře nebo mazat, s potvrzující hláškou. Dále je možné přidávat jak nové zaměstnance, tak cvičitele. U cvičitelů je to pouze jméno, popis a popřípadě název fotky, která bude uložena ve složce cvicitele s příponou .jpg. Při přidávání zaměstnanců pak probíhá po odeslání formuláře kontrola duplicity loginů a kontrola hesla. Fotky zaměstnanců, jejichž název se ukládá do tabulky, musí být ve složce zamestnanci s příponou .jpg. Po přidání zaměstnance se již zaměstnavatel k heslu zaměstnance nedostane, protože při jeho úpravě toto heslo již nevidí.

### 5.2.12 Aktualizace ceníku a produktů

Tyto dva odkazy jsou implementovány naprosto stejně a může k nim přistupovat zaměstnanec s právy provozního. Jediné co se mění je název tabulky, ve které se s daty pracuje. Aktualizace ceníku pomáhá spravovat nabídku produktů, které je možno zakoupit přímo ve fitcentru. Oproti tomu se pomocí aktualizace produktů spravuje nabídka produktů, které si může přihlášený uživatel objednat přes IS. Po kliknutí na jeden z odkazů se zobrazí listbox s výpisem druhů, pomocí kterého je možno produkty filtrovat, a tabulka s produkty. Produkty v tabulce jsou seřazeny podle druhu a ceny. Dále je zde možno přidávat produkty přes formulář (s kontrolou povinných údajů), upravovat produkty nebo je mazat. Formulář pro úpravu produktu je vidět na obrázku 5.10.

**PRODUKTY**

» přidat produkt

» zpět na seznam produktů

**Upravit údaje**

Upravujete údaje

údaje s hvězdičkou jsou povinné

<input type="text" value="Creatin"/>	Název *
<input type="text" value="tablety"/>	Druh *
<input type="text" value="creatinové tablety, 300ks"/>	Popis
<input type="text" value="600"/>	Cena *

Obrázek 5.10: Formulář pro úpravu produktu.

### 5.2.13 Rezervace tělocvičny

V této části menu může zaměstnanec s právy provozního přidávat, upravovat a mazat zarezervované časy tělocvičny. Výpis hlavní tabulky je stejný jako ten, který je uveden v podkapitole 5.2.4.2.

Základ přidávání rezervací je stejný jako v ostatních případech. Jedná se o formulář, ve kterém je nutno vyplnit povinné údaje. Pokud nejsou všechny vyplněny, zobrazí se formulář znovu. V případě, že jsou zadány všechny povinné údaje, je nutné zkontrolovat dobu nové rezervace, aby se nekryla s již zadanou rezervací. K tomu byla použita funkce `mktime()`, která vrátí počet sekund datumu podle zadaných parametrů (hodiny, minuty, sekundy, měsíce, dnu a roku) od data 1. 1. 1970. Jelikož kontrolujeme jen čas v hodinách a minutách, jsou ostatní parametry zadány přes proměnné napevno. Na začátku této kontroly, převedeme pomocí zmíněné funkce, nově zadané hodnoty cvičení, zasílané přes superglobální proměnné z formuláře. Následně kontrolujeme, jestli je doba do větší než od a jestli je zadaná doba v rozsahu otevírací doby fitcentra. Pokud je doba zadaná správně, je provedena kontrola na zadávání první rezervace v daný den. V případě, že je v tabulce řádek s hodnotou popisu nikdo, je tento řádek vymazán. Následně se z tabulky vyberou časy již uložených rezervací v patřičný den a jsou také převedeny na řetězec sekund. Každá jednotlivá rezervace je porovnávána přes tři podmínky křížení rezervací. V případě křížení s jakoukoliv dobou je nastavena proměnná `$pom do 1`. Dalším krokem je kontrola této proměnné. Pokud je rovna 1, vypíše se hláška o překrytí dob rezervace, v opačném případě je nová rezervace uložena do tabulky.

Úprava rezervace probíhá přes formulář, kde je možno ponechat stávající den nebo vybrat nový. Princip kontroly podmínek překrývání rezervací je pak stejný. Při úpravě rezervace cvičení je pak ještě nutno zkontrolovat, jestli se upravovaný údaj nestal první rezervací daného dne nebo jestli nebyl posledním. Tato kontrola probíhá tak, že při `SELECTu` záznamů z nového dne se kontroluje, jestli nějaký záznam nemá popis nikdo. Tento řádek by byl následně smazán. Druhá podmínka kontroluje záznamy z původního dne. Pokud `SELECT` nevybere žádný řádek, je vložen k příslušnému dni řádek prázdný s popisem nikdo.

Při mazání rezervace je nutné pouze zkontrolovat, jestli nebyl smazán poslední údaj dne. Pokud ano, je vložen prázdný řádek s popisem nikdo.

### 5.2.14 Chatbox

Do chatboxu můžou přistupovat všichni návštěvníci IS, ale s různými omezeními.

Nepřihlášený uživatel může pouze sledovat obsah chatboxu. Na začátku výpisu chatboxu se zjistí počet jeho příspěvků a podle toho se vyhotoví stránkování. Následně se z tabulky Gbook vyberou příspěvky v daném rozsahu. Poté se pro každý záznam zjistí, zda ho přidal uživatel nebo zaměstnanec. Pomocí toho se zjistí, ze které tabulky má být jméno a příjmení vybráno. Následně se vypíše tabulka kde je napsáno kdo a kdy příspěvek vložil a text příspěvku. Tento text je ještě nutné přeformátovat kvůli možnosti zapisovat např. tučný či podtržený text nebo možnost vložení html



odkazu či kontaktního emailu. Tohle přeformátování probíhá pomocí funkce Formatovani() definované v \_fce.php.

Pokud je už uživatel přihlášen nebo je přihlášen zaměstnanec, je umožněno přidávání nových příspěvků. To se provádí pomocí formuláře. Ten obsahuje několik tlačítek pro formátování text, textové pole pro zápis příspěvku a dvě tlačítka. Tyto tlačítka slouží buď k přidání příspěvku nebo smazání napsaných znaků do textového pole. Při kliknutí na formátovací tlačítko se zavolá metodou onclick příslušná funkce, která je uložena v formatovani.js. Pokud uživatel klikne na tlačítko přidat a v textovém poli je nějaký text, vloží se příspěvek do tabulky v databázi. Podle toho, jestli se jedná o uživatele nebo zaměstnance je vyplněno dané ID, druhé má hodnotu 0.

Poslední možností těchto uživatelů je citování jiného příspěvku. Pokud je citace vybrána, zjistí se nejprve kdo citovaný příspěvek napsal a pak se z příslušné tabulky vyberou údaje o autorovi a text citovaného příspěvku. Toto je následně vloženo do textového pole. Po tomto následuje kód na vypsaní příspěvků stejně jako u nepřihlášených uživatelů s tím, že je v tabulce možnost reagovat.

Pokud je do systému přihlášen majitel nebo provozní, má navíc možnost mazat nevhodné příspěvky. Toto mazání je umožněno po kontrole práv a v tabulce se možnost smazat objevuje pouze jim. Vzhled chatboxu je na obrázku 5.11.



Obrázek 5.11: Vzhled chatboxu.

## 6 Závěr

Protože se v současné době rozšiřují počítače a internet do většiny domácností i malých firem pro usnadnění práce, bylo by dobré dělat informační systémy i pro menší firmy zaměřené na služby občanům. Jelikož chodím do fitcentra a jsem často svědkem zbytečného hledání a zapisování nejrůznějších údajů do bloků, rozhodl jsem se udělat informační systém fitcentra RELAX. Tento informační systém musí informovat návštěvníky o službách a novinkách ve fitcentru, poskytovat jim nějaké výhody a také musí usnadnit práci zaměstnancům fitcentra. Pro tyto účely byl implementován hlavně elektronický systém permanentek, online vyřizování objednávek a pro zábavu a zvýšení návštěvnosti stránek také chatbox.

Cílem této bakalářské práce bylo vytvoření funkčního prototypu informačního systému na základě zjištěných požadavků od zadavatele projektu. Tyto požadavky byly formulovány před započítím práce a v průběhu analýzy a návrhu systému byly měněny tak, aby co nejvíce vyhovovaly zaběhnuté organizaci ve fitcentru. Po návrhu systému následovala nejnáročnější část projektu, implementace. Systém byl implementován na základě návrhu a výsledkem této implementace byl funkční informační systém, který splňuje všechny požadavky. Jelikož nejsou v současnosti prohlížeče kompatibilní a odlišně interpretují zdrojový kód, musel být v průběhu implementace systém testován pod různými prohlížeči.

V budoucnu, po zavedení systému do provozu, by bylo vhodné stávající systém podle požadavků uživatelů aktualizovat. Zjistí se například, kde je potřeba dát stránkování a kde ne. Dále by bylo možné, pokud by se majitelé fitcentra a přilehlého bowlingového a squashového centra dohodli, rozšířit tyto stránky o zmíněné aktivity. V případě, že by se fitcentrum rozrostlo, by bylo vhodné domluvit se s dodavatelem vyživovacích doplňků ohledně přístupu do správy objednávek, aby už objednávky nemuseli přeposílat provozní.

Vytvořením této bakalářské práce jsem získal znalosti ohledně komunikace se zákazníkem a společného řešení vzniklých problémů. Dále jsem se zdokonalil v programování PHP a naučil jsem se pracovat se vzhledem WWW pomocí kaskádových stylů.



# Literatura

- [1] Lynch, P. J., Hortonová, S.: Základní průvodce webdesignem. Brno: Zoner software, 2004.
- [2] Kosek, J.: HTML – tvorba dokonalých www stránek. Praha: Grada Publishing, 1998.
- [3] Ullman, L.: PHP a MySQL – Názorný průvodce tvorbou dynamických WWW stránek. Brno: Computer press, 2004.
- [4] Holzschlag, M. E.: HTML A CSS – jdi do toho. Praha: Grada Publishing, 2006.
- [5] Staniček, P.: CSS Kaskádové styly – Kompletní průvodce. Brno: Computer press, 2003.
- [6] Brázda, J.: PHP 5 – začínáme programovat. Praha: Grada Publishing, 2005.
- [7] Hruška, T.: Pojem informačního systému [studijní opora IIS]. Brno 2006.
- [8] Široký, J.: Informační systémy. Dokument dostupný na URL:  
[http://homen.vsb.cz/~s1i95/ISVDAS/is/IS\\_uvod.htm](http://homen.vsb.cz/~s1i95/ISVDAS/is/IS_uvod.htm) (cit. 25. 4. 2008)
- [9] Šmíd, V.: Životní cyklus informačního systému. Dokument dostupný na URL:  
<http://www.fi.muni.cz/~smid/mis-zivcyk.htm> (cit. 2008-04-25)
- [10] Pokorný, M.: Vyvíjíme databázový a informační systém II, 12. 5. 2004. Dokument dostupný na URL: <http://www.dbsvet.cz/view.php?cisloclanku=2004051201> (cit. 29. 4. 2008)
- [11] Zendulka, J.: Životní cyklus programového díla. Dokument dostupný na URL:  
[http://www.fit.vutbr.cz/study/courses/PPS/public/pdf/2\\_Ziv\\_cykl.pdf](http://www.fit.vutbr.cz/study/courses/PPS/public/pdf/2_Ziv_cykl.pdf) (cit. 29. 4. 2008)
- [12] Wikipedie – otevřená encyklopedie. Dokument dostupný na URL:  
<http://cs.wikipedia.org> (cit. 29. 4. 2008)
- [13] Datamining – datové sklady a OLAP. Dokument dostupný na URL:  
<http://datamining.xf.cz> (cit. 29. 4. 2008)
- [14] Nešetřil, V.: Fázová organizace projektu a průběhové modely. Dokument dostupný na URL:  
<http://fmfi10.vsb.cz/639/qmag/mj19-cz.htm#m> (cit. 29. 4. 2008)
- [15] Interval.cz – webdesign a e-komerce denně. Dokument dostupný na URL:  
<http://interval.cz> (cit. 29. 4. 2008)
- [16] MySQL.com – The world's most popular open source database. Dokument dostupný na URL:  
<http://www.mysql.com> (cit. 29. 4. 2008)
- [17] Hassman, M.: CSS2.2 bude? Dokument dostupný na URL:  
<http://met.blog.root.cz/0710/css2-2-bude> (cit. 29. 4. 2008)
- [18] Janovský, D.: Jak psát web – o tvorbě internetových stránek. Dokument dostupný na URL:  
<http://www.jakpsatweb.cz> (cit. 29. 4. 2008)
- [19] PHP: Hypertext Preprocessor. Dokument dostupný na URL: <http://www.php.net> (cit. 29. 4. 2008)
- [20] Fogl, J.: Co je PHP a kde ho získat. Dokument dostupný na URL:  
<http://www.pcsvet.cz/art/article.php?id=1958> (cit. 29. 4. 2008)

- [21] Černý, J.: Parsování. Dokument dostupný na URL:  
<http://kam.mff.cuni.cz/~kuba/vyuka/programovani/xaver/parsovani.html> (cit. 29. 4. 2008)
- [22] Sova, J.: JavaScript – seznámení. Dokument dostupný na URL:  
[http://www.builder.cz/art/javascript/js\\_seznameni.html](http://www.builder.cz/art/javascript/js_seznameni.html) (cit. 29. 4. 2008)
- [23] Žďárek, R.: Úvod do JavaScriptu. Dokument dostupný na URL:  
<http://tvorba-webu.zdarek.com/javascript/uvod-do-javascriptu.php> (cit. 29. 4. 2008)
- [24] Merunka, V. Ph.D.: Metody tvorby IS. Dokument dostupný na URL:  
[http://kii.pef.czu.cz/~merunka/documents/for\\_students/U3V/U3V - metody tvorby IS - text.doc](http://kii.pef.czu.cz/~merunka/documents/for_students/U3V/U3V - metody tvorby IS - text.doc)  
(cit. 29. 4. 2008)
- [25] studijní materiály ze střední školy

# Seznam příloh

Příloha 1.: DVD se zdrojovými kódy.