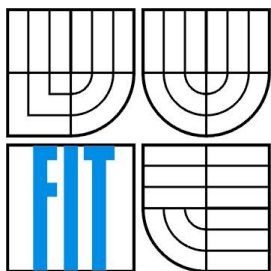


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INFORMATION SYSTEMS

## FILMOVÉ DATABÁZE

FILMING INFOBASES (DATABASES)

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

AUTOR PRÁCE  
AUTHOR

LUDĚK HRADIL

VEDOUCÍ PRÁCE  
SUPERVISOR

Ing. ROMAN LUKÁŠ, Ph.D.

BRNO 2007

## Zadání bakalářské práce

Řešitel: **Hradil Luděk**  
Obor: Informační technologie  
Téma: **Filmové databáze**  
Kategorie: Web

**Pokyny:**

1. Seznamte se s jazyky a prostředky pro tvorbu webových informačních systémů (XHTML, CSS, PHP, Javascript, MySQL).
2. Seznamte se s požadavky kladenými na IS filmové databáze. Rozsah systému konzultujte s vedoucím BP. Požadavky podrobně analyzujte.
3. Proveďte návrh systému. Při analýze požadavků a návrhu využijte vhodných modelovacích technik. Systém musí obsahovat i speciální funkčnost založenou na nějaké metodě z umělé inteligence. Výběr této metody konzultujte s vedoucím BP.
4. Daný systém implementujte.
5. Zhodnoťte dosažené výsledky, porovnejte váš systém s existujícími systémy, navrhnete další možné rozšíření do budoucna.

**Literatura:**

- Kosek, J.: PHP - Tvorba interaktivních internetových aplikací

Při obhajobě semestrální části projektu je požadováno:

- Body 1), 2) a 3)

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 paměťovém médiu (disketa, CD-ROM), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Lukáš Roman, Ing., Ph.D.**, UIFS FIT VUT  
Datum zadání: 1. listopadu 2006  
Datum odevzdání: 15. května 2007

**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: **Luděk Hradil**

Id studenta: 84361

Bytem: Velkomoravská 10, 695 01 Hodonín

Narozen: 24. 02. 1984, Hodonín

(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: Filmové databáze

Vedoucí/školitel VŠKP: Lukáš Roman, Ing., Ph.D.

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 návrhem a realizací informačního systému pro správu a prezentaci filmů a s tím souvisejících dat jako jsou tvůrci, filmové ceny apod. Dále informační systém obsahuje funkce pro správu uživatelů, malý redakční systém pro vkládání článků a další součásti běžné pro webový informační portál. Důraz je kladen na návrh vhodného datového modelu, intuitivní ovládání, přehlednost informací. Účelem celé práce je sběr uživatelských dat a následné spojení těchto dat s faktickými daty. Motivací k této práci mi byla největší světová internetová filmová databáze - IMDb a největší česká filmová databáze - CSFD. Mým cílem bylo vytvořit konkurence schopný systém, k těmto databázím.

## **Klíčová slova**

Databáze, MySQL, PHP, XHTML, CSS, JavaScript, AJAX, web, informační systém, film

## **Abstract**

This baccalaureate work deals with the proposal and realization of an information system for repair and presentation of films and with its related data such as creators, filming awards etc. This information system includes the function for a repair of users, a small editorial system for nesting articles and other single parts common for the web informative portal. The emphasis is on the proposal of a suitable data mock - up, transcendental operating, lucidity information. The purpose of the whole work is a collection of user's data and resulting connection of these data with actual data. Motivation to this work me was biggest world's Internet Filming Database - IMDb and biggest Czech filming database - CSFD. Mine aim be formed competitive system, to these databases.

## **Keywords**

Database, MySQL, PHP, XHTML, CSS, JavaScript, AJAX, web, information system, motion picture

## **Citace**

Jméno Příjmení: Název práce v jazyce práce, bakalářská práce, Brno, FIT VUT v Brně, rok  
Luděk Hradil: Filmové databáze, bakalářská práce, Brno, FIT VUT v Brně, 2007

# Filmové databáze

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Romana Lukáše, Ph.D. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....  
Luděk Hradil  
4.května 2007

## Poděkování

Chtěl bych poděkovat svému vedoucímu bakalářské práce, za odborné rady a nápady, které mi poskytl během psaní této práce.

© Luděk Hradil, 2007.

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

# Obsah

Obsah.....	1
1 Úvod.....	3
2 Teoretická část.....	4
2.1 Internet.....	4
2.1.1 Historie internetu.....	4
2.1.2 Základní služby (protokoly).....	5
2.1.3 Protokol HTTP.....	5
2.2 XHTML.....	6
2.3 Kaskádové styly.....	7
2.3.1 Výhody CSS.....	7
2.3.2 Nevýhody.....	8
2.3.3 Syntaxe.....	8
2.4 PHP - serverová část aplikace.....	9
2.4.1 Princip funkce PHP.....	9
2.4.2 Historie.....	9
2.4.3 Ukázka kódu.....	10
2.4.4 Typické vlastnosti jazyka PHP.....	10
2.4.5 Výhody PHP.....	10
2.5 MySQL databázová vrstva.....	11
2.5.1 Úložiště dat.....	12
2.5.2 Klienti MySQL.....	12
2.6 JavaScript - klientská část.....	13
2.6.1 AJAX.....	13
2.7 Informační systémy.....	15
2.7.1 Architektura.....	16
2.7.2 Internet.....	16
3 Současnost a budoucnost.....	17
3.1 Webové aplikace dnes.....	17
3.2 Budoucnost.....	17
4 Návrh řešení.....	19
4.1 Filmové informační systémy.....	19

4.2 Požadavky na systém.....	20
4.3 Zvolené nástroje.....	21
4.3.1 Webhosting a doména.....	22
4.4 Návrh databáze.....	22
4.5 Struktura webu.....	24
5 Implementace.....	27
5.1 Komunikace s databází.....	27
5.2 Implementační detaily.....	27
5.3 Predikce filmů.....	30
6 Závěr.....	32
6.1 Další rozšíření.....	32
Literatura.....	33
Seznam příloh.....	35



# 1 Úvod

Toto práce si klade za cíl navrhnout a implementovat informační systém pro vkládání, správu a prezentaci filmů. Konečný produkt bude webová aplikace založená na modelu třívrstvé architektury.

Motivací k tomuto projektu mi byly již existující filmové databáze. Např. největší filmová databáze na světě: **IMDb** (The Internet Movie Database), kterou nalezneme na adrese [www.imdb.com](http://www.imdb.com). Dále jsem hledal inspiraci na domácí půdě: **CSFD** (Československá filmová databáze), najdeme ji na adrese: [www.csfd.cz](http://www.csfd.cz) nebo **kFilmu.net**, na adrese [www.kfilmu.net](http://www.kfilmu.net). Mým cílem je konkurovat těmto projektům po stránce funkčnosti, grafické prezentace, uživatelské přístupnosti a v neposlední řadě snadností získávání nejrůznějších dat.

Realizace aplikace bude provedena pomocí skriptovacího jazyka PHP, který zajistí logiku na straně serveru. Pro uložení dat použiji databázi MySQL. Některé dodatečné funkce, které přináší komfort pro uživatele budou implementovány pomocí jazyka JavaScript. Strukturu webu popíšeme pomocí značkovacího jazyka XHTML 1.1, grafický vzhled zajistí kaskádové styly. Výběr těchto technologií a nástrojů je zvolen z důvodů jednoduchosti implementace, dostupnosti a ceny (všechny nástroje jsou zcela zdarma). Tyto technologie jsou v dnešní době nejčastěji používané pro tvorbu podobných systémů.

Práci jsem rozdělil do několika tématických okruhů. První větší kapitola nazvaná Teoretická část, přináší jednoduchý úvod do oblasti internetu a webu. Je zde stručně popsán princip a funkce základních stavebních kamenů informačních systémů tj. jazyka PHP, databáze MySQL a klientské části - jazyka Javascript. Dále se zde věnuji popisu pokročilejších technik jako je AJAX. Nechybí pár slov o značkovacím jazyku XHTML a kaskádových stylech.

V další kapitole nazvané Současnost a budoucnost, se zaměřím na vlastnosti a požadavky dnešních webových aplikací. Také zde najdeme pár slov o budoucnosti značkovacích jazyků.

Následovat bude kapitola popisující návrh databáze a celého systému, zdůvodnění výběru použitých technologií. Tato kapitola dále shrnuje požadavky na filmové informační systémy a požadavky na implementovaný systém.

Předposlední kapitola - Implementace se věnuje popisu zajímavých implementačních detailů, a vysvětlení funkčnosti predikce filmů.

V závěru jsou shrnuty získané poznatky a zhodnocení výsledného produktu.

## 2 Teoretická část

### 2.1 Internet

Při studiu této tematiky jsem čerpal z [1]. Internet je celosvětová soustava vzájemně propojených počítačových sítí. Vznikl na základě počítačové sítě Arpanet, což byl experimentální projekt amerického ministerstva obrany. V roce 1969 došlo k propojení několika počítačů na amerických univerzitách a v roce 1973 došlo k propojení počítačů i mimo USA. Vzhledem k tomu, že síť využívali především studenti, Arpanet se vzdálil od původního záměru sloužit vojenským účelům. Síť se stala plně mezinárodní a začalo se jí říkat Internet.

Internet slouží k přenášení informací a poskytování mnoha služeb, jako jsou elektronická pošta, chat, WWW stránky, sdílení souborů, on-line hraní her, vyhledávání, katalog a další.

World Wide Web (WWW, také pouze zkráceně web), ve volném překladu „Celosvětová pavučina“, je označení pro aplikace internetového protokolu HTTP. Je tím myšlena soustava propojených hypertextových dokumentů.

K prohlížení internetových stránek slouží internetový prohlížeč, který umí zobrazit webové stránky. Nejrozšířenější internetové prohlížeče jsou Internet Explorer, Mozilla Firefox, Opera a Netscape Navigator.

#### 2.1.1 Historie internetu

Převzato z [2].

- 1962 - Vzniká projekt počítačového výzkumu agentury DARPA.
- 1969 - Vytvořena experimentální síť ARPANET, první pokusy s přepojováním uzlů (4 uzly).
- 1973 - Zveřejněn TCP (Transmission Control Protocol).
- 1980 - Experimentální provoz TCP/IP v síti ARPANET, adresace IPv4, protokol DNS, směrovací protokoly.
- 1984 - Vyvinut DNS (Domain Name System).
- 1987 - Vzniká pojem „Internet“.
- 1987 - V síti je propojeno 27 000 počítačů.
- 1989 - Tim Berners-Lee publikuje návrh vývoje WWW (Information Management: A Proposal).
- 1990 - Tim Berners-Lee a Robert Cailliau publikují koncept hypertextu.

- 1990 - Končí ARPANET.
- 1991 - Nasazení WWW v evropské laboratoři CERN.
- 1993 - Marc Andreessen vyvíjí Mosaic, první WWW prohlížeč, a dává ho zdarma k dispozici.
- 1994 - Vyvinut prohlížeč Netscape Navigator.
- 1994 - Internet se komercializuje.
- 1996 - 55 milionů uživatelů.
- 2000 - 250 milionů uživatelů.
- 2005 - 900 milionů uživatelů.
- 2006 - více jak miliarda uživatelů.

## 2.1.2 Základní služby (protokoly)

Převzato z [2].

- SMTP - Simple Mail Transfer Protocol, elektronická pošta, e-mail
- FTP - File Transfer Protocol, přenos vzdálených souborů
- Telnet - Virtuální terminál, vzdálený přístup
- NFS - Network File System, sdílení vzdálených souborů
- DHCP - Dynamic Host Configuration Protocol, dynamická konfigurace síťové stanice
- SNMP - Simple Network Management Protocol, jednoduchý protokol pro správu sítě
- HTTP - Hypertext Transfer Protocol, World Wide Web
- DNS - Domain Name System, překlad doménových jmen

## 2.1.3 Protokol HTTP

Převzato z Wikipedie[3]. HTTP (Hyper Text Transfer Protocol) je internetový protokol určený původně pro výměnu hypertextových dokumentů ve formátu HTML. Používá obvykle port TCP/80, verze 1.1 protokolu je definována v RFC 2616. Tento protokol je spolu s elektronickou poštou tím nejvíce používaným a zasloužil se o obrovský rozmach internetu v posledních letech.

V současné době je používán i pro přenos dalších informací. Pomocí rozšíření MIME umí přenášet jakýkoli soubor (podobně jako e-mail), používá se společně s formátem XML pro tzv. webové služby (spouštění vzdálených aplikací) a pomocí aplikačních bran zpřístupňuje i další protokoly, jako je např. FTP nebo SMTP.

HTTP používá jako některé další aplikace tzv. jednotný lokátor prostředků (URL, Uniform Resource Locator), který specifikuje jednoznačné umístění nějakého zdroje v Internetu.

K protokolu HTTP existuje také jeho bezpečnější verze HTTPS, která umožňuje přenášená data šifrovat a tím chránit před odposlechem či jiným narušením.

Protokol funguje způsobem dotaz-odpověď. Uživatel (pomocí programu, obvykle internetového prohlížeče) pošle serveru dotaz ve formě čistého textu, obsahujícího označení požadovaného dokumentu, informace o schopnostech prohlížeče apod. Server poté odpoví pomocí několika řádků textu popisujících výsledek dotazu (zda se dokument podařilo najít, jakého typu dokument je atd.), za kterými následují data samotného požadovaného dokumentu.

Pokud uživatel bude mít po chvíli další dotaz na stejný server (např. proto, že uživatel v dokumentu klikl na hypertextový odkaz), bude se jednat o další, nezávislý dotaz a odpověď. Z hlediska serveru nelze poznat, jestli tento druhý dotaz jakkoli souvisí s předchozím. Kvůli této vlastnosti se protokolu HTTP říká bezstavový protokol – protokol neumí uchovávat stav komunikace, dotazy spolu nemají souvislost. Tato vlastnost je nepříjemná pro implementaci složitějších procesů přes HTTP (např. internetový obchod potřebuje uchovávat informaci o identitě zákazníka, o obsahu jeho „nákupního košíku“ apod.). K tomuto účelu byl protokol HTTP rozšířen o tzv. HTTP cookies, které umožňují serveru uchovávat si informace o stavu spojení na počítači uživatele.

## 2.2 XHTML

Značkovací jazyk pro popis dokumentů. Byl vyvinut jako nástupce jazyka HTML (HyperText Markup Language), jehož vývoj byl ukončen verzí 4.01. XHTML je zkratkou EXtensible HyperText Markup Language. XHTML je přísnější a čistější než jazyk HTML (např. z něj byly odstraněny některé elementy, které zajišťovaly grafický vzhled).

XML je jazyk, který bude v budoucnosti nositelem kompatibilních informací mezi různými platformami a různými programy (web, wap, databáze...). Cílem XML ovšem není zobrazovat data, ale být nositelem dat, lépe řečeno přesně je popisovat. Pro zobrazení dat na internetu byl vyvinut právě jazyk XHTML. XHTML je optimální prostředek, protože je kompatibilní jak s XML, tak i se současnými prohlížeči.

### XHTML 1.0

- Strict (striktní)
- Transitional (přechodové)
- Frameset (pro práci s rámci)

### XHTML 1.1

- Není tak rozšířený jako XHTML 1.0 a je ještě o něco přísnější.

- Vynechává již prakticky všechny prezentační vlastnosti.

## XHTML 2.0

- Dnes je ještě ve stádiu návrhu. Podle W3C (World Wide Web Consortium - je mezinárodní konsorcium jehož členové společně s veřejností vyvíjejí webové standardy pro WWW) nebude zpětně kompatibilní. Pro korektní zobrazení dokumentů v tomto jazyce bude potřeba příští generace prohlížečů. Více v kapitole 3.

## 2.3 Kaskádové styly

Čerpal jsem z Wikipedie[4]. CSS je zkratka pro anglický název Cascading Style Sheets, česky tabulky kaskádových stylů. Je to jazyk pro popis způsobu zobrazení stránek napsaných v jazycích HTML, XHTML nebo XML. Umožňuje formátovat dokumenty, definuje způsob jejich prezentace na koncových zařízeních, popisuje podobu stránek a styl jednotlivých prvků a přitom nijak neovlivňuje obsah dokumentů samotných.

Jazyk byl navržen standardizační organizací W3C. CSS vzniklo někdy kolem roku 1997. Byly vydány zatím dvě verze specifikace CSS1 a CSS2 (CSS 2.1), pracuje se na verzi CSS3.

Hlavním výhodou je umožnit návrhářům oddělit vzhled dokumentu od jeho struktury a obsahu. Původně to měl umožnit už jazyk HTML, ale v důsledku nedostatečných standardů a konkurenčního boje výrobců prohlížečů se vyvinul jinak. Starší verze HTML obsahují celou řadu elementů, které nepopisují obsah a strukturu dokumentu, ale i způsob jeho zobrazení. Z hlediska zpracování dokumentů a vyhledávání informací není takový vývoj žádoucí. Původním záměrem jazyka HTML byl kromě definování vazeb mezi dokumenty především popis jejich struktury. Určit která část je nadpisem, která odstavcem apod. To jak se tyto objekty mají zobrazit má být popsáno jinde (k tomu je právě vhodné CSS).

### 2.3.1 Výhody CSS

Používání kaskádových stylů ve srovnání se samotným HTML v praxi přináší výhody:

- **rozsáhlejší možnosti**

CSS nabízí rozsáhlejší formátovací možnosti než samotné HTML. Např. pro formátování bloku textu – tj. určení vzdálenosti od jejich elementu či okraje stránky nenabízí HTML nic. CSS má vlastnosti padding a margin.

- **konzistentní styl**

Na všech stránkách webové prezentace by měly být všechny nadpisy stejné úrovně, seznamy, zdůrazněné části textu apod. stejného stylu. S použitím formátovacích možností HTML je to obtížné – u každého objektu v každém dokumentu se vzhled objektu stále znovu nastavuje. S použitím CSS je to velmi jednoduché. Vytvoří se soubor stylu, který se připojuje k HTML dokumentu. Ve všech dokumentech jsou pak objekty stejného vzhledu.

- **oddělení struktury a stylu**
- **dynamická práce se styly**

Provést změnu stylu webu, který pro formátování vzhledu využívá jen možnosti HTML, znamená najít a nahradit všechny značky a změnit atributy mnoha dalších značek. V případě používání CSS znamená změna stylu webu přepsání jediného souboru – souboru stylů.

- **formátování XML dokumentů**
- **větší kompatibilita alternativních webových prohlížečů**
- **kratší doba načítání stránky**

Výhodou CSS oproti starému formátování v HTML je, že kód a obsah webu je uložen v souboru .html a veškerý design a formátování se načítá z jednoho souboru .css, který je většinou společný pro celý web. To znamená, že pokud chceme upravit design webu, stačí změnit pouze jeden soubor .css a změna se aplikuje na celý web. Také se soubor CSS uloží do mezipaměti prohlížeče a pokud není změněn, tak se načítá pouze jednou a zobrazení stránek se velmi urychlí.

### **2.3.2 Nevýhody**

Hlavní nevýhodou CSS je zatím stále špatná podpora v majoritních prohlížečích. Různé prohlížeče interpretují stejný CSS kód jinak a je někdy velmi obtížné jej napsat tak, aby se na všech (resp. na několika vybraných) prohlížečích výsledek zobrazil stejně. Situace se ale v roce 2006 značně zlepšila, v souvislosti s tím se s napětím očekával příchod Internet Exploreru (IE) 7, který by měl postupně vytlačit svého předchůdce IE 6, který byl častým zdrojem problémů. Nicméně ani IE 7 se striktně nedrží definice CSS 2.1.

### **2.3.3 Syntaxe**

Stylový předpis se skládá z posloupnosti pravidel. Každé pravidlo určuje vzhled některého elementu dokumentu, nebo skupiny elementů. Pravidlo začíná tzv. selektorem, který specifikuje („adresuje“) skupinu elementů. Selektor je následován seznamem deklarácí, které určují vzhled vybrané skupiny elementů. Celý seznam je uzavřen ve složených závorkách a jednotlivé deklarace jsou odděleny středníkem.

Podrobnější informace o CSS se dozvíte z [5]. Praktické ukázky naleznete v [6] nebo [7].

## 2.4 PHP - serverová část aplikace

Při studiu jsem čerpal z [8]. PHP (rekurzivní zkratka PHP: Hypertext Preprocessor, původně Personal Home Page) je skriptovací programovací jazyk, určený především pro programování dynamických internetových stránek. Nejčastěji se začleňuje přímo do struktury jazyka HTML, XHTML. PHP lze ovšem také použít i k tvorbě konzolových a desktopových aplikací.

PHP skripty jsou prováděny na straně serveru, k uživateli je přenášen až výsledek jejich činnosti. Syntaxe jazyka kombinuje hned několik programovacích jazyků (Perl, C, Pascal a Java). Obsahuje rozsáhlé knihovny funkcí pro zpracování textu, grafiky, práci se soubory, přístup k většině databázových serverů (MySQL, ODBC, Oracle, PostgreSQL, ...), podporu celé řady internetových protokolů (HTTP, SMTP, SNMP, FTP, IMAP, POP3, LDAP, ...).

V kombinaci s databázovým serverem (především s MySQL nebo PostgreSQL) a webovým serverem Apache je často využíván k tvorbě webových aplikací.

S verzí PHP 5 přichází výrazné zlepšení přístupu k objektově orientovanému programování, zpracování výjimek try/catch, zlepšení podpory XML a webových služeb, SQL Lite.

### 2.4.1 Princip funkce PHP

Procesor jazyka PHP tvoří součást HTTP serveru. Při přijetí požadavku na nějaký dokument HTTP server vyhodnotí typ daného dokumentu. Pokud se jedná o běžný soubor jako např. dokument HTML, obrázek apod. pošle tento soubor přímo jako odpověď klientovi. Pokud se však jedná o soubor PHP, předá jej nejprve procesoru PHP, ten provede vložený kód a vrátí výsledek v HTML, který je teprve poslán klientovi. Více se dozvíte z [13].

### 2.4.2 Historie

Od roku 1994 je PHP jedním z nejpoužívanějších způsobů tvorby dynamicky generovaných WWW stránek. Jeho tvůrce (Rasmus Lerdorf) jej vytvořil pro svou osobní potřebu přepsáním z Perlu do jazyka C. Sada skriptů byla vydána ještě v téže roce pod názvem Personal Home Page Tools, zkráceně PHP.

V polovině roku se systém PHP spojil s programem Form Interpreter stejného autora. Tak vzniklo PHP/FI 2.0. Zeev Suraski a Andi Gutmans v roce 1997 přepsali parser a zformovali tak základ PHP3. Současně byl název změněn na dnešní podobu PHP hypertext procesor. PHP3 vyšlo v roce 1998, bylo rychlejší, obsahovalo více funkcí. Také běželo i pod operačním systémem Windows.

V roce 2000 vychází PHP verze 4, později pak verze 5. Zatím poslední verzí je 5.2.1 z roku 2007.

### 2.4.3 Ukázka kódu

Takto v PHP vypadá program Hello world:

```
<?php
    echo "ahoj světe";
?>
```

### 2.4.4 Typické vlastnosti jazyka PHP

- PHP je interpretovaný, ne kompilovaný jazyk
- Jazyk PHP je dynamicky typový, tzn. že datový typ proměnné se určí v okamžiku přiřazení hodnoty.
- Díky tomu má PHP dva typy porovnání, '==' stejný jako v C, a '===' který platí jen když jsou oba dva výrazy stejného typu.
- Pole jsou heterogenní, mohou tedy obsahovat jakékoli údaje, stejně tak jako jejich indexy.

### 2.4.5 Výhody PHP

Převzato z [9].

- PHP je relativně jednoduché na pochopení
- PHP podporuje širokou řadu souvisejících technologií, formátů a standardů
- je to otevřený projekt s rozsáhlou podporou komunity
- dají se najít kvanta již hotového kódu k okamžitému použití nebo funkční PHP aplikace. Podstatná část z hotového kódu je šířena pod nějakou svobodnou licenci a dá se použít ve vlastních projektech
- PHP snadno komunikuje s databázemi, jako je MySQL, PostgreSQL a řada dalších
- PHP je multiplatformní a lze jej provozovat s většinou webových serverů a na většině dnes existujících operačních systémů



- PHP podporuje mnoho existujících poskytovatelů webhostingových služeb

Další jazyky, které konkurují PHP jsou např: ASP.NET, Java Server Pages (JSP), Perl, ColdFusion, Ruby on Rails a další.

Více o PHP v [10]. O PHP 5 se dočtete v [11].

## 2.5 MySQL databázová vrstva

MySQL je databázový systém, vytvořený švédskou firmou MySQL AB. Poprvé byl vydán na veřejnost v roce 1996. Jeho hlavními autory jsou Michael „Monty“ Widenius a David Axmark. Je k dispozici jak pod bezplatnou licenci GPL (General Public License), tak pod komerční placenou licenci.

MySQL je relační databázový server. Je to multiplatformní databáze. Komunikace s ní probíhá pomocí jazyka SQL<sup>1</sup> (Structured Query Language). Podobně jako u ostatních SQL databází se jedná o dialekt tohoto jazyka s některými rozšířeními.

Pro svou snadnou implementovatelnost (lze jej instalovat na Linux, MS Windows, ale i další operační systémy), výkon a především díky tomu, že se jedná o volně šiřitelný software, má vysoký podíl na v současné době používaných databázích. Dnes existuje po celém světě více než 5 milionů aktivních instalací MySQL. Mezi nejznámější uživatele patří např. Yahoo!, CNET Networks, NASA, Google, Cisco Systems nebo New Yorkská burza.

MySQL bylo od počátku optimalizováno především na rychlost, a to i za cenu některých zjednodušení: má jen jednoduché způsoby zálohování, a až donedávna nepodporovalo pohledy, trigger, a uložené procedury. Tyto vlastnosti jsou již součástí zatím poslední dostupné verze 5.027.

### Oblíbenost MySQL je dána

- flexibilitou – multiplatformní, široká škála API pro nejrůznější jazyky, vícejazyčná podpora, funkce
- výkonem – fulltextové vyhledávání, ukládání dotazů do cache, replikace, bezpečnost
- licenčními možnostmi
- komunitou uživatelů

---

<sup>1</sup> Standardizovaný dotazovací jazyk používaný pro práci s daty v relačních databázích

## 2.5.1 Úložiště dat

MySQL nabízí několik typů databázových tabulek (storage engine), které se liší svými možnostmi, použitím a způsobem ukládání dat do souborů:

- MyISAM - nejpoužívanější, bez podpory transakcí
- InnoDB - podpora transakcí
- BerkeleyDB (BDB)
- MEMORY (dříve HEAP) - práci s daty v paměti
- NDB Cluster - úložiště pro clusterované databáze (od verze 5.0)
- ARCHIVE - komprimované tabulky, bez podpory indexů
- CSV - ukládání dat v prostých textových souborech

## 2.5.2 Klienti MySQL

### mysql

Jedná se o příkazový řádek pro SQL, z něhož se dá ovládat téměř vše na serveru MySQL, včetně vytváření, modifikování, odstraňování databázových tabulek i celých databází, nastavování přístupových oprávnění uživatelů a další. Tento shell je součástí MySQL, a je to mocný nástroj pro veškerá nastavení. Jeho obecná syntax je:

```
mysql [volby] [název_databáze] [argumenty]
```

### phpMyAdmin

Webová aplikace určená pro administraci MySQL, která je napsaná v PHP. Je stabilní (vyvíjí se už od roku 1998), má bohatou paletu schopností. Přeložená do 49 jazyků. Nabízí vysoce optimalizované „klikací“ rozhraní, které značně redukuje možnost chyb zaviněných uživatelem. Je šířen zdarma pod licenci GNU General Public Licence.

Informace jsem čerpal z [10] a [11].

## 2.6 JavaScript - klientská část

JavaScript je multiplatformní, objektově orientovaný skriptovací jazyk, jehož autorem je Brendan Eich z tehdejší společnosti Netscape.

Nyní se zpravidla používá jako interpretovaný programovací jazyk pro WWW stránky, vkládaný přímo do HTML kódu stránky za účelem jejich vylepšení. Jsou jím obvykle ovládány různé interaktivní prvky (tlačítka, textová políčka) nebo tvořeny animace a efekty obrázků. Dále se jej s oblibou používá pro kontrolu formulářů, takže se stránka nemusí zbytečně odesílat na server a posílat zpět klientovi, když zapomene vyplnit některé údaje.

Jeho syntaxe patří do rodiny jazyků C/C++/Java. Slovo Java je však součástí jeho názvu pouze s marketingových důvodů a s programovacím jazykem Java jej vedle názvu spojuje jen podobná syntaxe. JavaScript byl v červenci 1997 standardizován asociací ECMA (European Computer Manufacturers Association) a v srpnu 1998 ISO (International Standards Organization). Standardizovaná verze JavaScriptu je pojmenována jako ECMAScript a z ní byly odvozeny i další implementace, jako je například ActionScript.

Program v JavaScriptu se obvykle spouští až po stažení WWW stránky z Internetu (tzv. na straně klienta), na rozdíl od ostatních jiných interpretovaných programovacích jazyků (např. PHP a ASP), které se spouštějí na straně serveru ještě před stažením z Internetu. Z toho plynou jistá bezpečnostní omezení, JavaScript např. nemůže pracovat se soubory, aby tím neohrozil soukromí uživatele.

Pojem klientský JavaScript vznikne integrací JavaScriptu do internetového prohlížeče. Klientský JavaScript používá objektový model dokumentů DOM. Spojením DOM a JavaScriptu lze dosáhnout dynamického chování webových stránek, pro něž se vžil název Dynamic HTML (DHTML). Čerpáno z [14], [19]

### 2.6.1 AJAX

Webové aplikace implementované pomocí technologie a jazyků uvedených v předchozích podkapitolách obsahují jeden problém, který je oddaluje od klasických desktopových aplikací. Problém je v tom, že vždy když klient (webový prohlížeč) potřebuje další data ze serveru, musí být kvůli znovu načtení stránky vytvořen nový požadavek HTTP, což brzdí aktivitu uživatele. Tento problém odstraňuje právě AJAX.

AJAX je zkratkou výrazu Asynchronous JavaScript and XML (asynchronní JavaScript a XML). Ajax může být chápán jako tzv. vylepšený JavaScript, protože jeho podstata spočívá v tom,

že JavaScriptu na straně klienta je umožněno volat v pozadí server a získat tak potřebná data. Tímto způsobem je možné aktualizovat některé části stránky bez nutnosti opětovně načítat celou stránku. AJAX přichází s řešením pro vyrovnaní zátěže mezi klientem a serverem tak, že jim umožní komunikovat na pozadí, zatímco uživatel pracuje se stránkou. S pomocí AJAXu je otevřená cesta k tvorbě inteligentnějších a pro uživatele lepších webových aplikací. Technologie AJAXu je součástí všech dnešních moderních prohlížečů.

Příklady webových aplikací založených na AJAXu: Google Suggest – našeptávač při hledání, Gmail, Mapy.cz atd.

### **AJAX se skládá z těchto součástí:**

- HTML (nebo XHTML) a CSS pro prezentaci informací
- JavaScript a DOM pro zobrazování a dynamické změny prezentovaných informací
- XMLHttpRequest pro asynchronní výměnu dat s webovým serverem (typicky je užíván formát XML, ale je možné použít libovolný jiný formát včetně HTML, prostého textu)
- Na straně serveru je potřebná technologie ke zpracování požadavků od klienta (např. PHP, ASP)

### **Historie**

Termín AJAX se poprvé veřejně objevil v dubnu 2005 v článku Jesse James Garretta, nazvaném Ajax: A New Approach to Web Applications (Ajax: Nový přístup k webovým aplikacím). Myšlenky, na kterých je AJAX založen, jsou však výrazně starší.

V roce 1998 představil Microsoft novou technologii nazvanou Remote Scripting, ve které v klientském prohlížeči běžel Java applet komunikující se serverem, přičemž tento applet poskytoval služby JavaScriptovým funkcím. Tato technika fungovala v MSIE od verze 4 i v Netscape Navigatoru od verze 4. V páté verzi IE zavedl Microsoft objekt XMLHttpRequest, který v roce 2000 využil v novém programu Outlook Web Access, který poskytuje webové rozhraní pro přístup k e-mailům na Microsoft Exchange Server.

Velká popularita a rozšíření AJAXu začala několika službami společnosti Google (nejdříve Gmail, posléze Google Maps a další).

## Nevýhody

Mezi nevýhody patří hlavně změny v paradigmatu používání webu: webové stránky se chovají jako plnohodnotná aplikace se složitou vnitřní logikou, nikoli jako posloupnost stránek, mezi kterými se lze navigovat i pomocí tlačítek Zpět a Další.

Problémem AJAXových aplikací také může být síťová latence: potřeba komunikace přes Internet má negativní dopady na rychlost odezvy a interaktivitu uživatelského rozhraní. Pokud uživateli není jasně signalizováno, že aplikace zpracovává jeho požadavek (a na pozadí komunikuje se serverem), jediné, co zaregistruje, je zpožděná reakce (mezitím se dokonce může snažit operaci spustit znovu, neboť se domnívá, že systém jeho příkaz ignoroval).

Další nevýhodou AJAXu je nutnost používat moderní grafické prohlížeče, které podporují potřebné technologie.

Vyhledávače nemusí být schopny indexovat všechny části stránky ajaxových aplikací.

JavaScript může být na straně klienta vypnutý, což má za následek nefunkčnost AJAXových aplikací. Proto je nutné vytvářet stránky i pro uživatele bez JavaScriptu. Více se dozvíte z [16].

## 2.7 Informační systémy

Informační systémy jsou systémy pro sběr, udržování, zpracování a poskytování informací a dat. Příkladem informačního systému může být kartotéka, telefonní seznam, účetnictví. Systém nemusí být nutně automatizovaný pomocí počítačů a může být i v papírové podobě.

*Informacemi* jsou sdělení, které odstraňuje nejistotu nebo nevědomost, *daty* míníme jakékoli zaznamenané poznatky či fakta. Jako zvláštní pojem zde vystupuje také *znalost* představující zobecnění poznání určité části reality. Informaci je možno také chápat jako data s nějakým přidaným významem (data + význam). Informace je údaj (množné číslo data), ke kterým si člověk přiřadí význam.

Organizace řízení tvorby a návrhu systému má dnes tyto fáze:

- úvodní studie,
- rozbor zadání,
- analytické modelování,
- systémový design,
- objektový design,

- implementace,
- zkušební provoz,
- nasazení.

## 2.7.1 Architektura

Velmi důležitým hlediskem je volba architektury. Téměř výhradně se používá třívrstvá architektura:

- presentační (interakce s uživatelem),
- funkční (vlastní aplikace, bezpečnost, propojení se světem, kontrola...),
- datová (vlastní data).

Důležitá je i bezproblémová integrace IS, která má dvě hlediska: **vnitřní**, kde jde o proškolení pracovníků, nastavení prostředí a podobně, a **vnější**, kde se jedná zejména o zákazníky a dodavatele.

## 2.7.2 Internet

Internet/intranet poskytuje skvělou platformu pro IS. Dnešní technologie zahrnují například:

- ASP/.NET - platforma Windows
- Java2EE - platforma vyvinutá firmou Sun (technologie např.: JSP, JSF nebo Struts)
- PHP - multiplatformní skriptovací jazyk
- XML - Extensible Markup Language, hraje podstatnou úlohu v dnešních IS, zejména ve výměně dat.

Převzato z Wikipedie[17].

## 3 Současnost a budoucnost

### 3.1 Webové aplikace dnes

Vše dnes směřuje k tomu, že webové aplikace stále více dosahují a budou dosahovat na klasické desktopové. Ať už funkčností, tak svým stylem ovládní. Moderní internetové technologie umožňují vytvářet nejen lepší vzhled, ale také kvalitnější internetové aplikace. Vzhled a způsob ovládní by se neměl podceňovat, protože je to právě zákazník, který si většinou vybírá podle těchto kritérií.

Stále více služeb je distribuováno pomocí webu, a proto je nutné vytvářet web. aplikace více přátelské. Tvorba takového softwaru byla vždy snadnější u desktopových aplikací než u webových. To proto, že web nebyl původně navržen pro komplexní funkcionalitu, ale pouze pro přenos textu a obrázků. Z tohoto důvodu je vyvíjeno mnoho technologií, které web. aplikacím dodávají efektivnosti, přístupnosti a výkonnosti. Např. applety Javy, Macromedia Flash nebo AJAX.

Přenášet funkcionalitu aplikace prostřednictvím webu přináší řadu starostí. Otázkou je proč se zabývat web. aplikacemi a nevěnovat se desktopovým? Webové aplikace se staly mimořádně oblíbené, kvůli následujícím důvodům:

- Levné a jednoduché dodat uživatelům – vše co uživatel potřebuje je webový prohlížeč a spojení do internetu (intranetu)
- Levné a jednoduché aktualizovat – aktualizace na serveru se projeví u všech uživatelů
- Jsou nezávislé na operačním systému
- Umožňují snazší centralizovanou správu dat – ke stejným datům můžeme přistoupit odkudkoliv

### 3.2 Budoucnost

Zde zmíním článek o budoucnosti značkovacích jazyků a jejich možný směr vývoje.

#### **XHTML 2.0 nebo HTML5**

Standard HTML 4.01, který se většinou používá pro tvorbu stránek a web aplikací je 8 let starý. Jeho odvozenina XHTML 1.0 je stará 7 let. Během té doby se změnil pohled na web a vyvstaly nové potřeby. Jsou různé představy o tom, jak situaci řešit.

W3C pracuje na standardu XHTML 2.0, který má za cíl nahradit HTML 4.01 a XHTML 1.0. Potíž je v tom, že jeho vývoj trvá konsorciu už hodně dlouho a navíc ho nevyvíjí jako zpětně kompatibilní a to budou hlavní důvody, proč se k jeho podpoře nepřihlásil žádný výrobce webového prohlížeče. Naopak Mozilla Foundation a Opera Software založili Web Hypertext Application Technology Working Group (WHATWG), kde pracují na vlastním standardu HTML5 (někdy také Web Application 1.0). Později se k nim přidali i Google a Apple.

HTML5 je zpětně kompatibilní a klade si za cíl být lepším základem pro vývoj web aplikací, než je HTML 4.01 a podstatně snadněji implementovatelný než by byl XHTML 2.0.

Poslední vývoj je takový, že WHATWG vyzvalo W3C, aby s nimi zahájilo jednání o použití HTML5 jako základu pro společný postup. Myslím, že W3C bude muset jejich nabídku přijmout, protože bez podpory na straně prohlížečů (a tu zastupuje WHATWG) se žádný standard prosadit nedá.

Článek byl převzat z webu Jana Korbela, který se zabývá budováním webu a web. aplikací [18]



## 4 Návrh řešení

### 4.1 Filmové informační systémy

#### Obecné požadavky na Filmové IS

Každý filmový informační portál nabízí základní paletu funkcí a vlastností. V některých rysech se liší méně v jiných více. Základní požadavek jsou informace o filmu a základních tvůrcích, jako jsou herci a režiséři. Další část tvoří samotní tvůrci a odkazy na filmy na kterých se podíleli. Toto základní provázání tvoří jádro celého systému. Tento základ může být obalen dalšími informacemi.

Mezi další nutné funkce patří správa uživatelů. Každý filmový portál potřebuje získávat data od uživatelů. Jako je hodnocení, které pak udává popularitu filmu. Hodnocení uživatelů se průměruje, a tím udává odpovídající určení „jak moc dobrý film asi je“. Jak je informace přesná záleží na počtu hodnocení daného filmu. Toto se řeší váženým průměrem, který zohlední počet lidí, kteří hodnotili, do celkového hodnocení filmu. Dále přesnost hodnocení závisí na různorodosti uživatelů ať už věkem, pohlavím nebo vzděláním. Jiná data, která se dají od uživatele získat, jsou komentáře, neboli slovní zhodnocení filmu. Dále jsou to příspěvky do diskuzí apod. Jiné filmové portály dovolují uživatelům vkládat různé další informace k filmu: zajímavosti, chyby ve filmech, plakáty apod.

Každý filmový portál musí přehledně prezentovat data v něm obsažená. Samozřejmostí jsou nejrůznější třídící, řadící funkce, které vyfiltrují jen záznamy podle předem vybraných hledisek. Jde o to usnadnit uživateli získání hledaných dat bez toho, aby něco napsal. Důležité jsou právě vyhledávací funkce. Vyhledávače by si při hledání měly poradit i s diakritikou.

Statistiky vznikají po určitém čase provozu systému. Postupem času jsou stále více objektivnější. Na každém filmovém portále najdeme seznamy nejlépe hodnocených filmů, nejhůře hodnocených apod. Zajímavé jsou statistiky počtu přístupů na jednotlivé filmy či tvůrce. Uživatelé by také měli mít možnost sestavit si vlastní žebříček svých nejoblíbenějších tvůrců a filmů. Takto můžeme určit pořadí filmů zase z trochu jiného pohledu. Získáme tím, také statistiky o tvůrcích. Hodnocení tvůrců se na filmových databázích vyskytuje pomálu. Hodnotit tvůrce podle výkonu za veškerou jejich kariéru se může zdát i poměrně zbytečné nebo málo objektivní. To je ovšem věc názoru. Pravdou je, že tyto údaje se velmi těžko získávají.

Přinášet novinky a nejrůznější události ze světa filmu, je to, co uživatele přinutí zavítat zpět na filmový web. Redakční systém musí podporovat každý vyspělejší filmový server. Pohodlí pro redaktory přidávat a vhodně publikovat články, by mělo být samozřejmostí.

Uživatelé potřebují místo, kde by mohli diskutovat o svých filmových zážitcích a o všem možném, kde by rovněž mohli navazovat kontakty. K tomu je potřeba diskuzní fórum. Opět jedna z dalších věcí, která nutí uživatele vrátit se.

To by mělo být základní shrnutí těch nejdůležitějších částí filmového portálu.

## 4.2 Požadavky na systém

V předchozí kapitole jsme si definovali požadavky na obecnou filmovou databázi (filmový portál). Nyní definuji základní požadavky na vytvářený systém v odlišných kategoriích:

### Vzhled a přístupnost

- jednotný vzhled, nezávislý na prohlížeči
- přístupnost i pro uživatele s nejnižším rozlišením
- výběr barevných grafických stylů
- několik druhů menu, měnících se podle vybrané kategorie role uživatele
- nápověda

### Funkce

- profily filmu, tvůrce, ocenění
- vkládání, editace filmů, tvůrců, filmových ocenění
- přiřazování tvůrců k filmu podle povolání, tím se zároveň tvůrcům přiřadí povolání
- v případě vložení herce k filmu, umět určit pořadí, dle důležitosti role a ukládat i názvy rolí, pomocí přetahování myši měnit pořadí herců
- přiřazovat k filmu a k tvůrci ocenění
- přidávat k filmu klíčová slova i s váhou, plakáty, obrázky, odkazy, zajímavosti, chyby, trailery
- vypisovat kompletní tvůrce filmu, ocenění filmu
- vypisovat hodnocení filmu, průběžné pořadí, kdo a jak hodnotil z uživatelů, hodnocení podle nejrůznějších kritérií – podle věku, pohlaví apod.
- v případě tvůrců vypisovat jejich celkovou filmografii, filmografii seřazenou podle různých vlastností, ocenění tvůrce
- přidávat k tvůrcům fotky, zajímavosti, odkazy

- vyhledávací funkce - hledání v názvech filmů, tvůrcích, popisku filmu, klíčových slovech, názvech rolí
- řazení a stránkování všech tabulkových výpisů
- třídění filmů podle roku, žánrů, klíčových slov, jazyků, států, vyhledávání vždy v dané kategorii
- třídění tvůrců podle povolání, plus vyhledávání vždy v dané kategorii
- nejrůznější statistiky a žebříčky

## **Databáze**

- vhodný datový model, lehce rozšiřovatelný

## **Uživatelé**

- kompletní správa uživatelů (registrace, přihlašování, práva, upravování profilu)
- bodování uživatelů
- vkládání si oblíbených uživatelů, pomocí přetahování myši měnit pořadí
- vkládání si oblíbených, tvůrců, filmů
- uživatelé hodnotí filmy, komentují, přispívají do diskuzí k filmům či tvůrcům
- zasílání soukromých zpráv mezi uživateli

## **Další funkce**

- odhad hodnocení filmů
- automatické přepočítání nejrůznějších statistik apod. (CRON – plánování spouštění skriptů)
- diskuzní fórum
- redakční systém – vkládání článků s obrázkem, vkládání klíčových slov, vyhledávání pomocí klíčových slov, editace článků, zveřejňování

Toto je základní výčet funkcí, které musí systém obsahovat.

## **4.3 Zvolené nástroje**

Nástroje a programovací jazyky jsem volil s ohledem na vhodnost pro daný problém, osobní zkušenosti s těmito nástroji a v neposlední řadě jejich bezplatnou dostupnost.

Výsledný produkt bude webová aplikace založená na modelu třívrstvé architektury:

- **Prezentační vrstva** – výstupní značkovací jazyk jsem vybral XHTML 1.1 s kombinací kaskádových stylů. Tento postup se mi jeví, jako aktuálně nejlepší styl tvorby web. aplikací. Kládl jsem důraz na validní výstup a kompatibilitu se všemi současnými internetovými prohlížeči. K tvorbě dalších funkcí rozšiřující web. aplikaci o uživatelské pohodlí jsem použil jazyk JavaScript. Systém bude počítat i s uživateli, kteří mají JavaScript zakázaný a nabídne jim vždy alternativu nezávislou na JavaScriptu. Pro ještě interaktivnější funkce se chystám použít techniky AJAXu, založené na asynchronní komunikaci se serverem pomocí JavaScriptu.
- **Aplikační vrstva** – největší část celého systému, tvořící logiku celé aplikace. Volba byla jasně na straně PHP, protože jsem již pár projektů řešil pomocí toho jazyka. Dále proto, že PHP má velkou základnu příznivců a je tedy snadné nalézt řešení nejrůznějších problémů. V projektu používám PHP ve verzi 5.
- **Datová vrstva** – pro ukládání dat jsem zvolil databázi MySQL 5.027.

### 4.3.1 Webhosting a doména

Informační systém bude veřejně dostupný z internetu. K těmto účelům jsem si objednal webhosting, který podporuje všechny potřebné nástroje a funkce (MySQL 5, PHP 5, Plánování úloh, emailová adresa, prostor 5GB, apod.) nutné k implementaci systému. Aby byl web pro uživatele snadno přístupný a zapamatovatelný pronajal jsem doménu druhého řádu. Název domény jsem vybral nafilmy.com. Web je tedy dostupný z URL: *http://www.nafilmy.com*.

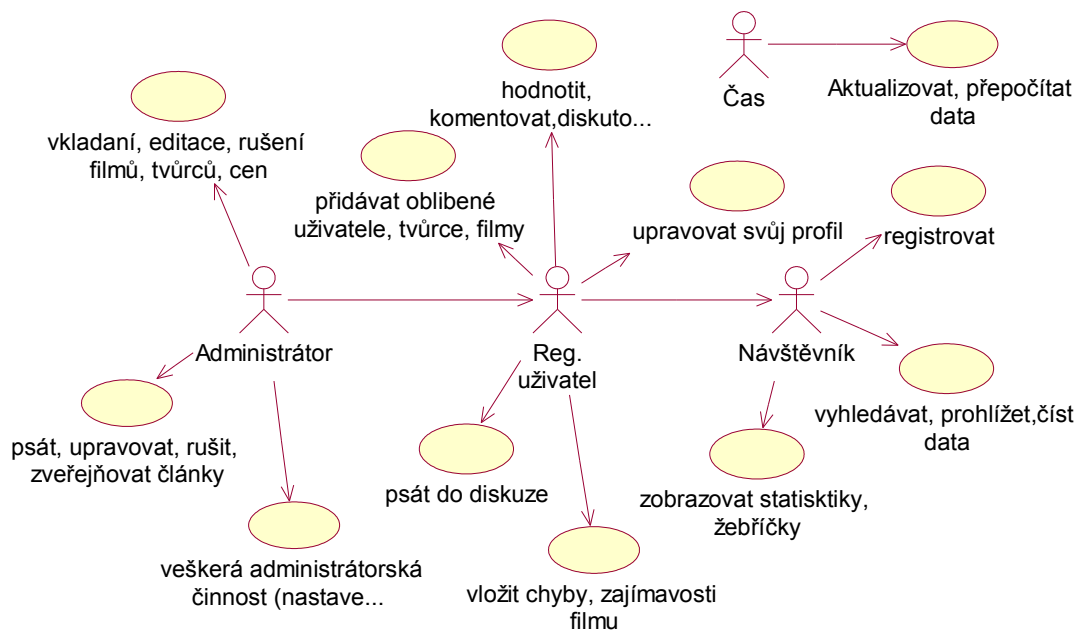
## 4.4 Návrh databáze

Na úvod představím základní role a akce, které mohou být nad systémem prováděny. Vyjadřuje je diagram případů užití (use case diagram) na obrázku 1. Vidíme na něm jednotlivé účastníky, kteří ovlivňují chod systému.

Pro návrh databáze jsem zvolil strukturovaný model návrhu, jehož zdrojem je ER model. ER model je založen na chápání světa jako množiny základních objektů – entit a vztahů. Mezi další části ER modelu patří ještě atributy. Pro kreslení ER diagramu jsem vybral notaci odvozenou z UML<sup>2</sup> (Unified Modeling Language).

---

<sup>2</sup> Je v softwarovém inženýrství grafický jazyk pro vizualizaci, specifikaci, navrhování a dokumentaci programových systémů



Obrázek 1: Use case diagram navrženého systému

## Normalizace tabulek

Hlavní důvody, proč normalizovat návrh:

- opakující se informace (tzv. redundance dat)
- nemožnost prezentovat určitou informaci
- složitá kontrola integritních omezení

Cíle normalizace:

- bezztrátovost dekompozice
- zachování všech závislostí
- dosažení minimálně BCNF (Boyce-Coddova normální forma) nebo 3 NF

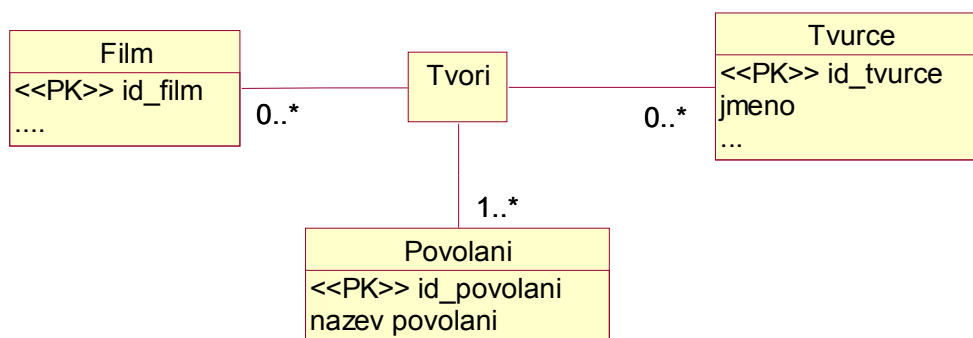
Normální formy:

- 1 NF (první normální forma) vyžaduje, aby všechny atributy nabývaly atomických hodnot (např. odstraněním složeného atributu).
- 2 NF (druhá normální forma) vyžaduje, aby tabulka byla v 1 NF a navíc platilo, že každý neklíčový atribut bude plně funkčně závislý na každém kandidátním klíči dané relace.

- 3 NF (třetí normální forma): vyžaduje, aby neexistoval žádný neklíčový atribut, který je tranzitivně závislý na nějakém kandidátním klíči relace.
- BCNF vyžaduje, aby nebyly funkční závislosti ani mezi atributy složeného klíče

Celá databáze se skládá z 52 tabulek. Všechny tabulky splňují BCNF. Celý ER diagram databáze naleznete jako přílohu této práce. Tabulky se dají rozdělit do několika skupin podle dat, které uchovávají. Jsou to filmy, tvůrci, uživatelé, filmové ceny, články, diskuze atd.

Jádro celé databáze tvoří tři hlavní objekty: film, tvůrce a uživatel. Zbývající objekty se na tyto tři hlavní vztahují. Jako příklad uvedu základní propojení filmu s tvůrcem. Každý film může mít více tvůrců i v různých povoláních (herec, režisér,...). Např. film mohl natočit jeden nebo i více režisérů. Ve filmu většinou hraje více herců apod. Opačně tvůrce může mít ve své filmografii více filmů. Taktéž tvůrce může vykonávat více povolání (zároveň je např. režisér a scénárista). Tento problém vyjadřuje ternární vztah, který vidíme na výseku z ER diagramu databáze na obrázku 2.



Obrázek 2: Ternární vztah, vyjadřující závislost mezi filmem a tvůrcem

## 4.5 Struktura webu

Pro rozvržení celého webu jsem zvolil třísloupcový design, který je pro podobné webové portály běžný a výhodný, protože v sobě dokáže zahrnout několik úrovní navigace a samotný obsah. Skládá se z těchto základních součástí (viz. obrázek 3):

1. **středního sloupce** – tvořícího hlavní obsah stránky
2. **hlavičky** – ta obsahuje dvě horizontální menu, logo, přihlašovací formulář, vyhledávací políčko a další dodatečné informace a ovládací prvky

### 3. patičky

4. **levého postranního sloupce** – obsahuje vertikální menu, různé informace, navíc se zde nabízí vhodné místo pro reklamu
5. **pravého postranního sloupce** – také obsahuje vertikální menu, plus prostor pro reklamu či další rozšíření

Dále můžeme layout rozdělit na tyto části:

6. hlavní menu – toto horizontální menu, slouží k rychlé navigaci
7. administrátorské menu – pro vkládání dat do databáze
8. levé menu – mění se v závislosti na prohlížených stránkách, jedná se o podmenu hlavního menu
9. pravé menu – mění se podle typu uživatelů a podle typu prohlížené sekce, zajišťuje provádění většinou vkladacích akcí s objektem (film – hodnotit film, komentovat film, tvůrce – vložit obrázek apod.).
10. logo
11. přihlašování nebo odhlašování uživatelů, přístup k funkcím spojeným s uživatelským účtem (nastavení, posílání pošty mezi uživateli)
12. vyhledávací dialog s výběrem oblasti, ve které má probíhat hledání
13. položky obsahu stránky – v tomto případě profil filmu (plakát, název filmu, hodnocení, herci)

Systém je navržen tak, aby co nejvíce šetřil čas uživatele. Přihlašovací dialog je přístupný z jakékoliv stránky, taktéž vyhledávání a oznámení o příchozí zprávě.

Rozměry viditelné šířky webu byly stanoveny s ohledem na uživatele s rozlišením 800 x 600 bodů<sup>3</sup>. Toto rozhodnutí je diskutabilní, sám jsem při návrhu váhal, protože uživatelů s tímto rozlišením stále klesá. Pohybuje se kolem 5-7 %, kdežto před 3 lety to bylo asi 34 %. Toto rozlišení časem zmizí, ale stále je tu relativně početná skupina, pro kterou je nutné optimalizovat internetové stránky. I za cenu ztráty prostoru pro obsah. Volba rozlišení zaslouží zvážení všech pro a proti. Vhodným návrhem můžeme tento problém částečně vyřešit. Do pravé části stránky (která se uživateli s nižším rozlišením nezobrazí) vložíme méně důležité prvky webu, které nejsou pro základní manipulaci s webem zapotřebí (např. reklama). Uživatel, tak nebude muset často používat vodorovného posuvníku.

---

<sup>3</sup> Nižší rozlišení v dnešní době, nepovažuji za příliš pravděpodobné



Obrázek 3: Návrh rozložení webu



# 5 Implementace

## 5.1 Komunikace s databází

Ke komunikaci s databází pomocí PHP jsem použil rozšíření mysqli v PHP, což je nástupce staršího rozšíření mysql. Mysqli je součástí PHP až od verze 5. Ke své činnosti vyžaduje min. MySQL 4.1.

Mysqli přináší oproti mysql řadu vylepšení, jako je objektově orientovaný přístup, předpřipravené příkazy, podporu transakcí, ladění dotazů a další.

Ukázka práce s mysqli:

- Spojení s databází a výběr databáze:

```
$db -> new mysqli("hostitel", "jméno", "heslo", "databáze");
```

- Vykonání dotazu:

```
$query = "INSERT INTO tab SET jmeno='Jan', prijmeni='Novák';"
```

```
$db -> query($query);
```

- Uzavření spojení:

```
$db->close();
```

V projektu jsem vytváření připojení řešil pomocí návrhového vzoru Singleton. Návrhový vzor Singleton zajišťuje, že třída má vždy jen jednu instanci a poskytuje globální přístupový bod k této instanci. Čili vytvořením instance třídy mám zaručenu její unikátnost. Díky tomu zajistím pouze jedno připojení k databázi. Spojení s databází uzavírám v destrukturu třídy.

## 5.2 Implementační detaily

### Základní vykreslení

Výstupem aplikace je validní XHTML 1.1 webová stránka. Její generování zajišťují třídy `Page_top` a `Page_bottom`. První třída se stará o vykreslení horní části webu. Třída obsahuje veřejné metody pro definování různých vlastností stránky. Např. metodu pro určení položek jednotlivých menu, titulku stránky, import zadaných kaskádových stylů, souborů JavaScriptu apod. Druhá třída zase generuje patičku stránky. Použití tříd pro vykreslení základní struktury webu mi přišlo výhodné, protože každá stránka může mít jiné vlastnosti a obsah. Pokud chci změnit styl zobrazení stránky, využiji dědičnosti a tím pádem jednoduše dosáhnu kýženého výsledku.

## Přihlašování

Přihlašování a odhlašování uživatelů je zajištěno třídou `Login`. Obsahuje metody pro přihlášení, odhlášení, odhlášení neaktivních uživatelů, aktualizaci informací o přihlášeném uživateli apod.

## Vyhledávání

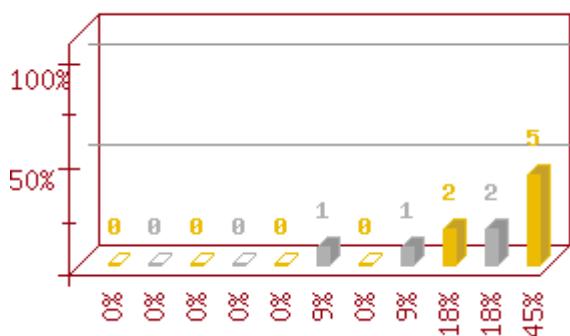
V systému je zabudováno několik různých typů vyhledávání a na různých místech. Globální vyhledávání přístupné odkudkoliv, obsahuje výběr prostoru hledání. Můžeme tak vyhledávat názvy filmů, tvůrce, v popisku filmů, v názvu rolí a podle klíčových slov. Vyhledávač si poradí i s diakritikou a velikostí písmen, navíc můžeme hledat jen podle části hledaného výrazu. Z důvodů nalezení menší a přesnější množiny výsledků je nutné zadat alespoň tři znaky.

Princip funkce: hledanému řetězci odstraním diakritiku a převedu jej na malá písmena. Začnu získávat položky z databáze podle vybraného prostoru hledání. Ještě než začnu porovnávat upravený hledaný výraz, připravím pro porovnání také řetězec získaný z databáze (odeberu z něj diakritiku a převedu na malá písmena). Až nyní začínám porovnávat.

Hledání v popisku představuje fulltextové vyhledávání, které zajištěno pomocí MySQL. Stačí vytvořit fulltextový index nad daným sloupcem, ve kterém chceme vyhledávat. Zbývá už jen položit speciální dotaz.

## Generování grafu a úprava obrázků

GD je open source grafická knihovna pro dynamické vytváření obrázků. Knihovna je přístupná pro PHP nebo Pearl. GD vytváří PNG, JPEG, GIF a další formáty obrázků. Obvykle se používá pro generování grafů, náhledů obrázků apod. Ve své práci jsem ji využil na generování grafu, kvůli přehlednějšímu zobrazení statistik hodnocení. Ukázka grafu je na obrázku 4.



Obrázek 4: Vygenerovaný graf pomocí grafické knihovny GD

Dále jsem využil tuto knihovnu pro úpravu vkládaných obrázků do databáze. Každý vložený obrázek upravím tak, že mu změním rozměry (při zachování poměru stran). Je to výhodné v případě, že uživatel vloží velký obrázek ve vysoké kvalitě, který by tak zabíral zbytečně místo.

## **Našeptávač při vyhledávání**

Na technice AJAXu je založen tzv. našeptávač. Při hledání automaticky doplňuje zadávaný výraz a tím urychluje práci vyhledávání. Funkce je založená na tom, že po zadání znaku do vyhledávacího pole se odesílá na pozadí asynchronní požadavek na server. Ten provede vyhledání v databázi podle zadaných znaků a vrátí v podobě XML seznam odpovídajících položek. Ten je vykreslen pomocí JavaScriptu do připravených div elementů. Poté přijde na řadu CSS, pomocí kterého graficky zformátují nabídku s odkazy na nalezené položky.

## **Kontrola formulářů**

Další vylepšení založené na AJAXu. Automatické kontrola formulářů. Opět urychluje práci se systémem tím, že v případě špatně vyplněné (nebo nevyplněné) položky formuláře, ihned vyvolá chybové hlášení. Uživatel chybu opraví a formulář odesílá až když je správně vyplněn. Odpadá tak čekání na odeslání a případné znovu načtení formuláře. Výhoda je v tom, že veškerá ověřovací logika je jen na straně serveru a pomocí asynchronního volání na pozadí dochází k průběžné kontrole. Pokud není zapnut JavaScript formulář používá jen kontrolu na serveru pomocí PHP, dostupnou až po odeslání formuláře.

## **Posunování myši**

Opět velmi užitečná funkce založená na AJAXu a JavaScriptové knihovně (<http://script.aculo.us>) pro přetahování objektů na stránce myši. Tuto techniku využívám na dvou místech. Jednak pro řazení pořadí herců (viz obrázek 5) u konkrétního filmu a pro určování pořadí oblíbených uživatelů.

Pomocí myši uchopíme položku a táhnutím nahoru nebo dolů změníme pořadí. V případě, že chceme položku odstranit přesuneme ji do koše (červeně ohraničená oblast).

## Tvůrci k filmu Kmotr

Zadejte jméno tvůrce

Vyhledat

1 Brando Marlon Corleone [X]  
2 Pacino Al Michael Corleone [X]

3 Duvall Robert Tom Hagen [X]

4 Keaton Diane Kay Adamsová [X]

PRO SMAZÁNÍ VHOĎTE SEM

Obrázek 5: Přetahování myši v praxi

## 5.3 Predikce filmů

Databáze obsahuje zajímavou funkci pro předpověď úspěšnosti filmů. Je založena na procházení dat v databázi. Na základě získaných informací a pomocných výpočtů je schopna určit předpokládané hodnocení filmu. Podle množství dat, které sloužilo pro odhad, spočítá jak přesný výsledek asi je. Podle provedených zkušebních testů se výsledná přesnost odhadu pohybuje kolem 70 až 85 %. Toto číslo velmi závisí na množství dat potřebných k odhadu.

Odhad se provádí podle kritérií režisér, herci, scénárista, ocenění, žánr a klíčová slova. Každá kategorie má přidělenou určitou váhu, která určuje její důležitost. Stanovení těchto hodnot je klíčové. Vyžaduje jistou zkušenost, sledování výsledků a jejich případnou korekturu. Považuji se za filmového nadšence, sleduji žebříčky hodnocení filmů, trochu znám filmovou historii a nejdůležitější prvky a povolání při tvorbě filmů. Z tohoto hlediska bylo pro mne určení jednotlivých vah relativně snadné. Mnou stanovené váhy vidíme v tabulce 1.

### Stanovení přesnosti odhadu

Stanovení přesnosti odhadu, neboli množství dat, které bylo použito pro odhad. Základním předpokladem je určit kolik informací je nutno ke stoprocentnímu odhadu. První podmínka je, aby byli

k filmu přiřazení alespoň jeden režisér, jeden scénárista a min. 3 herci. Dále pak žánr filmu a klíčová slova. Dalším požadavkem je počet filmů, na kterých se tvůrce podílel ještě před natočením filmu, pro který hledáme odhad. Opět jsem stanovil určitou hodnotu, kterou bude potřeba v budoucnosti zvýšit, protože je nastavena na malou hodnotu (aby vůbec systém produkoval nějaký odhad). To znamená, že odhad je tím pádem méně přesný. Z důvodů použitelnosti v databázi, kde je zatím málo dat, to ale pro začátek stačí. Posledním kritériem je počet hodnocení každého filmu daného tvůrce. Nastaven je zase na malou hodnotu, a proto je nutno ji v budoucnu zvyšovat. Přesnost je udávána v procentech.

č.	kategorie	váha
1	herci	0,35
2	režisér	0,3
3	scenárista	0,1
4	ocenění tvůrců	0,1
5	klíčová slova	0,08
6	žánry	0,07

*Tabulka 1: Váhy jednotlivých kritérií*

## Stanovení odhadu

Odhad pracuje tím způsobem, že vybere veškerá dostupná fakta o zjišťovaném filmu, především tvůrce. Získáme ho zprůměrováním všech hodnocení filmů každého tvůrce. Dále zprůměruji mezivýsledky v dané kategorii a následně roznásobím s váhou v jednotlivých kategoriích. Mezitím zjistíme přesnost, neboli množství dat pro každou kategorii a podle váženého průměru rozpočítám nedostatečnou kategorii rovnoměrně podle váhy do ostatních kategorií. Např. u filmu chybí režisér, proto jeho celou váhu rozložím na zbylé kategorie.

Nastínil jsem zde zjednodušený popis principu odhadu hodnocení. Snažil jsem se, aby metoda pracovala podobným způsobem jakým já sám odhaduji hodnocení filmů. Tato funkce mi ušetří čas tím, že už nebudu muset procházet jednotlivé tvůrce a odhadovat výsledek. Samozřejmě, že metoda nemůže zohlednit určité situace, kdy se nakonec film stane propadákem, např. kvůli špatnému času uvedení do kin nebo se zase stane hitem díky nové herecké hvězdě. Odhad je spíše zaměřen na oblíbenost u diváků. Např. odhad úspěšnosti filmu v kinech a jeho výsledné tržby by mohl být návrhem pro další rozšíření v této oblasti.

## 6 Závěr

Tato práce se zabývala návrhem a tvorbou filmového informačního portálu, obsahujícího funkce konkurence schopné jiným oblíbeným filmovým portálům. Můj cíl byl dosáhnout minimálně stejné funkční úrovně CSFD a přiblížit se světové filmové databázi IMDb. Podle dosažených výsledků jsi trůfám říct, že se mi to podařilo. V některých ohledech hůře v jiných lépe.

Implementoval jsem funkce, které u jiných filmových portálů nenajdeme. Jako např. predikci úspěšnosti filmů, uživatelsky přívětivé funkce založené na technice AJAXu, jako jsou kontrola formulářů, našeptávač u vyhledávání apod.

Toto téma bakalářské práce jsem si zvolil, protože se zajímám o filmy a vše co s tím souvisí. Proto práce na ní byla pro mě i zábavou. Rozšířila mi obzory v oblasti programování webových aplikací a přinesla nové zkušenosti.

Na závěr bych rád uvedl jeden malý paradox. Pracoval jsem na filmové databázi a jsem velkým fanouškem filmu. A díky této práci jsem položil oběť pro mě nejvyšší, a sice tu, že jsem během vývoje a psaní textu neviděl žádný film.

### 6.1 Další rozšíření

V rámci dalšího studia a diplomové práce, uvažuji o pokračování na tomto projektu. Je zde stále co vylepšovat. A to zejména umělá inteligence pro odhad hodnocení filmů. Přidat další predikce např. pro tvůrce. V tomto směru je cesta stále otevřená. Dále nový a více přehledný design. Vylepšení diskuzního fóra, které je zatím jen základní. Přidání lepších vyhledávacích technik, které nejsou založeny jen na porovnávání, ale které jsou schopny vyhledat i výraz, který se nějakým způsobem podobá hledanému výrazu (např. při překlepu nebo nalezení synonyma). Export a import dat pomocí XML. Pokročilejší a propracovanější redakční systém.

# Literatura

- [1] Stejskal, J. *Vytváříme WWW stránky pomocí HTML, CSS a JavaScriptu*, Computer Press, 2004
- [2] Wikipedia, Internet [online], 2007, [cit. 4.5.2007] dostupné z URL:  
<http://cs.wikipedia.org/wiki/Internet>
- [3] Wikipedia, HTTP [online], 2007, [cit. 4.5.2007] dostupné z URL:  
<http://cs.wikipedia.org/wiki/HTTP>
- [4] Wikipedia, Cascading Style Sheets [online], 2007, [cit. 5.5.2007] dostupné z URL:  
[http://cs.wikipedia.org/wiki/Cascading\\_Style\\_Sheets](http://cs.wikipedia.org/wiki/Cascading_Style_Sheets)
- [5] Staniček, P. *CSS Kaskádové styly Kompletní průvodce*, Computer Press, 2003
- [6] Staniček, P. *CSS Hotová řešení*, Computer Press, 2006
- [7] Cederholm, D. *Flexibilní webdesign*, Computer Press, 2006
- [8] Wikipedia, PHP [online], 2007, [cit. 5.5.2007] dostupné z URL:  
<http://cs.wikipedia.org/wiki/Php>
- [9] LinuxSoft, PHP (1) - Historie a budoucnost [online], 2007, [cit. 6.5.2007] dostupné z URL:  
[http://www.linuxsoft.cz/article.php?id\\_article=171](http://www.linuxsoft.cz/article.php?id_article=171)
- [10] Williams, Hugu E., Lane, D. *PHP a MySQL Vytváříme webové databázové aplikace*, O'Reilly, 2002
- [11] Gilmore, W. *Velká kniha PHP MySQL 5*, přepracované vydání u roku 2005, Zoner Press, 2006
- [12] Welling, L., Tompsonová, L. *PHP a MySQL rozvoj webových aplikací*, druhé vydání, SoftPress, 2003
- [13] Hruška, T. Studijní opora, Internetové aplikace (WAP) IV, část PHP [online], 2007, [cit. 6.5.2007] dostupné z URL:  
<https://www.fit.vutbr.cz/study/courses/WAP/private/opory/OporaWAP4ProgramovaniServeru.pdf>
- [14] Škultéty, R. *Java Script, Programujeme internetové aplikace*, 2. vydání, Computer Press, 2004
- [15] Wikipedia, MySQL [online], 2007, [cit. 7.5.2007] dostupné z URL:  
<http://cs.wikipedia.org/wiki/MySQL>
- [16] Darie, C. aj. *AJAX a PHP tvoříme interaktivní webové aplikace*, Zoner Press, 2006
- [17] Wikipedia, Informační systém [online], 2007, [cit. 8.5.2007] dostupné z URL:  
[http://cs.wikipedia.org/wiki/Informa%C4%8Dn%C3%AD\\_syst%C3%A9m](http://cs.wikipedia.org/wiki/Informa%C4%8Dn%C3%AD_syst%C3%A9m)
- [18] Korbel, J. Čeká nás HTML5 nebo XHTML 2.0? [online], 2007, [cit. 8.5.2007] dostupné z URL: <http://jankorbel.cz/2007/04/15/ceka-nas-html5-nebo-xhtml-20>

- [19] Wikipedia, JavaScript [online], 2007, [cit. 7.5.2007] dostupné z URL:  
*<http://cs.wikipedia.org/wiki/Javascript>*



# Seznam příloh

Příloha 1. ER diagram návrhu databáze

Příloha 2. Screenshot aplikace

Příloha 3. CD obsahující:

zdrojové kódy,

elektronická verze tohoto dokumentu,

návod na zprovoznění Filmové databáze,

software potřebný pro provoz a vývoj informačního systému,

manuál - součástí program jako nápověda

Úvod Novinky Uživatelé Filmy Tvůrci Statistika Diskuze  
 Styl 1 - modrá Styl 2 - oranžová  
 Přihlášen: admin  
 Nastavení  
 Pošta  
 Odhlásit

**NaFilmy.com** Filmová databáze

Názvy, jména vyhledat Vložit film Vložit tvůrce Vložit článek Vložit cenu Přítomní

**Základní info**  
 Tvůrci  
 Popis  
 Komentáře  
 Diskuze  
 Detailní hodnocení  
 Klíčová slova  
 Ocenění  
 Chyby  
 Odkazy  
 Zajímavosti  
 Galerie  
 Plakáty  
 Trailer  
 Ostatní

**Narozeniny slaví:**  
  
 Sofia Coppola (36)

**Skrytá identita**  
  
 The Departed  
 Rok: 2006  
 Délka: 149 min  
 Státy: USA,  
 Žánr: Drama, Krimi, Triller, Akční,  
 Jazyk: Angličtina,  
 Režie: Martin Scorsese,  
 Hrají:

Foto	Herec	Role
	<b>Leonardo DiCaprio</b>	Billy
	<b>Matt Damon</b>	Colin
	<b>Jack Nicholson</b>	Costello
	<b>Martin Sheen</b>	

**Popis:**  
 Příběh se odehrává v Bostonu, městě ovládaném irskou mafií v čele s jejím bosem Frankem Costelem (J. Nicholson)

**77 %**  
 54. film

admin	80 %
notice	60 %
aaa	80 %
ccc	90 %
xixao	70 %
vito	80 %
Celk. hodnocení: 6	
Počet přístupů: 50	
Denní počet: 12	
Od: 27.04.2007	
Odhadnout hodnocení filmu	
<b>Odhad filmu</b>	
<b>74 %</b>	
(přesnost: 65 %)	

Hodnotit/Komentovat  
 Diskutovat  
 Upravit film  
 Vložit plakát  
 Vložit do galerie  
 Vložit trailer  
 Vložit chybu  
 Vložit zajímavost  
 Vložit odkaz  
 Vložit klíčová slova  
 Vložit tvůrce  
 Vložit ocenění

W3C XHTML 1.1

© 2007 Luděk Hradil

