

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 PRO SPRÁVU IT PROJEKTŮ  
MALÝCH FIREM

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

AUTOR PRÁCE  
AUTHOR

MAREK ČEVELA

BRNO 2007



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INFORMATION SYSTEMS

# INFORMAČNÍ SYSTÉM PRO SPRÁVU IT PROJEKTŮ MALÝCH FIREM

PROJECT TIME TRACKER

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

AUTOR PRÁCE  
AUTHOR

MAREK ČEVELA

VEDOUCÍ PRÁCE  
SUPERVISOR

MGR. TOMÁŠ MASOPUST

BRNO 2007

## Zadání bakalářské práce

Řešitel: **Čevela Marek**  
Obor: Informační technologie  
Téma: **Informační systém pro správu IT projektů malých firem**  
Kategorie: Databáze

### Pokyny:

1. Seznamte se s principy tvorby dynamických www stránek
2. Analyzujte požadavky na tento systém. Pro analýzu využijte modelovacích technik jazyka UML.
3. Seznamte se s prostředím databázového systému MySQL a skriptovacími jazyky PHP, Perl a problematikou počítačových sítí.
4. Seznamte se důkladně s požadavky kladenými na informační systém pro správu projektů a analyzujte je. Systém bude zaznamenávat dobu strávenou nad projektem, zakládat nové projekty, přiřazovat je do různých kategorií, bude možno určit předpokládané datum dokončení projektu atd. O každém projektu bude vést detailní informace.
5. Podle výsledků analýzy navrhnete koncepci IS.
6. Navrženou koncepci realizujte.
7. Zhodnoťte dosažené výsledky a diskutujte možné pokračování v projektu.

### Literatura:

- Kanisová, H., Müller, M.: UML srozumitelně. Computer Press, 2004.
- Welling, L., Thomsonová, L.: PHP a MySQL - rozvoj webových aplikací - 2. vydání, SOFTPRESS, 2003.
- Škultéty Rastislav: JavaScript, programujeme internetové aplikace, Computer Press 2001.

Při obhajobě semestrální části projektu je požadováno:

- Body 1 až 5.


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: **Masopust Tomáš, Mgr.**, 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: **Marek Čevela**  
Id studenta: 84090  
Bytem: Ečerova 961/15, 635 00 Brno  
Narozen: 24. 09. 1984, Brno  
(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 pro správu IT projektů malých firem  
Vedoucí/školitel VŠKP: Masopust Tomáš, Mgr.  
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**

Předmětem této bakalářské práce je seznámení se s prostředky pro vývoj a provozování informačních systémů. Konkrétním cílem bylo vytvořit Informační systému pro správu projektů, který by umožnil zaznamenávání času stráveném vývojáři určitou činností při vývoji projektu. Má za úkol shromažďovat záznamy o činnosti vývojářů v týmu a získaná data poté reprodukovat a poskytovat tím informace o průběhu projektu a výsledcích vývojářů.

## **Klíčová slova**

správa projektů, záznam času, time management, XSLT, PHP, MySql

## **Abstract**

The objective of this bachelor's thesis was acquaint oneself with technologies to develop and run information systems. The practical part of this work is to develop information system Project Time tracker, which enables to monitor time that developers spend on project. The task of system is to gather entries about developer's work on project, process data and provide information about project progress and developer's outcomes.

## **Keywords**

project management, time tracking, time management, XSLT, PHP, MySql

## **Citace**

Marek Čevela: Informační systém pro správu IT projektů malých firem, bakalářská práce, Brno, FIT VUT v Brně, 2007

# Informační systém pro správu IT projektů malých firem

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Mgr. Tomáše Masopusta.

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

.....  
Marek Čevela  
1.5.2007

## Poděkování

Tímto bych chtěl poděkovat vedoucímu mé práce Mgr. Tomáši Masopustovi za ochotu a rady při konzultacích.

© Marek Čevela, 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 .....	3
Úvod .....	5
1 Použité technologie .....	6
1.1 Apache HTTP server .....	6
1.1.1 Schopnosti serveru Apache .....	6
1.1.2 Nároky informačního systému na server Apache .....	7
1.2 PHP .....	7
1.2.1 Nároky informačního systému na jazyk PHP .....	7
1.3 XSLT šablony .....	8
1.3.1 Nároky informačního systému na XSLT zpracování .....	9
1.4 Databáze MySql .....	9
1.4.1 Schopnosti databáze .....	9
1.4.2 Úložiště dat .....	11
1.4.3 Nároky informačního systému na databázi MySql .....	12
1.5 CSS – kaskádové styly .....	12
1.5.1 Problém zobrazení v různých prohlížečích .....	13
1.6 JavaScript .....	13
1.6.1 Nefunkčnost JavaScriptu v IE .....	13
1.7 Shrnutí minimálních požadavků .....	13
2 Návrh .....	14
2.1 Účel informačního systému .....	14
2.2 Požadavky systému na návrh .....	14
2.3 Datové struktury .....	15
3 Implementace .....	17
3.1 Uživatelské role v systému .....	17
3.2 Správa projektů .....	17
3.2.1 Oprávnění .....	19
3.3 Záznamy v projektu .....	19
3.3.1 Oprávnění zobrazovat záznamy ostatních .....	19
3.3.2 Oprávnění manipulovat se svými záznamy .....	20
3.3.3 Filtry .....	21
3.4 Statistiky .....	21
3.4.1 Oprávnění zobrazovat statistiky .....	21
3.5 Bezpečnost systému .....	22



3.5.1	Hesla .....	22
3.5.2	Získávání hodnot z GET a POST polí .....	23
4	Možnosti dalšího vývoje .....	24
4.1	Bezpečnost .....	24
4.2	Statistiky .....	24
4.3	Automatický systém .....	24
	Závěr .....	26
	Literatura .....	27
	Seznam příloh .....	28
	Příloha 1 .....	29

# Úvod

Při vývoji projektu je třeba vědět, jak jednotlivé části vývoje trvají dlouho, aby se mohly práce na projektu určitým způsobem plánovat a řídit. V týmové spolupráci toto platí dvojnásob a přibývá zde i potřeba monitorovat, co už je na projektu hotové, kdo to udělal a také, jak dlouho to trvalo. Tyto informace slouží zejména řídicím pracovníkům a umožňují jim lépe rozdělovat úkoly mezi podřízené a také určovat zásluhy jednotlivých pracovníků na vývoji projektu a tomu případně přizpůsobovat prémie nebo mzdy. Tato technická zpráva pojednává o informačním systému, který je určen k zaznamenávání délky času stráveného jednotlivými vývojáři při vývoji projektu, aby pak následně tyto informace zpracoval a zobrazoval manažerovi projektu i vývojářům. Systém umožňuje, aby uživatelé, kteří jej používají, získali přehled o postupu ve vývoji projektu, také o hotových částech a o výsledcích ostatních pracovníků.

V kapitole 1 jsou popsány veškeré technické prostředky, které byly potřeba pro vývoj informačního systému a také ty, které jsou potřeba pro provoz tohoto systému. Je zde i uvedeno, které specifické konfigurační nastavení jsou třeba pro provoz systému.

V kapitole 2 je popsán účet informačního systému a také to, jaké jsou požadavky na návrh systému. V závěru kapitoly jsou popsány použité datové struktury.

V kapitole 3 jsou popsány implementované funkce a schopnosti systému.

V kapitole 4 je pojednáno o dalším možném vývoji informačního systému.

První kapitola práce má název Úvod. Slouží k zasazení řešené problematiky do širšího kontextu a v podobě stručného obsahu jednotlivých kapitol definuje strukturu písemné práce.

# 1 Použité technologie

Pro vývoj a provozování Informačního systému pro správu projektů jsem využil nejrozšířenější webový server Apache HTTP server, skriptovací programovací jazyk PHP verze 5, a pro generování výsledného zdrojového kódu jsem použil šablony XSLT. Data byla uložena v databázi MySQL. Pro grafické rozvržení a formátování byly standardně použity CSS styly a u formulářů Javascript pro větší interaktivitu a lepší přehlednost pro uživatele.

Dále si v této kapitole popíšeme podrobněji použité technologie, případně nějaké specifické požadavky, které vyžaduje tento vytvořený systém.

Některé informace v této kapitole byly čerpány ze zdrojů [4] – [7].

## 1.1 Apache HTTP server

Apache je webový server, který se v současné době používá na většině serverů. Jeho úspěch je dán několika faktory – cena (je zdarma), výkon (jedná se o vícevláknový server), je multiplatformní, a obsahuje množství rozšiřujících modulů. Původně se jmenoval NCSA http, avšak vývoj tohoto systému trval jen krátkou dobu a poté byl zastaven. Sami správci webových serverů tento systém průběžně upravovali a rozvíjeli, přičemž později se tato činnost stala organizovanou. Vznikl nový název z anglického slovního spojení „A patchy server“ (záplatovaný server) a organizace Apache Group.

### 1.1.1 Schopnosti serveru Apache

Server Apache je v současnosti velmi vyvinutý systém, který má mnoho schopností a nabízí velké množství nastavení. Hlavní částí je jádro, přičemž mezi hlavní a nejvyužitelnější vlastnosti patří například možnost zakládat virtuální hosty, virtuální servery, uživatelské adresáře či bezpečnostní restriktce. Mezi konkrétní vylepšení jádra verze 2.0 patří již zmíněná schopnost pracovat jako vícevláknový server. Rovněž od této verze existuje podpora více protokolů a možnost vytvořit si vlastní modul, v němž lze implementovat komunikaci s libovolným protokolem. Rovněž je důležitá podpora IPv6 protokolu. S verzí 2.0 vznikla možnost filtrování, což znamená, že moduly mohou být vytvořeny jako filtry a mohou měnit výstup serveru po zpracování pro klienta. Je však možný i opačný postup, kdy se naopak mohou měnit vstupní data od klientů předtím, než se dostanou ke zpracování serverem. Stejně tak jsou filtry schopny i měnit výsledky CGI skriptů. V aktuální verzi 2.2 došlo zejména k několika vylepšením již existujících funkčních částí jádra, například pro autentizaci, proxy nebo cacheování. Rovněž došlo k upgradu knihovny regulárních výrazů pro Perl verze 5, dále přibýlo více možností při filtrování, nově podpora souborů větších než 2 GB a lepší podpora SQL databází.

Neméně důležitou součástí serveru jsou moduly, které lze k jádru připojit. Zavádějí se při spuštění Apache a zajišťují plnění rozmanitých požadavků, například poskytují podporu komunikace mezi serverem a databázemi, umožňují šifrovaný přenos SSL aj. Mezi zajímavé moduly patří například Rewrite, který přepisuje URL objektu, což má využití zejména v SEO oblasti. WebDAV slouží rovněž k více účelům, pracuje s právy a skupinami uživatelů a může být mj. použita pro omezení DOS útoků. Modul User tracking slouží pro monitorování pohybu uživatelů po serveru a má tedy hlavně bezpečnostní význam.

## 1.1.2 Nároky informačního systému na server Apache

Vytvořený informační systém pro správu projektů nemá na Apache server žádné speciální nároky, pro provoz postačuje standardní konfigurace serveru. Konfigurace se nastavuje v konfiguračním souboru http.conf v podadresáři conf.

## 1.2 PHP

Původní název je Personal Home Page, později se změnilo na PHP: Hypertext Preprocessor. Je to skriptovací programovací jazyk s tím, že skripty jsou prováděny na straně serveru a uživatel vidí až výsledek jejich činnosti. Používá se především pro programování dynamických internetových stránek, ale lze jej použít i k tvorbě konzolových a desktopových aplikací.

PHP se stalo velmi oblíbeným především díky jednoduchosti použití a také tomu, že vzniklo kombinací vlastností více programovacích jazyků. Syntaxe jazyka kombinuje hned několika zavedených programovacích jazyků (Perl, C, Java a Pascal). PHP je nezávislý na platformě a tudíž skripty fungují na mnoha různých operačních systémech. Obsahuje rozsáhlé knihovny funkcí pro zpracování textu, grafiky, práci se soubory a další. V PHP je implementován přístup k většině databázových serverů jako MySQL, PostgreSQL, ODBC, Oracle; také podporuje mnoho internetových protokolů například HTTP, SMTP, SNMP, FTP, IMAP, POP3 a další.

### 1.2.1 Nároky informačního systému na jazyk PHP

Je vyžadována verze PHP 5 a vyšší, přičemž mezi požadované vlastnosti patří podpora objektově orientovaného programování. To bylo při vývoji informačního systému použito a právě až PHP 5 výrazně zlepšuje přístup k objektově orientovanému programování. V systému jsou využívány i sessions, ale ty by měly být standardním doplňkem PHP jádra od verze 3. V PHP5 také velmi pokročila podpora XML s čímž souvisí následující podkapitola o DOM.

#### 1.2.1.1 Document Object Model (DOM)

Mezi důležité požadavky informačního systému na PHP jazyk patří DOM funkce, které jsou standardem definovaným organizací W3C pro přístup a práci s XML dokumenty. Ve vytvořeném

informačním systému jsou následně tyto XML dokumenty zpracovávány XSLT šablonami. Tyto DOM funkce není potřeba nijak instalovat, neboť jsou standardní součástí php jádra již od verze 5, kde byly kompletně přepracovány dle standardu. Tyto DOM funkce i jiná XML rozšíření jsou založeny na knihovně libxml2 původně vyvinuté v Gnome projektu. Minulá verze DOM obsažená v PHP4 nebyla vytvořena v souladu s DOM standardem definovaným W3C, protože nedodržovala jména metod a také obsahovala chyby v práci s pamětí. Zmíněné skutečnosti vedly k tomu, že tato verze zůstala v experimentální verzi.

### 1.2.1.2 Konfigurace v souboru php.ini

Konfigurace PHP se provádí v souboru php.ini, přičemž umístění tohoto souboru závisí na použitém operačním systému.

- **register\_globals = Off** - zapnutou direktivu register\_globals samozřejmě implementovaný informační systém nevyžaduje a z bezpečnostního hlediska je vhodné ji vypnout. Zapnutá totiž způsobuje, že z dat, která jsou přijímána od uživatele, se automaticky vytvoří globální proměnné. To může při špatné inicializaci vlastních proměnných vést k podstrčení jejich hodnot.
- **register\_argc\_argv = On** - direktivu register\_argc\_argv je třeba mít zapnutou, neboť direktiva určuje zda má PHP deklarovat proměnné argv&argc, které by ukládaly informace do GET pole. Pomocí těchto proměnných je v informačním systému řízena navigace a menu nastavuje tyto hodnoty do URL adresy.
- **extension=php\_mysql.dll** – toto rozšíření je nutné mít připojené pro zajištění komunikace PHP s MySql databází.
- **extension=php\_xsl.dll** – rovněž je nutné toto rozšíření, které obsahuje XSLT procesor (konkrétně libxml2 procesor) a umožňuje nám pracovat s XSLT šablonami. Také obsahuje podporu jazyka XSL a XPath, ve kterém jsou implementovány šablony.
- **extension=php\_domxml.dll** – toto rozšíření musí zůstat zakomentováno. Obsahuje sice podporu pro DOM XML, ale je to starší verze používaná pouze v PHP4. V této starší verzi mají metody jiné názvy a je zde použit i jiný XSLT procesor – Sablotron. V informačním systému byly použity DOM funkce, které podporuje PHP 5 a jsou standardem W3C. Pokud bude toto rozšíření zapnuto, implicitně se bude používat starší definice DOM funkcí a systém tím pádem nebude funkční.

## 1.3 XSLT šablony

Transformace XSLT (eXtensible Stylesheet Language Transformations) slouží ke generování dokumentů nebo obecně souborů v libovolném požadovaném formátu, a to na základě zdrojových dat a šablony. Tato transformace se provádí pomocí XSLT procesoru, přičemž procesorem je míněn

program podporující tuto transformaci. Procesor XSLT může být napsán v libovolném programovacím jazyku a také jsou již vytvořeny XSLT procesory pro všechny známe platformy. Většinou jsou ale již knihovny pro XSLT transformace implementovány do programovacího jazyka, jako je tomu například u PHP 5. V současnosti jsou Nyní je aktuální verze XSLT 2.0 společně s XPath 2.0, které byly po několikaletém vývoji počátkem letošního roku uznány konsorciem W3C jako standardy.

K provedení transformace je potřeba připravená XSLT šablona napsaná v jazyce XSL a zdrojová data ve formátu XML, jež společně tvoří vstup pro XSLT procesor, který na výstupu produkuje šablonou předepsaný formát. V tom spočívá síla XSLT - z jednoho zdrojového XML dokumentu můžeme generovat několik různých dokumentů, nejčastěji například HTML, XHTML, jiné XML nebo prostý text.

XSLT se využívá zejména na straně serveru, v němž aplikační logika vygeneruje XML data, která jsou před odesláním klientovi formátována na požadovaný výstup. Takovýto systém umožňuje důsledné oddělení aplikační vrstvy, která generuje XML data, od prezentační vrstvy, což je zpracování XSLT transformací. Takto postavený systém je velmi flexibilní z hlediska jeho rozšíření či dalších úprav výstupního formátu dat.

### **1.3.1 Nároky informačního systému na XSLT zpracování**

Vzhledem k tomu, že je požadována PHP verze 5, ve které je již XSLT procesor implementován, tak není třeba žádná dodatečná instalace ani konfigurace.

## **1.4 Databáze MySql**

MySql je relační databázový systém vytvořený švédskou firmou MySql AB. Jedná se o víceuživatelskou a také multiplatformní databázi, která funguje ve většině operačních systémů. Komunikace s ní probíhá pomocí jazyka SQL, jedná se o dialekt tohoto jazyka s některými rozšířeními.

Tato databáze je velmi oblíbená, a to především z důvodu, že se jedná o volně šiřitelný software, který lze získat zdarma. Je optimalizována pro výkon, takže je velmi rychlá.

### **1.4.1 Schopnosti databáze**

Databáze MySql byla od počátku svého vývoje velmi jednoduchá a až v posledních letech začaly přibývat nové vlastnosti, které programátorům chyběly. Mezi tyto vlastnosti např. patří:

- Cizí klíče – cizí klíč je sloupec tabulky, který odkazuje většinou na jiný sloupec v jiné tabulce, ale může odkazovat i do stejné tabulky. Hodnoty musí být shodné s některou hodnotou v odkazovaném sloupci, čímž se vytváří reference. Při operacích nad

databázi se provádí kontrola této shody, což se označuje jako referenční integrita. Cizí klíče jsou implementovány od verze 3.23, a to pouze v tabulkách typu InnoDB.

- Transakce – transakce je skupina příkazů, která manipuluje s daty v databázi. Od transakce jsou požadovány vlastnosti, které se dají vyjádřit zkratkou ACID. Znamená to, že transakce musí být atomická a je tedy brána jako celek, z čehož vyplývá, že se musí provést všechny příkazy v ní uvedené nebo žádný. Dále musí být konzistentní, což znamená, že neporuší konzistentnost databáze. Transakce musí být rovněž izolovaná, aby na ni nemohly mít vliv současně běžící jiné transakce. Další významnou vlastností transakce je to, že je trvalá, což znamená, že i po jejím ukončení jsou data trvale uložena. Transakce jsou podporovány od verze 3.23 pouze v tabulkách typu InnoDB.
- Poddotaz – neboli vnitřní dotaz. Je to takový dotaz, který je umístěn uvnitř jiného dotazu, tzv. vnějšího dotazu, a to nejčastěji v jeho podmínce. Poddotazy jsou implementovány od verze 4.1.
- Uložená procedura – je to část programu, která je oddělená od svého okolí a je uložena v databázi. Lze s ní zacházet stejně jako s každým jiným objektem v databázi. Má interface, tedy seznam parametrů pro přijetí komunikace od okolí a také může mít vlastní proměnné. Uložené procedury jsou většinou psány ve specifickém jazyku konkrétní databáze, který je rozšířením dotazovacího jazyka SQL. Mezi výhody uložených procedur patří skutečnost, že je lze volat z několika míst programu, nebo třeba i z oddělených programů, a přitom kód procedury bude uložen pouze jednou. Druhou výhodou je skrytí databázových operací a umožnění komunikace přes jednodušší rozhraní procedury.
- Triggery – umožňují v případě nějaké události spustit sadu příkazů. Podobají se uloženým procedurám v tom, že jsou také uloženy v databázi a mají stejnou syntaxi. Triggerům však nelze předávat vstupní parametry a nemohou vracet sadu záznamů. V triggerech se také nesmějí objevit některé příkazy, například příkaz ALTER, příkazy pro řízení transakcí, pro volání procedur, pro nastavování práv a další. Oproti uloženým procedurám, obsahují ale také něco navíc. Mají přístup k datům, která právě mění, například pokud aktualizujeme nějakou tabulku, tak trigger, který se spustí před touto aktualizací, má přístup k hodnotám těch řádků, které chceme měnit. Triggery jsou implementovány od verze 5.0.
- Pohledy – pohled je virtuální databázová struktura, která může obsahovat data z jedné či více tabulek. Jinak řečeno jsou to statické dotazy, s nimiž lze pracovat, jakoby to byly tabulky. To znamená, že z nich lze především vybírat data, ale někdy je možné i data vkládat. Databázové systémy by se bez pohledů obešly, ale práce s nimi může být mnohem pohodlnější. Pohledy mohou sloužit jako jiné pojmenování tabulek, dokáží

vybírat z tabulek jen určité řádky nebo sloupce a mohou obsahovat i výrazy jako vestavěné funkce MySQL nebo uživatelské funkce. Pohledy umožňují spojení dat z více tabulek a dokáží se také odkazovat na jiné tabulky. Jejich nespornou výhodou je rovněž skutečnost, že nezabírají v databázi žádné místo, neboť neobsahují data, ale jsou pouze předpisem pro zobrazení dat. Naopak jejich nevýhodou je fakt, že u složitějších pohledů může být získání výsledku časově náročné, protože data musí být při každém použití pohledů získána z podkladových pohledů a tabulek. Mezi další nevýhody rovněž patří to, že složitější pohledy mohou být velmi nepřehledné. Pohledy jsou implementovány od verze 5.0.

## 1.4.2 Úložiště dat

MySQL nabízí několik typů databázových úložišť (storage engine), které se liší svými možnostmi, použitím a způsobem ukládání dat do souborů. Mezi dvě nejpoužívanější patří MyISAM a InnoDB.

### 1.4.2.1 MyISAM

MyISAM je výchozím úložištěm dat v MySQL, ale nemá tolik rozšíření jako InnoDB. Každá tabulka v systému MyISAM je na disku uložena ve třech souborech, které začínají jménem tabulky. V souboru typu .frm je uložena definice tabulky, soubor s příponou .MYD obsahuje data a v souboru s příponou MYI se nachází index tabulky.

### 1.4.2.2 InnoDB

Předností InnoDB patří podpora cizích klíčů a transakcí, a to na rozdíl od MyISAM, která toto neumožňuje. Mezi InnoDB a MyISAM však existují i další rozdíly, např.:

- Pro obnovení po výpadku databáze používá InnoDB záznamy z jejich logů, zatímco MyISAM prochází a opravuje indexy všech tabulek, které mohly být teoreticky zasaženy výpadkem.
- MyISAM používá systémovou cache pro čtení nebo zapisování v tabulkách a je tedy závislá na operačním systému, zatímco InnoDB používá pro tyto operace vlastní engine,.
- InnoDB fyzicky ukládá řádky s daty v pořadí udaném primárním klíčem, zatímco MyISAM ukládá řádky tak, jak jsou vkládány. Data seřazená podle primárních klíčů může být výhodné tehdy, pokud se často hledá záznam podle primárního klíče, neboť vyhledávací doba disku je kratší a vyhledávání zabere i méně paměti. Na druhou stranu, seřazování dat v InnoDB vyžaduje také určité systémové nároky a čas.
- MyISAM podporuje na rozdíl od InnoDB kompresi dat, InnoDB tedy spotřebovává více diskového místa i paměti RAM. V budoucnu se komprese plánuje i pro InnoDB.



- InnoDB podporuje cizí klíče a transakce, což jí dává schopnost provádět i kontrolu integrity dat. Tato kontrola se provádí při každé transakci a může zpomalovat běh databáze. MyISAM toto neumí.

### 1.4.3 Nároky informačního systému na databázi MySql

Ačkoli se zdá, že úložiště InnoDB nabízí mnohem více možností, pro vytvořený informační systém bylo použito úložiště MyISAM, neboť požadavkům systému dostačuje. Mazání z více tabulek najednou je vyřešeno pomocí triggerů., což vyžaduje využití databáze MySql 5.0 nebo vyšší, neboť starší verze databáze triggerů nepodporují.

Pro správu MySql databáze je velmi oblíbený nástroj phpMyAdmin., který ovšem neumí pracovat s triggerů. Pro vytvoření triggerů se tedy dá použít například MySql Query Browser přímo od společnosti MySql AB.

## 1.5 CSS – kaskádové styly

Anglický název je Cascading Style Sheets a je to jazyk pro popis způsobu grafického zobrazení webových stránek v prohlížeči, napsaných v HTML, XHTML nebo XML. Hlavní smysl spočívá v oddělení vzhledu dokumentu od jeho struktury a obsahu. Dnes už se tyto kaskádové styly používají pravděpodobně na každé nově tvořené webové stránce, neboť jiné způsoby jsou již zastaralé a CSS nemá srovnatelného konkurenta. Nyní je aktuální CSS2 či CSS2.1, ale připravuje se verze CSS3 s dalšími funkcemi.

Mezi nesporné výhody kaskádových stylů patří rozsáhlejší možnosti úprav než samotné HTML, konzistentní styl ve více dokumentech najednou a při úpravě grafického zobrazení je mnohem jednodušší práce s kaskádovými styly, než se samotným HTML. Kaskádové styly umožňují také formátování XML dokumentů a urychlují zobrazení stránky, neboť soubor CSS se uloží do mezipaměti prohlížeče, a pokud není změněn, načítá se pouze jednou.

Naopak mezi nevýhody patří špatná podpora v majoritních prohlížečích, zejména v IE 6. V důsledku této skutečnosti dochází k různým interpretacím CSS kódu a stránka nemá ve všech prohlížečích stejný vzhled. Řešení existuje ve vytvoření více CSS souborů s tím, že podle prohlížeče, který o stránku požádá, připojíme odpovídající CSS soubor. Existují také různé triky, jak dosáhnout kýženého výsledku.

Styl stránky se může deklarovat třemi způsoby, a to přímo u formátovaného elementu, nebo pomocí stylopisu v hlavičce stránky, nebo použitím externího stylopisu v souboru .css, na který stránka odkazuje. Výhoda odděleného souboru je, že se na něj dá nalinkovat více stránek a vytvořit jejich jednotný vzhled. Při rozsáhlejších projektech se dají soubory .css vkládat do sebe a utvořit tak přehlednější strukturu.

## 1.5.1 Problém zobrazení v různých prohlížečích

Informační systém byl vytvořen pomocí metod CSS2. Kaskádové styly jsou ale pouze záležitostí prohlížeče, takže lze předpokládat, že pokud nepoužíváte nějaký starší prohlížeč, tak máte podporu CSS2. Systém byl otestován v prohlížečích Opera, Firefox, IE7 a IE6 a bylo shledáno, že je funkční pro všechny tyto prohlížeče. U prohlížeče IE6 se pouze vyskytly malé nepřesnosti v grafickém zobrazení, například špatná délka input boxů ve formulářích.

## 1.6 JavaScript

JavaScript je objektově orientovaný skriptovací jazyk, který se používá zpravidla jako interpretovaný programovací jazyk pro WWW stránky. Zdrojový kód se vkládá přímo do HTML kódu stránky a spouští se až po načtení stránky, tedy na straně klienta. Funguje pouze v prohlížeči a je na něm tedy závislý; přičemž existují různé odlišné verze jazyka i prohlížečů, což vede k častým chybám. Z tohoto důvodu jsou tedy pomocí něj ovládány obvykle jen méně důležité prvky na webové stránce, například různé interaktivní prvky jako tlačítka, menu, animace či efekty. Z bezpečnostního hlediska JavaScript nemůže přistupovat k souborům (kromě cookies) a ani k žádným systémovým objektům, také neumí uložit žádná data.

### 1.6.1 Nefunkčnost JavaScriptu v IE

Ve vytvořeném informačním systému je pomocí JavaScriptu implementováno oznámení o špatně vyplněných formulářích, dále se automaticky provádí vkládání aktuálního data a času do formulářů, což je významné pro pohodlnější a rychlejší práci s těmito formuláři. Tyto vlastnosti nejsou pro fungování systému nezbytně nutné, avšak práce se systémem by bez nich mohla být složitější, a proto je doporučeno mít Javascript zapnutý. Funkčnost je otestována v prohlížečích Firefox, Opera a Internet Explorer verze 6 a 7.

Dále je pomocí JavaScriptu implementován interaktivní kalendář ve formulářích, u něj je funkčnost ověřena pouze u prohlížečů Firefox a Opera, v Internet Explorer kalendář nefunguje. Poněvadž je JavaScript interpretovaný jazyk a neexistuje k němu žádný překladač, který by hlásil možné chyby, je objevení problému velmi náročné.

## 1.7 Shrnutí minimálních požadavků

- PHP 5 se zapnutou podporou XSLT šablon
- MySql 5 (kvůli podpoře triggerů)
- CSS 2.0

## 2 Návrh

Úvodní část této kapitoly pojednává o účelu informačního systému,, v další části se zaměříme na to, jaké požadavky by měla navržená datová struktura splňovat a v poslední části kapitoly je obsažen ER diagram s konkrétnějším výkladem.

### 2.1 Účel informačního systému

Informační systém pro správu projektů spadá do kategorie produktů, které monitorují a analyzují čas strávený při určité činnosti. Jeho hlavním úkolem je monitorovat činnost vývojářů v týmu na určitém zadaném projektu. Systém získaná data poté přehledným způsobem zobrazuje a poskytuje tím informace manažerovi projektu, například v jakém stádiu se projekt nachází, nebo kolik času strávili určití pracovníci na jednotlivých částech projektu. Vývojářům pracujícím na projektu systém také poskytuje celkovou informaci o tom, jak postupuje vývoj projektu a rovněž získají přehled o jejich dosud vykonané činnosti, časové náročnosti, případně o produktivitě v týmu.

Systém funguje z vlastní iniciativy uživatelů, kteří jej používají. Mám tím na mysli skutečnost, že si sami vývojáři měří čas strávený nad určitými činnostmi a následně vkládají do systému záznamy s popisem činnosti a stráveným časem. Tyto údaje systém zpracovává, uchovává a generuje z nich statistiky.

Příínos systému je podle mého názoru i v tom, že může významně odlehčit týmové komunikaci, a to zejména v případech, kdy jsou vývojáři závislí na výsledcích činnosti jiných pracovníků. Systém jim umožňuje kdykoliv nahlížet na průběh a výsledky jejich činnosti, což přispívá ke zkvalitnění spolupráce.

### 2.2 Požadavky systému na návrh

Mezi hlavní požadavky patří, aby každý uživatel mohl mít založený neomezený počet projektů, v nichž si vytváří kategorie. Projekt může mít neomezený počet kategorií, přičemž do každé kategorie je možno rovněž tak vložit neomezené množství záznamů. Mimo to se k projektu přiřazují vývojáři, kteří do něj mají za úkol vkládat záznamy o své činnosti. V projektu může být takto veden libovolný počet vývojářů, ale vždy alespoň minimálně jeden, a to je právě zakladatel projektu. Ten je v systému veden jako manažer projektu, ale zároveň je automaticky do projektu přiřazen i jako vývojář, aby do něj také mohl vkládat záznamy.

Krom toho si každý uživatel při vytváření kategorií v projektu buduje svůj seznam kategorií, což je pouze seznam názvů kategorií, které již použil. Při zakládání dalších projektů nemusí uživatel

již nové kategorie vypisovat ručně, ale pokud by chtěl do projektu vložit název kategorie, který již dříve použil, stačí jej vybrat z tohoto seznamu.

U každého přiřazeného vývojáře v projektu jsou také vedena jeho oprávnění, která platí vždy jen pro ten daný projekt. Manažer projektu tak má možnost stanovit různá privilegia individuálně každému vývojáři ve svém projektu.

Protože mají vývojáři možnost měnit své záznamy v projektu, je potřeba aby měl manažer přehled o těchto změnách. Tato funkce je důležitá zejména proto, že při větším počtu záznamů by se změny provedené vývojáři daly jen těžko sledovat a kontrolovat.

## 2.3 Datové struktury

Použité datové struktury jsou znázorněny v Entity Relationship Diagramu, který je zobrazen na následující stránce. Pochopení všech skutečností pouze podle diagramu by nemuselo být úplně jednoduché, a proto následuje výklad k jednotlivým entitám a vztahům.

Hlavní relací je relace „project“, která obsahuje základní informace o projektech v systému. Mezi její atributy patří id, což je jednoznačný identifikátor a primární klíč entity projekt, následuje jméno projektu, uživatelské jméno uživatele, který projekt založil a je tedy jeho manažerem, dále čas předpokládaný pro dokončení projektu, datum započetí projektu a datum předpokládaného dokončení projektu, posledním atributem jsou poznámky k projektu.

Relace „project“ je ve vztahu s relací „workers“ s kardinalitou 1:M a tento vztah je v relaci „workers“ reprezentován atributem id\_proj, což znamená, že v jednom projektu může být přiřazeno jeden nebo více vývojářů. Další atributy obsahují informace o uživatelském jméně vývojáře, který je do projektu přiřazen a informace o oprávněních tohoto vývojáře vzhledem k tomuto projektu.

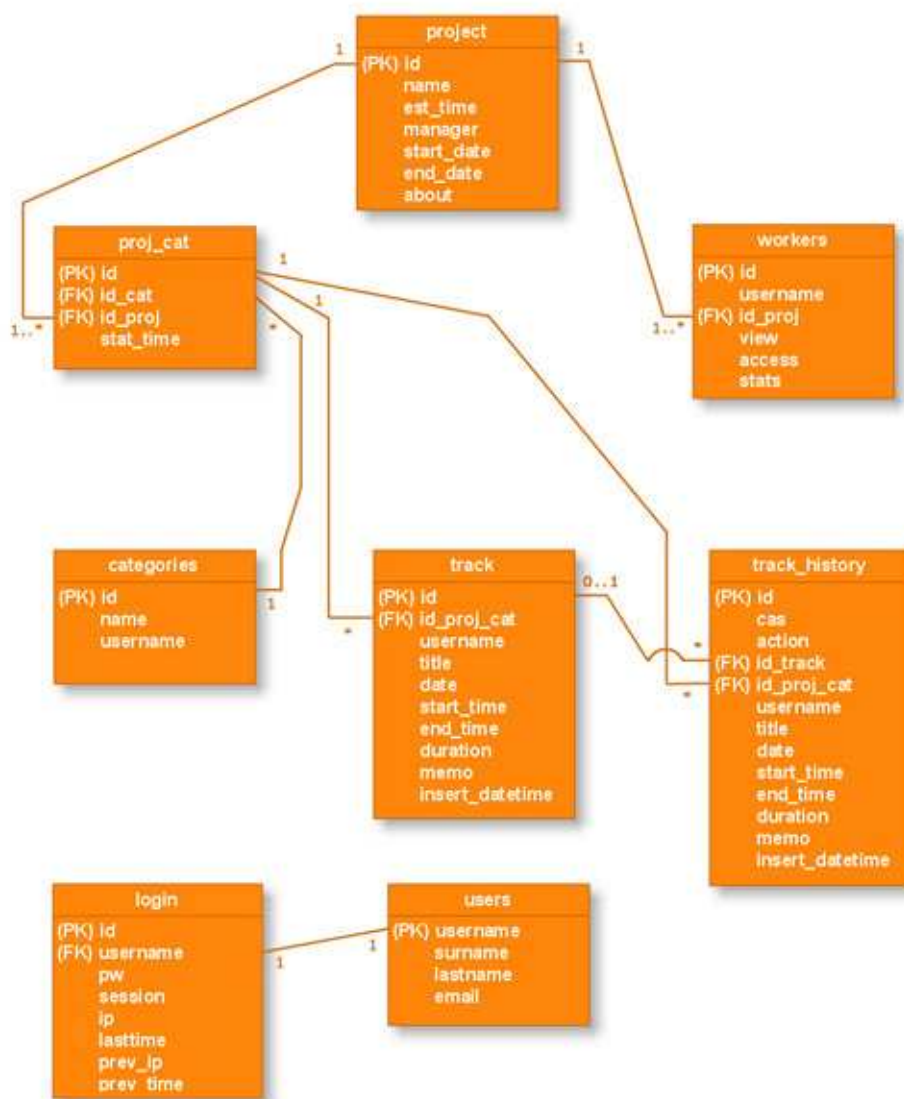
Druhou relací, se kterou je relace „project“ ve vztahu s kardinalitou 1:M, je relace „proj\_cat“. Tato relace nám fakticky tvoří spojení mezi projektem, názvy kategorií v projektu a záznamy uloženými v projektu. Obsahuje cizí klíč id\_cat, který je ve vztahu s relací „categories“ obsahující jméno kategorie a uživatelské jméno uživatele, který tento název kategorie vytvořil. Množina entit v této relaci „categories“, ve které jsou shodná uživatelská jména nám vlastně tvoří, již zmíněný, seznam kategorií jednotlivých uživatelů. Kromě klíče id\_cat obsahuje relace proj\_cat ještě i druhý cizí klíč, a to id\_proj, který reprezentuje spojení s relací „project“ s kardinalitou M:1, což znamená, že k jednomu projektu může být přiřazena jedna nebo více entit relace „proj\_cat“, respektive kategorií.

Další relací je „track“, která obsahuje informace o záznamech. Je ve vztahu M:1 s relací „proj\_cat“ prostřednictvím atributu id\_proj\_cat. Každá entita v relaci „track“ je tedy jednoznačně spojena s entitou v relaci „proj\_cat“, která je následně ve vztahu s relací „project“, z čehož vyplývá, že záznamy jsou tímto nezaměnitelně spjaty s kategoriemi i projekty. Atributy v relaci „track“ obsahují informace o autorovi vytvořeného záznamu, o názvu záznamu, o datu, o časovém intervalu

odkdy do kdy byla činnost, o které záznam referuje, vykonávána. Další atributy obsahují také délku trvání činnosti, případné poznámky k záznamu a přesné datum a čas vložení záznamu do systému.

Velmi podobnou k relaci „track“ je relace „track\_history“. V té jsou uloženy informace o změnách záznamů a v jejich atributech jsou uloženy hodnoty měněné entity z relace „track“ předtím než byla změněna. Mimo to přibyly atributy s informací o čase, kdy se editace záznamu uskutečnila, zda šlo o editaci nebo smazání a také cizí klíč id\_track, který reprezentuje vztah s entitou obsahující aktuální záznam, pokud tedy nebyl smazán.

Poslední dvě entity jsou „login“ a „users“, které jsou ve vztahu 1:1. Entita „login“ obsahuje entity s informacemi pro přihlašování a pohyb uživatele po systému. Jsou zde uložena hesla uživatelů, IP adresy, ze kterých přistupují, session řetězce, čas přihlášení, dále IP adresa minulého přístupu a čas minulého přihlášení. V entitě „users“ jsou doplňující informace o uživateli, jako křestní jméno, příjmení a emailová adresa.



Obr. 2.1, ER diagram reprezentující datové struktury systému

## 3 Implementace

V této kapitole se pojednává zejména o implementovaných funkcích informačního systému a rovněž je zde popsáno, jaké role mohou uživatelé v systému získat a jaká mají oprávnění. Okrajově jsou v této kapitole obsaženy také informace o tom, jak systém používat.

Při implementaci systému bylo čerpáno ze zdrojů [1] – [3].

### 3.1 Uživatelské role v systému

Pro práci se systémem je potřeba se nejdříve zaregistrovat do systému. Hlavním identifikačním znakem každého uživatele je jeho uživatelské jméno, které musí být tedy unikátní a nelze tudíž zaregistrovat dva uživatele se stejným uživatelským jménem.

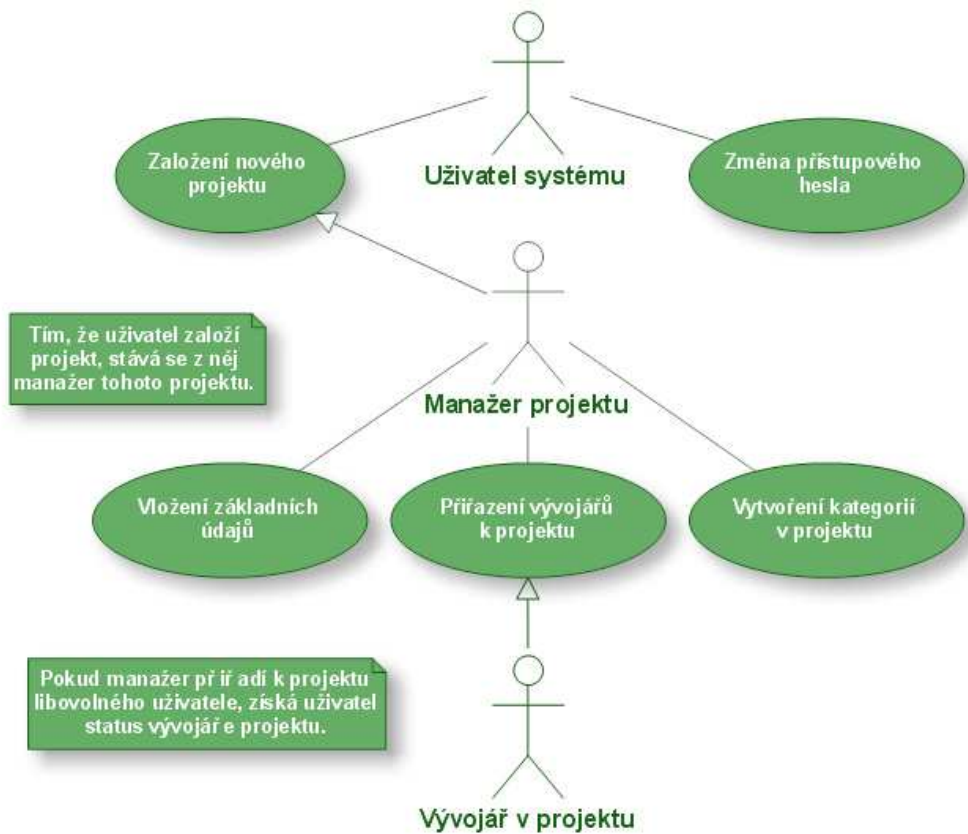
Po přihlášení nového uživatele do systému má tento uživatel prakticky pouze možnost založit nový projekt s tím, že založením projektu se stává jeho manažerem. Poté může například zadat základní údaje o projektu, vytvořit v projektu kategorie nebo přiřadit jiného uživatele do projektu jako vývojáře projektu, což je další role, kterou tento informační systém umožňuje. Jakmile se vývojář přihlásí do systému, objeví se jím vytvořený projekt v jeho seznamu projektů a může k němu přistupovat. Grafické znázornění je na obrázku 3.1.

Založením projektu se tedy uživatel stává nejenom manažerem projektu, ale fakticky je také zároveň přiřazen do projektu jako vývojář, což mu umožňuje vkládat záznamy a zobrazovat statistiky o svém profilu.

### 3.2 Správa projektů

Pod položkou Správa projektů se v informačním systému skrývají možnosti provádění různých změn v projektu. Pokud je uživatel manažerem nějakého projektu, je zde uveden seznam všech jeho projektů, které založil. Může si vybrat z několika akcí, které lze s projektem provést. Mezi tyto akce patří editace základních údajů projektu, tj. změna názvu projektu, editace předpokládaného času potřebného pro dokončení projektu, editace data započetí a dokončení projektu a zadání poznámek k projektu. V editaci základních údajů se také provádí editace kategorií v projektu. Nabízí se také možnost vložit novou kategorii do projektu, změnit název kategorie nebo kategorii smazat, avšak tím se smažou i všechny záznamy v ní obsažené.

Ve správě projektů lze rovněž provádět úpravy oprávnění vývojářů zařazených v projektu a přidávat nové vývojáře do projektu. Naopak zde neexistuje možnost vývojáře z projektu úplně vymazat, což je mimo jiné i z toho důvodu, že vývojáři, který by byl smazán z projektu, by se tento projekt přestal v jeho systému zobrazovat a mohl by to považovat i za závadu. Jestliže chce manažer



Obr. 3.1, Use Case diagram znázorňující získávání rolí v systému



Obr. 3.2, Use Case diagram znázorňující možnosti manažera projektu v záložce Správa projektů



Obr. 3.3, Use Case diagram znázorňující možnosti vývojáře v záložce Správa

vyřadit nějakého vývojáře z projektu, stačí mu odebrat oprávnění a tím mu znemožní práci se záznamy v projektu, ale zároveň v projektu zůstanou předešlé záznamy tohoto vývojáře. Pokud by chtěl manažer smazat i všechny jeho záznamy, může to udělat jednotlivě, ale předpokládá se, že tato situace nastane spíše výjimečně.

Pokud je uživatel pouze vývojář v nějakém projektu a není jeho manažerem, je pod kartou správa projektů rovněž uveden seznam všech projektů, do kterých je přiřazen. U každého tohoto projektu jsou také uvedena oprávnění, která má vývojář pro tento projekt udělena. Názorně toto ukazují diagramy zobrazené na obrázcích 3.2 a 3.3.

### **3.2.1 Oprávnění**

Jak již bylo řečeno, tato oprávnění uděluje manažer projektu vývojářům přiřazeným v projektu. Pokud manažer vloží nového vývojáře do projektu, tak ten implicitně získává veškerá oprávnění pro daný projekt. Manažer poté může tato oprávnění editovat a uděluje tři typy oprávnění, přičemž každá sada těchto oprávnění je vždy fixována pouze na konkrétní projekt a jednotlivého vývojáře.

První typ oprávnění vývojáři dává možnost zobrazovat záznamy ostatních vývojářů v projektu., druhý typ oprávnění mu umožňuje manipulovat se svými záznamy a třetí typ oprávnění znamená možnost zobrazovat statistiky projektu. Podrobnější význam těchto oprávnění je blíže popsán v následující kapitole 3.3 zabývající se záznamy a v kapitole 3.4 zabývající se statistikami.

## **3.3 Záznamy v projektu**

Záznamy projektu lze zobrazovat při kliknutí na název projektu v levém menu Moje projekty. Vkládání záznamů do projektu a jejich zobrazování je jedna z nejdůležitějších vlastností systému. Vkládat záznamy do projektu lze dvěma způsoby, přičemž jedním z nich je větší formulář, na který se dostaneme přes položku Nový záznam v horním menu. Druhý způsob je přes menší formulář, který se nachází v levém menu při zobrazování záznamů určitého projektu. V hlavičce stránky, na které je výpis záznamů projektu, se nám zobrazí název projektu, uživatelské jméno manažera projektu a také suma všech časů ve výpisu. Výpis ovládáme pomocí Filtrů, o kterých je podrobněji pojednáno v kapitole 3.3.3. Jednotlivé záznamy se řadí podle data činnosti, kterou zaznamenávají, přičemž u každého záznamu je rovněž uvedeno odkdy do kdy se na činnosti přibližně pracovalo, jak dlouho činnost trvala, kategorie a popis této činnosti a také jméno toho, kdo ji vložil.

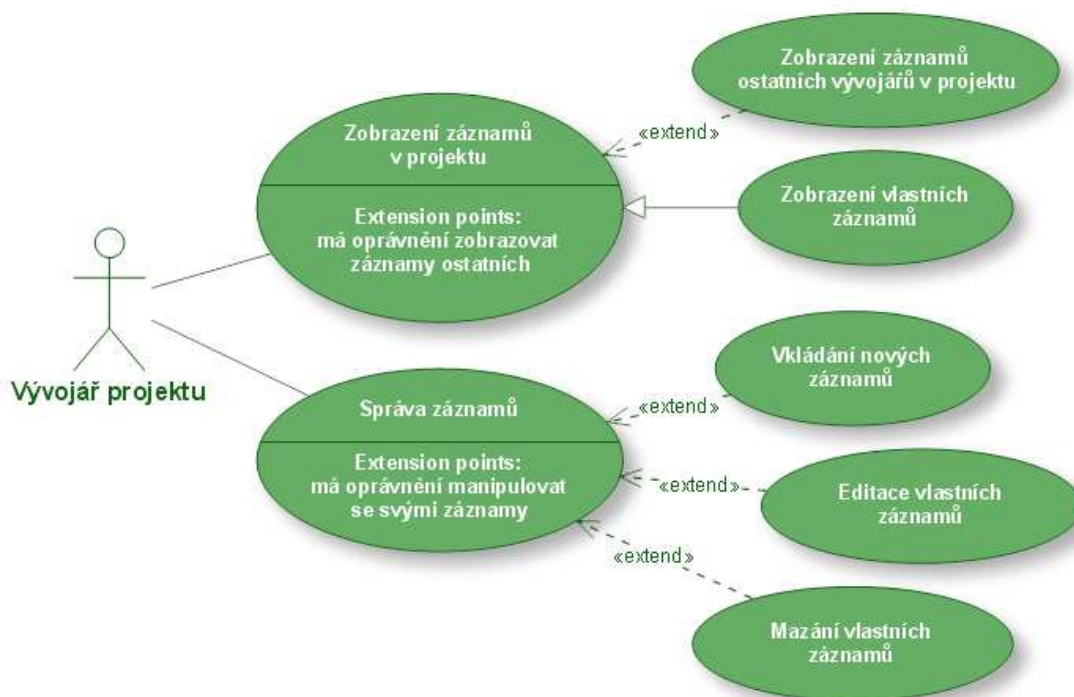
### **3.3.1 Oprávnění zobrazovat záznamy ostatních**

Toto oprávnění umožňuje vývojáři zobrazovat záznamy ostatních vývojářů v projektu. Pokud má tedy vývojář uděleno toto oprávnění, zobrazují se mu ve výpisu záznamů daného projektu záznamy i





Obr. 3.4, Use Case diagram znázorňující možnosti manažera projektu pro práci se záznamy



Obr. 3.5, Use Case diagram znázorňující možnosti vývojáře pro práci se záznamy

ostatních vývojářů, kteří pracují na projektu. Pokud toto oprávnění nemá, zobrazují se mu pouze jeho vlastní záznamy vytvořené v tomto daném projektu. Manažerovi projektu se vždy zobrazují všechny záznamy v projektu od všech vývojářů.

### 3.3.2 Oprávnění manipulovat se svými záznamy

Oprávnění umožňuje manipulovat s vlastními záznamy. Vývojář, který má v projektu uděleno toto oprávnění, může vytvářet nové záznamy a editovat nebo mazat své stávající záznamy. Pokud by mu

bylo toto oprávnění odebráno, prakticky mu to znemožní práci s projektem v systému, protože nemůže dělat žádnou z těchto akcí. Manažer projektu má samozřejmě oprávnění spravovat své záznamy, ale kromě toho může spravovat i záznamy ostatních vývojářů v projektu. Má tedy možnost zakládat nové záznamy ze svého uživatelského jména, může editovat veškeré záznamy ostatních vývojářů v projektu a je oprávněn smazat jakýkoliv záznam v projektu. Dále je mu zpřístupněn výpis všech změn na záznamech v daném projektu. Pokud tedy vývojáři poté, co vloží nové záznamy, je následně změní, může se manažer podívat na historii těchto změn.

### **3.3.3 Filtry**

Filtry slouží k filtrování seznamu záznamů, čímž nám tedy umožňují specifikovat podmínku výběru záznamů a zobrazit následně jen záznamy, které nás zajímají. První možnost je zobrazit vše, takže se zobrazují veškeré záznamy v projektu. Dále lze vybrat určitou kategorii a nechat si zobrazit všechny záznamy z této jedné kategorie. Poslední možností je výběr konkrétního vývojáře v projektu a zobrazení pouze záznamů vložených tímto vývojářem.

Menu Filtry se nachází v levém sloupci a je zobrazeno pouze tehdy, jestliže prohlížíme záznamy. Je vytvořeno jako formulář s přepínacími prvky a výběrovými seznamy obsahujícími seznam vývojářů a seznam kategorií v projektu. Formulář se automaticky odešle při výběru libovolné položky z výběrových polí nebo při kliknutí na přepínací tečku, čímž dojde k zobrazení pouze těch záznamů, které požadujeme zobrazit.

## **3.4 Statistiky**

Statistiky ovládáme pomocí malého formuláře, který má funkci jako menu. Můžeme zobrazovat dva druhy statistiky, a to statistiky projektu nebo statistiky uživatele.

Statistiky projektu vychází z odhadovaného času, který zadal manažer. Zobrazuje se zde celkový průběh projektu, což znamená procentuální poměr mezi součtem všech odpracovaných hodin na projektu a odhadovaným časem. Ve statistikách se dále zobrazuje celková odpracovaná doba každé kategorie a její procentuální podíl na celém projektu. U každé kategorie je navíc obsažen seznam vývojářů projektu, celková doba jejich práce v té dané kategorii a procentuální podíl na celém projektu.

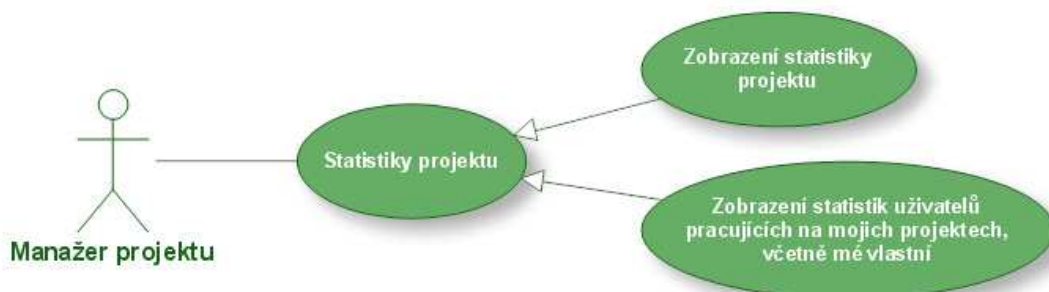
Ve statistikách uživatele je výpis všech jeho záznamů ze všech projektů, ve kterých je zapojen a celkový součet času všech těchto činností.

### **3.4.1 Oprávnění zobrazovat statistiky**

Všichni uživatelé systému si mohou zobrazit statistiku svého profilu v systému a manažeři projektu si mohou prohlížet statistiky všech svých projektů a všech vývojářů, kteří na těchto projektech pracují.

Statistika vývojářů je zde zpracována pouze ze záznamů, které jsou v projektech toho daného manažera.

Manažer může odebrat vývojáři oprávnění zobrazovat statistiky, což znamená, že vývojář si již nebude moci prohlížet statistiky tohoto projektu. Dále platí, že žádný vývojář v projektu nemá oprávnění prohlížet si statistiky ostatních vývojářů.



Obr. 3.6, Use Case diagram znázorňující možnosti manažera při zobrazování statistik



Obr. 3.7, Use Case diagram znázorňující možnosti vývojáře při zobrazování statistik

## 3.5 Bezpečnost systému

V jazyce PHP se dá mnoho věcí udělat jednoduše, ale současně také velmi nebezpečně pro chod skriptu. Proto je potřeba se tímto tématem zabývat a snažit se vytvářet bezproblémové aplikace z hlediska používání.

### 3.5.1 Hesla

Hesla jsou samozřejmě v databázi uložena šifrovaně pomocí hašovací funkce MD5. Tato funkce je jednosměrná, což znamená, že z haše zašifrovaného řetězce nelze zpětně určit původní řetězec. V informačním systému je implementován formulář pro změnu hesla, což je podstatná funkce. Prostor pro vylepšení systému zde však existuje, neboť pokud uživatel heslo zapomene, nelze jej už získat a zapomenuté heslo se musí přepsat novým. V tomto případě by bylo přínosné, aby systém měl schopnost rozesílat maily s novým heslem těmto uživatelům.

### 3.5.2 Získávání hodnot z GET a POST polí

Pomocí GET polí a hodnot obsažených v URL je v informačním systému řízena navigace. Hodnoty jsou čerpány z proměnné argv, takže se systém dá jen těžko nějakým způsobem ohrožit podstrčením podezřelých hodnot. Řetězec v proměnné argv se buď shoduje s řetězcem ve skriptu a podle toho se generuje stránka, nebo se shoda nenajde a vygeneruje se úvodní stránka. Mimo tohoto se v URL stránky posílají ještě proměnné, které specifikují id projektu nebo záznamu, se kterým se má následně provést zvolená operace. Tyto hodnoty jsou vždy čísla, takže pro bezpečnost systému je stačí přetypovat na integer. Při dalším zpracování systémem je také kontrolováno, zda přihlášený uživatel nepodstrčil do těchto proměnných id čísla projektu nebo záznamu, ke kterým by neměl právo přistupovat.

Kontroly hodnot ve formulářových polích jsou prováděny JavaScriptem na straně uživatele pro oznámení o špatně vyplněných polích. Pokud by toto uživatel ignoroval, vypnul si Javascript a vyplnil špatné hodnoty, tak databáze data nesprávného typu nebo formátu odmítne. Jako bezpečnostní prvek se dá považovat i log změn na záznamech, protože pokud by chtěl uživatel například provést pokus o MySQL inject pomocí formuláře záznamů, bude tato snaha zaznamenána v logu změn záznamů. U tohoto informačního systému se předpokládá, že bude provozován například v intranetu firmy, takže by zde nemělo existovat velké riziko napadení.

## 4 Možnosti dalšího vývoje

### 4.1 Bezpečnost

Bezpečnost je nejspíše oblast, která by se dala neustále nějakým způsobem zdokonalovat. Techniky a možnosti útočníků se také časem mění, takže je potřeba toto sledovat a reagovat na nebezpečí. Určitě by se daly v systému zdokonalit kontroly formulářových polí, například pomocí regulárních výrazů, zejména, pokud by měl být systém nasazen do prostředí internetu. Existují i důmyslnější způsoby přihlašování, například technika výzva-odpověď, kdy server pošle klientovi výzvu, klient k této výzvě připojí své heslo a serveru pošle otisk tohoto spojení. Server na své straně provede totéž a pokud se výsledky shodují, tak může proběhnout přihlášení. Využít by se dal i protokol HTTPS, pomocí něhož jsou data posílána šifrovaně. Pro uložení hesel je použita hašovací funkce MD5, která by se dala nahradit funkcí SHA-1, jež je považována za bezpečnější, avšak je také pomalejší. I u bezpečnostních vylepšení je třeba dbát na rychlost a hledat nejvhodnější řešení či kompromisy.

### 4.2 Statistiky

Ve vyvinutém informačním systému jsou statistiky tvořeny výhradně z délky časových intervalů, které byly stráveny při činnostech na projektu při jeho vývoji. Je těžké vymyslet nějaké jiné měřítko, podle kterého by se měřila výkonnost a zásluhy vývojáře na projektu. Například statistiky podle počtu záznamů by dle mého názoru neměly příliš vypovídající charakter. Lepší varianta by byly statistiky dle napsaných řádků zdrojového kódu, ale ve statistikách není zahrnuta jeho správnost a funkčnost, takže toto řešení by také nebylo příliš objektivní. Pokud by se měly měřitelným způsobem vyjádřit zásluhy vývojáře na projektu a z nich například stanovit jeho mzdu, měl by být tento úkol nejspíše přenechán manažerovi projektu. Mohlo by mu být například umožněno stanovit základní mzdu pro každou z kategorií projektu a dále možnost ohodnotit každého vývojáře formou prémie.

### 4.3 Automatický systém

Jedna z možností, jakým směrem by se mohl systém dále vyvíjet je jeho větší automatizace, která by ulehčila vývojářům od vkládání záznamů. Představme si, že by vývojáři pracovali na nějakém kódu v jazyce C, PHP, Javě nebo jakémkoli jiném, který se ukládá do souborů jako text. Každý vývojář by měl své soubory podepsané, nebo by je shromažďoval ve své složce. Soubory by monitoroval nějaký démon neboli program běžící na pozadí pracovní stanice a výsledky své činnosti by předával webové části systému, která by je, tak jako je tomu i nyní, zpracovávala a zobrazovala statistiky. Mezi schopnosti démona by mohlo patřit monitorovat časy otevírání a zavírání souborů a změny v nich,

případně počítat nově napsané řádky a dále monitorovat, zda je focus zaměřen na programy, které jsou potřebné k vývoji projektu. Z tohoto by pak mohl systém určovat dobu strávenou prací na projektu, dále rozlišit dobu strávenou v jednotlivých aplikacích, například focus zaměřený na textový editor identifikovat jako psaní kódu, dobu strávenou v prohlížeči považovat například za čas strávený studiem teorie nebo testováním v případě, že by šlo o vývoj webových technologií.

V další vyspělejší verzi by se mohla rozpoznávat i syntaxe ve vytvářených kódech, analyzovat jednotlivé vyvíjené funkce nebo třídy a tvořit statistiky týkající se konkrétní kostry tvořeného programu.

# Závěr

V průběhu práce na tomto projektu jsem samostatně zpracoval návrh a vývoj rozsáhlejšího systému, což mně umožnilo zejména prověřit si dosud získané teoretické znalosti při praktické činnosti. Konkrétní přínos vidím ve zdokonalení se v práci se XSLT šablonami a v získání více poznatků o databázích, zejména pak MySQL. V systému jsem dosud neobjevil žádné chyby či závady jeho funkčnosti, což hodnotím pozitivně. To ale neznamená, že systém je dokonalý a nenabízí prostor k dalším inovacím. Jak jsem již popsal v předchozích kapitolách, u řady funkcí lze uvažovat s mnoha vylepšeními, které by vedly zejména ke zjednodušení a zrychlení práce uživatelů se systémem.

Přínosnou pro mě určitě byla i skutečnost, že jsem získal konkrétnější představu o tom, jak je časově náročné zpracovat projekt takového rozsahu, což přispělo k mé schopnosti plánování práce.

# Literatura

- [1] Cederholm D.: Webdesign s webovými standardy, Zoner Press, 2001.
- [2] Meyer E.: Ovládněte kaskádové styly, Zoner Press, 2004.
- [3] WWW: Eric Meyer: Vertical Graphs with CSS and PHP. [online].  
URL [http://www.terrell.ca/design/vertical\\_bar\\_graphs/](http://www.terrell.ca/design/vertical_bar_graphs/)
- [4] WWW: Zajíc Petr: Seriál o MySql – Triggery. [online].  
URL [http://www.linuxsoft.cz/article.php?id\\_article=1019](http://www.linuxsoft.cz/article.php?id_article=1019)
- [5] WWW: Sládek Petr: Jak přežít Apache. [online].  
URL <http://interval.cz/serialy/jak-prezit-apache/>
- [6] WWW: Ch. Stocker: XML in PHP 5. [online].  
URL <http://devzone.zend.com/node/view/id/1713>
- [7] WWW: Zajíc Petr: Seriál o PHP. [online].  
URL <http://www.linuxsoft.cz/php/>
- [8] WWW: Kosek Jiří: Základní principy XSLT. [online].  
URL <http://www.kosek.cz/xml/xslt/uvod.html>



# Seznam příloh

Příloha 1. Ukázka grafických uživatelských rozhraní.

Příloha 2. CD obsahující zdrojové kódy.

# Příloha 1

Ukázka výčtu záznamů projektu. Horní lišta je hlavní menu, na boční straně je miniformulář pro vkládání nových záznamů, menu Moje projekty pro výběr projektu a menu Filtry.

**Σ 69:46 projekt1 Manažer projektu: admin**

Od - Do	Trvání	Činnost	Kategorie	Vložil	Akce	Poznámky
<b>13. května 2007</b>						
14:15 - 15:50	01:35	funkce pro výpis záznamů	vývoj	admin	<input type="checkbox"/> <input type="checkbox"/>	
<b>01. května 2007</b>						
08:30 - 10:30	02:00	třída záznamy	vývoj	user3	<input type="checkbox"/> <input type="checkbox"/>	
<b>19. dubna 2007</b>						
09:30 - 10:15	00:45	test formuláře nového projektu	testování	admin	<input type="checkbox"/> <input type="checkbox"/>	
10:15 - 11:45	01:30	test formuláře editace projektu	testování	admin	<input type="checkbox"/> <input type="checkbox"/>	
13:00 - 14:35	01:35	třída pro statistiky	vývoj	admin	<input type="checkbox"/> <input type="checkbox"/>	
15:00 - 16:50	01:50	implementace grafů	vývoj	user1	<input type="checkbox"/> <input type="checkbox"/>	
<b>18. dubna 2007</b>						
09:45 - 10:45	01:00	test grafiky formulářů	testování	admin	<input type="checkbox"/> <input type="checkbox"/>	
14:00 - 15:45	01:45	kontrola údajů v přihlašovacím form.	vývoj	admin	<input type="checkbox"/> <input type="checkbox"/>	
<b>12. dubna 2007</b>						
10:00 - 12:10	02:10	tabulka historie záznamů	návrh	admin	<input type="checkbox"/> <input type="checkbox"/>	dodělat oprávnění
16:15 - 17:55	01:40	tabulka projekty	vývoj-databáze	admin	<input type="checkbox"/> <input type="checkbox"/>	
<b>11. dubna 2007</b>						

Ukázka statistik projektu.

**Statistiky**

Zvolte projekt:  projekt1 nebo vývojáře vašich projektů:  admin

**projekt1**

**63%**

Celkově uděláno 63% což je 69:46 z celkově předpokládaných 110:8.

**vývoj**

**29%**

V této kategorii je odpracováno 31:37, což je 29% z celkového odhadu na dokončení projektu.

**Doba práce vývojářů v této kategorii:**

admin	25:47	23%
user1	1:50	2%
user3	2:00	2%