

UNIVERZITA PARDUBICE

Fakulta elektrotechniky a informatiky

Webový systém pro evidenci chyb v software a správu
projektů

Josef Jadrný

Bakalářská práce

2011

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Josef JADRNÝ**
Osobní číslo: **I07636**
Studijní program: **B2646 Informační technologie**
Studijní obor: **Informační technologie**
Název tématu: **Webový systém pro evidenci chyb v software a správu projektů**
Zadávací katedra: **Katedra informačních technologií**

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

V teoretické části práce bude provedena analýza požadavků na systém pro evidenci chyb v software a správu projektů. Následně budou porovnány minimálně 3 volně dostupné systémy řešící uvedenou problematiku.

V praktické části bude navržen a realizován uvedený systém, v němž bude možné zakládat a spravovat projekty a přidávat k nim problémy, aktualizovat jejich stav či je prohlásit za vyřešené. Aplikace bude obsahovat správu uživatelů a uživatelských skupin. Každá uživatelská skupina bude mít různá práva podle toho, jakou roli má v rámci daného projektu.

Příklad uživatelských skupin:

- * Návštěvník - může pouze prohlížet
- * Reportér - může přidávat nově nalezené problémy(reporty) do daného projektu
- * Programátor - může se přiřadit ke konkrétnímu problému, měnit stav reportovaných problémů či je prohlásit za vyřešené
- * Tester - může již vyřešený reportovaný problém znovu otevřít k řešení či ho prohlásit za odzkoušený
- * Správce projektu - může měnit prioritu a důležitost reportovaných problémů
- * Administrátor - správa uživatelů, přiděluje práva, spravuje projekty a DB

Rozsah grafických prací:

Rozsah pracovní zprávy:

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

Seznam odborné literatury:

***THOMSON-LECKY, Ed; NOWICKI D. Steven. PHP 6. Programujte profesionálně. Brno: Computer Press, 2009. 709 s. ISBN 978-80-251-3127-5.**

***OPPEL, A. Databáze bez předchozích znalostí. Brno: Computer Press, 2006. 320 s. ISBN: 80-251-1199-7.**

***KOFLER, M. Mistrovství v MySQL 5. Brno: Computer Press, 2007, 808 s. ISBN: 978-80-251-1502-2.**

Vedoucí bakalářské práce:

RNDr. David Žák, Ph.D.
Katedra informačních technologií

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

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



prof. Ing. Simeon Karamazov, Dr.

děkan



L.S.



Ing. Lukáš Čegan, Ph.D.

vedoucí katedry

V Pardubicích dne 31. března 2011

Prohlášení autora

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

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

Souhlasím s prezenčním zpřístupněním své práce Univerzitní knihovně.

V Pardubicích dne 6. 4. 2011

Josef Jadrný

Poděkování

Rád bych touto cestou poděkoval vedoucímu mé práce RNDr. Davidu Žákovi, Ph.D. za jeho věcné připomínky, rady a konzultace při tvorbě této práce.

Anotace

V teoretické části se zabývám problematikou požadavků na systém pro evidenci chyb v software a správu projektů a porovnáním některých systému řešící tuto problematiku. V praktické části popisují vývoj a funkčnost vlastní aplikace pro správu projektů.

Klíčová slova

WWW, systém pro správu projektů, databáze, PHP, programování v teamu, GOLEM

Title

Web-based bugtracking system

Annotation

In this theoretical part I will be considering about problem of bugtracking systems at large and about comparsion of some systems which are solveing this problem. In practical part I will be writing about programming of my own program for bugtracking and his functions.

Keywords

WWW, web-based bugtracking systém, database, PHP, team programming

Obsah

Seznam zkratk	10
Seznam obrázků	11
Seznam tabulek	12
1 Úvod	12
2 Úvod do problematiky skupinového programování	13
2.1 Programování v teamu	13
2.2 Team programming a internet	13
2.3 Problémy při programování na dálku v teamech	13
2.4 Bug tracking system.....	14
3 Analýza požadavků na systém pro správu projektů	15
3.1 Databáze chyb	15
3.2 Uživatelské role.....	15
3.3 Databáze projektů	15
3.4 Komplexní správa databáze	15
3.5 Přehlednost a přizpůsobení uživateli.....	15
3.6 Registr změn	16
3.7 Komunikační systém.....	16
3.8 Integrace dodatečných systémů	16
4 Porovnání některých volně dostupných systémů	17
4.1 Mantis Bug Tracker	17
4.1.1 Základní informace	17
4.1.2 Výhody a nevýhody	17
4.1.3 Shrnutí Mantisu.....	18
4.2 Bugzilla	19
4.2.1 Základní informace	19
4.2.2 Výhody a nevýhody	20
4.2.3 Shrnutí Bugzilly	21
4.3 Redmine	21
4.3.1 Základní informace	21
4.3.2 Výhody a nevýhody	22
4.3.3 Shrnutí.....	23

4.4 Další zástupci aplikací pro správu systémů a bug tracking	23
4.5 Kvalita volně dostupných testovaných systémů	24
4.6 Srovnání porovnávaných systémů	24
4.7 Shrnutí a závěr	26
5 Úvod k vlastní aplikaci nazvané Golem project manager	27
5.1 Motivace.....	27
5.2 Předmluva k aplikaci Golem project manager.....	27
6 Popis aplikace	29
6.1 Základní charakteristika aplikace.....	29
6.2 Požadavky na systém	30
6.3 Instalace	30
6.4 Tabulková struktura	31
6.5 Hlavní okno aplikace	34
6.6 Uživatelské skupiny	35
7 Architektura Golem project manageru.....	38
7.1 Přihlášení do aplikace	38
7.2 Registrace nového uživatele register.php.....	39
7.3 Úvodní stránka Index.php	40
7.4 Správa článků clanky.php	40
7.5 Správa uživatelů uzivatele.php	41
7.6 Správa projektů projekty.php	41
7.7 Správa reportů report.php	41
7.8 Detaily vybraného reportu reportdetail.php	42
7.9 Sledování logů logy.php	42
7.10 Nastavení účtu nastaveni.php.....	43
7.11 Nastavení Golema nastavenigolem.php	43
7.12 Další soubory	43
8 Zabezpečení aplikace	44
8.1 Systém rolí	44
8.2 Práce s hesly.....	44
8.3 Formuláře	44
8.4 Porušení referenční integrity	45
8.5 Záloha účtů.....	45

8.6	System zachytávání chyb	46
9	Závěr	47
10	Použité zdroje:	48
	Příloha A zdrojový kód se seznamem chybových hlášení.....	49

Seznam zkratek

HTML	HyperText Markup Language
XML	Extensible Markup Language
PHP	Hypertext Preprocessor
MD5	Message-Digest algorithm
CSS	Cascading Style Sheets
WYSIWYG	What you see is what you get
UML	Unified Modeling Language
W3C	World Wide Web Consortium
WWW	World Wide Web
ICQ	I Seek You
GNU	General Public License
SMTP	Simple Mail Transfer Protocol
TCL	Tool Command Language
CVS	Concurrent Version System
SVN	Sunversion
OCI	Oracle Call Interface
SHA	Secure Hash Algorithm

Seznam obrázků

Obrázek 1 – Logo Mantis	17
Obrázek 2 – Prohlížení změn v Mantisu	19
Obrázek 3 – Logo Bugzilla.....	19
Obrázek 4 – Ukázka přidání nového reportu v Bugzille	21
Obrázek 5 – Logo Redmine.....	21
Obrázek 6 - Ukázka fronty úkolů v Redmine.....	23
Obrázek 7 – Golem project manager logo.....	28
Obrázek 8 - ER diagram	31
Obrázek 9 - Hlavní okno aplikace	35
Obrázek 10 - Use case diagram	37
Obrázek 11 - Přihlašovací okno	38
Obrázek 12 - Registrace nového uživatele	39
Obrázek 13 - Regist změn	40
Obrázek 14 – Výpis logů	42
Obrázek 15 - Nastavení účtu	43
Obrázek 16 – Ukázka chybového systému.....	46

Seznam tabulek

Tabulka 1 – Shrnutí výhod a nevýhod Mantisu, zdroj vlastní.....	17
Tabulka 2 – Výhody a nevýhody Mantisu, zdroj vlastní a www.bugzilla.org	20
Tabulka 3 – Výhody a nevýhody Redmine	22
Tabulka 4 – Porovnání sledovaných systémů, zdroj vlastní.....	25
Tabulka 5 - Procentuální výsledky porovnání systémů, zdroj Tabulka 4	26
Tabulka 6 – Význam položek z tabulky práv	36
Tabulka 7 - Uživatelské role v Golem project manageru	36
Tabulka 8 - Stav reportů	41

1 Úvod

Má bakalářská práce se zabývá problematikou správy rozsáhlých projektů, se zaměřením na obor informačních technologií, kdy velmi často na takovýchto projektech spolupracuje velký počet lidí – programátorů, v různých časech na různých místech a je potřeba jejich práci koordinovat. V těchto případech narůstá potřeba nějakého centrálního systému pro koordinaci práce jednotlivých členů týmu a evidenci chyb, jelikož většina ostatních možností komunikace se v těchto případech stává minimálně neefektivním.

V první části je provedena analýza požadavků kladených na takovýto systém a porovnání tří aplikací, které uvedenou problematiku řeší. V kombinaci s provedenou analýzou uvádím nalezené nedostatky těchto systémů a výhody proti jiným aplikacím. V druhé, praktické části je nejprve uvedena motivace, která mě vedla ke zpracování tohoto tématu a následně je předveden vlastní návrh takové aplikace, která řeší některé z nalezených nedostatků. Dále je rozepsána struktura aplikace, její grafické rozhraní i použité technologie. Značnou část jsou také probírány jednotlivé uživatelské role, jelikož ty jsou pro takovýto systém klíčové. Některé komplikovanější části programu jsou rozepsány podrobněji. Část práce je věnována také bezpečnosti aplikace, například jak pracuje s hesly, jakým způsobem se pracuje s daty z formulářů, či s přístupem k databázi.

2 Úvod do problematiky skupinového programování

2.1 Programování v teamu

V dnešní době je většina softwarových produktů, ať už programů či třeba her, velmi složitá a obsahuje obrovské množství různých dat. Dříve na programování jednodušších aplikací často stačil jeden programátor. S novými technologiemi a uplynulým časem však nároky uživatele značně pokročily, nyní se již uživatel nespokojí s příkazovým řádkem a jednoduchou vektorovou grafikou, a tak se programy stávají složitějšími a propracovanějšími. Programováním takovýchto aplikací by jeden člověk strávil třeba i celý život, nemluvě o tom, jak široké vědomosti by k tomu potřeboval. Proto se dnes již většina aplikací tvoří v týmech, kde každý má svůj konkrétní úkol a čas na jeho dokončení. Takové týmy se skládají z mnoha lidí různých oborů a zkušeností, řízených většinou manažerem, který práci jednotlivých členů koordinuje a zodpovídá za výsledný projekt. Tato metoda [1] se nazývá programování ve velkém (programming in the Large) a její koncepci uvedli Frank DeRemer a Hans Kron v článku „Programming-in-the-Large Versus Programming-in-the-Small“, IEEE Trans. on Soft. Eng. 2(2) v roce 1976.

2.2 Team programming a internet

Masivní rozšíření internetu je obrovskou příležitostí pro všechny, kteří využívají programování v týmu. Velké firmy mohou ušetřit nemalé výdaje za nákup a udržování pracovních prostor. Programátoři již nemusí sedět v kancelářích v jedné budově a domlouvat se na společném postupu, každý má možnost využít některý z existujících protokolů a systémů pro programování přes internet a realizovat svoje úkoly v pohodlí domova či oblíbené restaurace, takzvaně na dálku.

2.3 Problémy při programování na dálku v teamech

Uvedl jsem zde dva nedomyšlitelné standardy vývoje aplikací moderní doby. Programování v týmech a programování na dálku, bez kterých by se aplikace dnešní doby jen těžko obešly. Efektivně používat tyto dvě koncepce společně však může přinést i některé problémy, s kterými je potřeba si poradit. Asi jako největší problém se jeví koordinace a komunikace členů týmu. Je jen těžko představitelné, že skupina stovky programátorů bude komunikovat každý s každým, například mobilními telefony, maily nebo online komunikátory typu ICQ. Samozřejmě lze tento problém částečně řešit skupinovými rozhovory, například na chatu, i toto řešení však není vhodné, programátoři mohou pocházet úplně z jiných zemí a časových pásem. Sjednocení stovky lidí z různých zemí v jeden čas na internetu je stále téměř nemožný úkol, nemluvě o tom, že jedna z hlavních výhod programování na dálku z pohodlí domova je právě jakási časová nezávislost - kdy není podstatné, kdy na svém úkolu pracuji, ale abych ho dokončil do určitého času (tzv. deadline). Shrnutím lze říci, že největším problémem se v takovýchto případech stává neefektivní komunikace a koordinace práce členů týmu.

2.4 Bug tracking system

Bug tracking system (systém pro správu projektů) je softwarová aplikace, která je navržena tak, aby napomáhala zajišťovat kvalitu při sledování hlášených softwarových chyb při práci programátorů. Pokouší se řešit výše uvedené problémy integrovaným systémem pro správu projektů, kde každému projektu lze přidávat problémy či reporty a těm přiřazovat nějaký stav dle toho, v jakém stadiu vývoje se konkrétní problém nachází. Dále takový systém nabízí pokročilou správu uživatelských rolí, kde každé roli jsou v systému přiřazena nějaká práva podle toho, o jakou roli se jedná. Každému uživateli je pak přiřazena právě jedna role podle toho, jaký úkol v rámci projektu zastává. Bug tracking system je nesmírně cenný při vývoji softwarových produktů, které jsou používány v rozsáhlých společnostech a při rozvoji softwarových produktů. Slouží také komunikaci a koordinaci práce větších programátorských skupin.

3 Analýza požadavků na systém pro správu projektů

3.1 Databáze chyb

Hlavní složkou bug tracking systému je databáze, která zaznamenává údaje o známých chybách. Obsahuje informace o tom, kdy byla hlášena chyba, její závažnost, chybné chování programu, a podrobnosti o tom, jak chybu reprodukovat, jakož i údaje o totožnosti osoby, která chybu nahlásila a případné programátory, kteří mají chybu opravit.

Chyby jsou v některých systémech označovány také jako reporty či problémy. Každá chyba prochází jakýmsi životním cyklem. V jakém stadiu životnosti se právě chyba nachází, udává její stav. Na začátku je jí přidělen stav označující její nalezení a následně je tento stav měněn a posouván dále komunitou uživatelů, kteří mají oprávnění chybu zpracovávat, až do jejího vyřešení.

3.2 Uživatelské role

Nedílnou součástí každého bug tracking systému je dostatečný počet uživatelských rolí, kde každému uživateli je přiřazena nějaká role podle toho, jakou funkci v rámci projektu zastává. Nejčastěji se role dělí na administrátory spravující databázi, správce projektů, kteří přidělují uživatelům role a koordinují práci na projektu, programátory řešící a opravující chyby, testery a reportéry, kteří projekt testují a vyhledávají chyby, které následně evidují v systému.

3.3 Databáze projektů

Databáze projektů je velice důležitá v případě, že je tvořena aplikace natolik složitá, že je nutné tuto rozdělit na více menších částí, aby se tak zvýšila přehlednost. Každý projekt pak obsahuje své chyby, které jsou od ostatních projektů a k nim patřícím chybám oddělené.

3.4 Komplexní správa databáze

Každý takový systém by měl mít integrován administrátorské rozhraní pro komplexní správu uživatelů, chyb i projektů tak, aby tato správa byla nejen jednoduchá, ale i maximálně efektivní.

3.5 Přehlednost a přizpůsobení uživateli

Hojně využívaný bug tracking systém může velice rychle nabýt mnoha dat, proto je potřeba, aby se každý uživatel snadno v takovém systému orientoval a lehce našel požadované informace. Základním pravidlem je, že aplikace by se měla přizpůsobit

uživateli, a ne uživatel aplikaci. Z tohoto důvodu určitě neuškodí, když aplikace obsahuje rozdílná rozhraní tak, aby uživatel, který ke splnění svého úkolu v rámci celého projektu nepotřebuje dobrou znalost informačních technologií, netrpěl nadbytečnými informacemi z důvodu velké komplexnosti bug tracking systému, nýbrž pouze údaji, které jsou pro dokončení jeho úkolu potřebné.

3.6 Registr změn

Velice užitečnou funkcí se jeví registr změn, ve kterém jsou uloženy veškeré změny, které byly u chyb provedeny. Takový registr by měl poskytovat možnost zobrazit ke každé chybě všechny změny na ní provedené tak, aby bylo jasné vidět kdo a jakým způsobem s chybou pracoval. Dále by měl umožňovat zobrazení posledních změn v systému tak, aby měl nově přihlášený uživatel ihned přehled o aktivitách a změnách v projektu.

3.7 Komunikační systém

Komunikační systém není nezbytnou součástí systémů pro správu projektů, ve většině případů je však spolu s bug tracking systémem paralelně veden další systém, zastávající úlohu komunikátoru na způsob fóra pro lepší koordinaci úkolů. Z toho důvodu se domnívám, že nějaký jednoduchý integrovaný redakční systém zajistí ušetří spoustu práce případným uživatelům.

3.8 Integrace dodatečných systémů

Jako cennou a nepostradatelnou funkcí, je možnost integrovat do systému pro správu projektů další odlišné aplikace, které rozšíří jeho možnosti.

4 Porovnání některých volně dostupných systémů

4.1 Mantis Bug Tracker

4.1.1 Základní informace

Mantis Bug Tracker je volně dostupný open-source web-based bug tracking systém, uvolněný pod licencí GNU. První [2] verze byla uvolněna v lednu roku 2006, jeho vývoj však započal již o 6 let dříve v roce 2000. Prvním autorem byl Kenzaburo Ito, od roku 2002 však projekt převzal Victor Boctor. Mantis je zdarma pro použití a upravování. Uživatelské rozhraní Mantisu obsahuje obarvený seznam problémů, které upozorňují uživatele na aktuální stav různých problémů.



Obrázek 1 – Logo Mantis

Požadavky na systém:

- PHP 4.3.0 nebo vyšší
- MySQL databáze 4.1.1 nebo vyšší
- Webový server (Apache a další)

4.1.2 Výhody a nevýhody

Základním znakem práce s Mantisem je mnoho možností a funkcí, které jsou vykoupeny nižší přehledností a větší složitostí. V aplikaci je dostupná i forma příkazového řádku, se kterým se dá ovládat mnoho dostupných funkcí. Výhody a nevýhody práce s Mantisem jsou shrnuty v následující tabulce (Tabulka 1), detailnější popis některých je specifičtěji rozepsán níže.

Tabulka 1 – Shrnutí výhod a nevýhod Mantisu, zdroj vlastní

výhody	nevýhody
Časté aktualizace	Nepřehlednost
Vytváření vlastních rolí	Absence komunikačních prostředků
Přidávání souborů k problémům	Nadměrná složitost
Rozsáhlý registr změn	Některé dotazy se při delším provozování systému provádí dlouho
K dispozici čeština	Po aktualizaci ze staré verze může dojít k chybám
Vytváření vlastních stavů	V poslední verzi problémy s maily

výhody	nevýhody
Propracovaný manuál	-
Možná instalace plug-inů	-
Lze lehce přeprogramovat	-

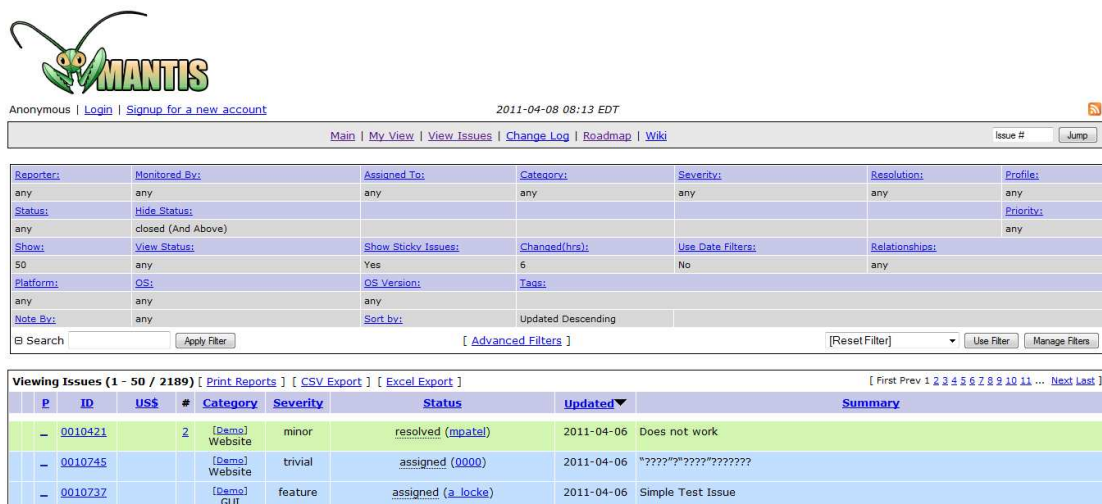
Jako první nejspíše zaujme možnost vytváření vlastních rolí či editace stávajících. V tomto ohledu je Mantis velice komplexní a umožňuje naprosté přizpůsobení potřebám uživatele. Jako jeden z mála podporuje i přidávání souborů přímo k reportům, takže není problém chybu při jejím výskytu například ofotit či zkopírovat výpis a následně jej vložit přímo k reportu. Další poměrně cennou výhodou je propracovaný registr změn, kde lze podrobně nastavit mnoho parametrů pro vyhledávání, lze tak například vyhledávat pouze změny provedené určitým uživatelem za určitou dobu v konkrétním reportu. Tato funkce velice usnadňuje práci, zvláště při delším používání programu s větším počtem reportů. K dispozici je na oficiálních stránkách i propracovaný obrázkový manuál ve více jazykových mutacích. Dále Mantis podporuje instalaci některých plug-inů, například chybějící redakční systém či integrované fórum, dle mého názoru by však takový systém měl tyto věci obsahovat už v základní instalaci. Díky přehlednému PHP kódu lze lehce najít požadované funkce a upravit je dle své potřeby.

Z nevýhod bych vytkl například nepřehlednost, která je způsobena velkou komplexností a složitostí celého systému, dále jsem se při delším používání a velkém objemu nahlášených chyb často stýkal s pomalou odezvou programu, která byla nejspíše způsobena složitými dotazy a velkým počtem atributů reportů. Po aktualizaci ze staré verze také docházelo k chybám a nesmyslným výstupům z aplikace, nestabilita byla pravděpodobně způsobena razantním přechodem na o dva roky novější verzi, s čímž autoři nejspíše nepočítali. Dále jsem řešil problémy s nechodícími maily, jelikož Mantis používá jednoduché PHP metody pro odesílání emailů, které bohužel některé hostingové servery blokují, pravděpodobně z důvodu ochrany před spamem. Problém lze vyřešit poměrně složitější instalací poštovního SMTP serveru a jeho implementace do Mantisu pomocí některého z poskytovaných plug-inů. Nutno dodat, že autoři o problému věděli a lze jej dle poskytnutých návodů vyřešit.

4.1.3 Shrnutí Mantisu

Mantis je velice rozsáhlý systém, lze jej doporučit spíše pro zkušené informatiky, pro jeho horší přehlednost, složitost a náročnost na počítačovou gramotnost uživatele, avšak jeho možnosti převyšují většinu ostatních bug tracking systémů.

Ukázku prohlížení změn reportů můžete vidět na následujícím obrázku (Obrázek 2). Barva podkreslení reportu značí jeho momentální stav. Na obrázku je viditelná velká škála možností při hledání ve změnách, kde lze nastavit konkrétní parametry hledání.



The screenshot shows the Mantis issue tracking interface. At the top, there is a navigation bar with links like 'Main', 'My View', 'View Issues', 'Change Log', 'Roadmap', and 'Wiki'. Below this is a search and filter section with various criteria like 'Reporters', 'Monitored By', 'Assigned To', 'Category', 'Severity', 'Resolution', 'Profile', 'Status', 'Hide Status', 'View Status', 'Show Sticky Issues', 'Changed (hrs)', 'Use Date Filters', 'Relationships', 'Platform', 'OS', 'OS Version', 'Tags', and 'Note By'. A search bar and 'Apply Filter' button are also present. The main part of the screenshot is a table titled 'Viewing Issues (1 - 50 / 2189)'. The table has columns for 'P', 'ID', 'ISS #', 'Category', 'Severity', 'Status', 'Updated', and 'Summary'. Three issues are visible in the table:

P	ID	ISS #	Category	Severity	Status	Updated	Summary
-	0010421	2	[Demo] Website	minor	resolved (mpatel)	2011-04-06	Does not work
-	0010745		[Demo] Website	trivial	assigned (0000)	2011-04-06	????????????????????
-	0010732		[Demo] GUI	feature	assigned (a_locke)	2011-04-06	Simple Test Issue

Obrázek 2 – Prohlížení změn v Mantisu

4.2 Bugzilla

4.2.1 Základní informace

Bugzilla [3] je webová aplikace pro sledování chyb (bug tracking) původně vyvinutá a používaná organizací Mozilla. Jde o open source software s licencí Mozilla Public License, která je používána v mnoha dalších open source projektech. Bugzilla byla původně napsána Terryem Weissmanem v roce 1998, jako open source náhrada obdobného systému používaném v Netscape Communications. Původně byla napsána v Tcl, ale ještě před zveřejněním byla přepsána do Perlu. První dostupná verze byla Bugzilla 2.0.



Obrázek 3 – Logo Bugzilla

Význam [3] pojmu chyba (bug) je zde velmi obecný, neboť se tato aplikace používá nejen pro evidenci chyb v pravém slova smyslu, ale i pro návrhy na vylepšení a požadavky nových funkcí. Podobné aplikace se z tohoto důvodu často nenazývají jako nástroje pro bug tracking, ale jako nástroje pro issue tracking.

Požadavky na systém:

- PHP 4.3.0 nebo vyšší
- MySQL databáze 4.1.1 nebo vyšší
- Perl 5
- Poštovní SMTP server
- Webový server (Apache a další)

4.2.2 Výhody a nevýhody

Základním znakem bugzilly je celkem přívětivé pracovní prostředí, kde lze nastavit mnoho parametrů zobrazení pracovní plochy, k dispozici je i mnoho šablon s možností použití grafického zobrazení. Výhody a nevýhody jsou shrnuty v následující tabulce (Tabulka 2) a rozepsány pod tabulkou.

Tabulka 2 – Výhody a nevýhody Mantisu, zdroj vlastní a www.bugzilla.org

výhody	nevýhody
Celkem přehledné	Chybí oficiální lokalizace do češtiny, pro některé verze lze použít neoficiální
Propracované fulltextové vyhledávání	Absence komunikačních prostředků
Možnost grafického zobrazení reportů	Oproti konkurenci málo funkcionalit v základní verzi
Optimalizovaná práce s databází v nové verzi	Náročnější na instalaci a zprovoznění
Mnoho plug-inů řešící chybějící funkce	Nutná instalace SMTP serveru
Vytváření uživatelských skupin a vlastních rolí	-
Možná instalace plug-inů	-

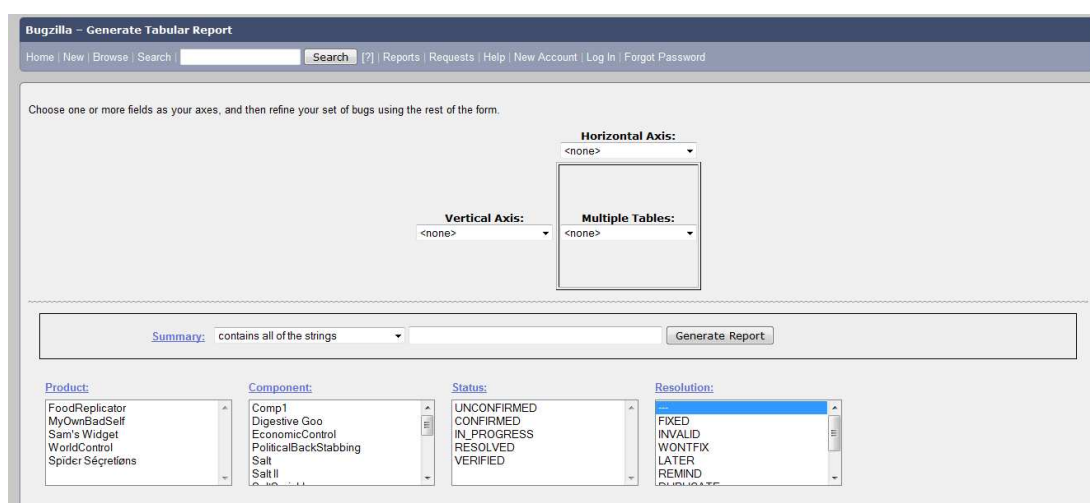
Bugzilla na první pohled zaujme přehledným grafickým zobrazením, k dispozici je mnoho šablon uživatelského rozhraní, a tak lze její podobu přizpůsobit vlastním potřebám. Bugzilla disponuje propracovaným fulltextovým vyhledávačem, kde lze detailně nastavit možnosti vyhledávání. Oficiální stránky projektu [3] tvrdí, že v poslední verzi došlo k přepracování jádra aplikace a významnému zrychlení práce s aplikací. Další podstatnou výhodou je velké množství různých přídatných aplikací, ať už upravujících vzhled nebo přidávajících nové funkce.

Z nevýhod uvedu například chybějící oficiální lokalizaci do češtiny, lze však dohledat neoficiální, použitelnou na většinu verzí Bugzilly, v některých verzích je však problém s kódováním, tedy špatným zobrazováním českých znaků. Bugzilla neobsahuje v základní verzi žádné komunikační prostředky, díky poskytovaným plug-inům však není problém dodatečného doinstalování, nicméně takovéto funkce by dle mého názoru měly

být v základním balíčku, lze tedy poukázat na málo možností základní verze. V některých případech však může být tento fakt i výhodou. Instalace Bugzilly je mírně složitější, je zde potřeba instalovat více prvků včetně poštovního SMTP serveru, podrobné návody lze však najít na oficiálních stránkách.

4.2.3 Shrnutí Bugzilly

Ukázku přidávání nového reportu lze vidět na dalším obrázku (Obrázek 4). Ze zobrazení je vidět uživatelsky přehledný průvodce, který automaticky nabízí tabulky s možnostmi.



Obrázek 4 – Ukázka přidání nového reportu v Bugzille

4.3 Redmine

4.3.1 Základní informace

Redmine [4] je volně dostupný open-source projekt manažer, jehož hlavní výhodou je použití graficky přívětivého rozhraní a časových diagramů problémů. Stejně jako většina ostatních podobných aplikací podporuje práci s více projekty. Redmine je vytvořen pomocí plnohodnotného Frameworku pro vývoj webových aplikací Ruby on Rails. První verze byla zveřejněna 25. června 2006 a je pravidelně každý rok aktualizována. Jeho instalace je však proti ostatním aplikacím obtížná, k jeho zprovoznění je potřeba následujících systémů:



Obrázek 5 – Logo Redmine

Požadavky na systém:

- Operační systém Linux
- Ruby & Ruby on Rails & Rack správné verze
- Databázový server MySQL 5.0
- Knihovny pro práci s repositáři
- SMTP poštovní server

4.3.2 Výhody a nevýhody

Kromě zmíněného grafického prostředí patří do hlavních výhod Redmine přehlednost a jednoduchost, kdy určitě nebude problém při používání systému začátečníkem. Navíc má v sobě integrováno hned několik podpůrných pomocných programů, jako rozsáhlý redakční systém, fórum, či grafickou správu uživatelských rolí s možností vytvářet vlastní nové role. Nejdůležitější výhody a nevýhody jsou shrnuty v následující tabulce.

Tabulka 3 – Výhody a nevýhody Redmine

výhody	nevýhody
Stále vychází nové verze	Lokalizovány do češtiny jsou pouze některé komponenty
Vytváření vlastních rolí	Náročné na zprovoznění
Přehledné grafické rozhraní	Obtížný na úpravy kódu
Spolupracuje s některými protokoly pro vzdálené programování a repositáři	Nepodporuje plug-iny
Možno ovládat přes email	Některé dotazy se v dlouho používané verzi vykonávají delší dobu
Rozsáhlý správce souborů	Chybový systém je oproti konkurenci málo flexibilní
Propracovaný manuál a help	-
Integrovaný redakční systém a fórum	-
Automatický registrační systém	-

Redmine je ze všech tří zde uvedených systémů nejmladší, jeho vývoj stále probíhá, výhodou proti ostatním je, že v základní verzi obsahuje podporu protokolů pro vzdálené programování, jako je SVN či CVS. Dále disponuje možností ovládání emaily, jako například přidání nového reportu pomocí emailu, což může usnadnit práci především méně zkušeným uživatelům. Také obsahuje rozsáhlý správce souborů, kde je možno pomocí podporovaných protokolů ukládat aktuální revize souborů.

Jako hlavní nedostatek bych uvedl málo možností pro úprav uživatelských rolí a chyb, kdy jsou role pevně dány a lze jim pouze upravovat některá práva. Dále Redmine nepodporuje přidávání plug-inů, většinu funkcí, které se v konkurenčních systémech přidávají plug-iny už obsahuje v základní verzi, například redakční systém. Dále je pro použitý způsob programování za pomoci Frameworku velmi obtížný na editaci kódu

4.3.3 Shrnutí

Redmine je poměrně nový a rychle vyvíjecí se systém pro správu projektů. Obsahuje mnoho nových funkcí, které ostatní aplikace nemají, na druhou stranu některé funkce postrádá, nebo nejsou tak obsáhlé jako u konkurence. Program bych doporučil spíše méně zkušeným, kteří nepotřebují extra funkce, ale spíše efektivnost a přehlednost.

Ukázka fronty úkolů ke zpracování v nejnovější verzi Redmine (Obrázek 6).

Fronta	Stav	Priorita	Předmět	Aktualizováno	Kategorie	
8142	Feature	New	Normal	auto preview on the right side of content text area	2011-04-13 15:17	Wiki
8136	Feature	New	Normal	Assign Priority based on Category	2011-04-13 15:38	Issues
8134	Feature	New	Normal	Restrict edit issue possibilities per role	2011-04-12 18:10	Issues permissions
8127	Feature	New	Normal	Public queries only for current project	2011-04-11 15:29	Issues
8121	Feature	New	Normal	Allow overriding direction of part of text	2011-04-11 09:33	Text formatting
8120	Feature	New	Normal	Code syntax highlighting in Wiki only works with Textile formatting	2011-04-10 17:11	Text formatting
8119	Feature	New	High	Easy Web GUI update	2011-04-09 23:51	Roadmap
8105	Feature	New	Normal	Add attachment as a link	2011-04-08 12:14	Attachments
8099	Feature	New	Normal	multiple project status summary page	2011-04-08 01:06	Projects
8095	Feature	New	Normal	Allow overriding the default text formatting in each project	2011-04-10 17:56	Text formatting
8094	Feature	New	Normal	Let manager set text direction of VCS commit messages per project	2011-04-10 21:06	I18n
8084	Feature	New	Normal	Quicksearch should use the "Category" field	2011-04-07 19:42	Search engine
8082	Feature	New	Normal	Issue controls	2011-04-06 17:13	Issues permissions
8080	Feature	New	Normal	Option for alphabetical only sorting of versions (date ignored)	2011-04-06 01:37	UI
8079	Feature	New	Normal	Support localized digits	2011-04-05 23:16	I18n
8078	Feature	New	Normal	News with attachments	2011-04-06 09:21	News
8073	Feature	New	Normal	Automated cleaning of identical issues	2011-04-05 17:11	Issues
8063	Feature	New	Normal	optionally suppress redundant display information on issue listings	2011-04-04 16:37	Issues
8058	Feature	New	Normal	Separation of content language from UI language	2011-04-04 15:16	I18n
8050	Feature	New	Normal	Nightful workflow field enhancement: visible, read only and mandatory	2011-04-03 02:15	Issues workflow

Obrázek 6 - Ukázka fronty úkolů v Redmine

4.4 Další zástupci aplikací pro správu systémů a bug tracking

Kromě zde zmiňovaných Mantisu, Bugzilly a Redmine existuje celá řada dalších, ať už volně dostupných či placených systémů. Patří mezi ně například nejstarší GNATS, jehož první verze vyšla roku 1992 v jazyce C či Flyspray. Má práce se zaměřuje pouze na tzv. bug tracking systémy typu klient-server, tedy systémy pro vyhledávání a sledování chyb, existuje však ještě další, blízce navazující odvětví nazývané originálním anglickým názvem Issue tracking systems. Liší se především v tom, že Issue tracking systém slouží předně k řešení problémů obyčejných uživatelů či zákazníků, kde na jedné straně stojí právě oni uživatelé, kteří mají nějaký, ať už individuální nebo globální problém a na straně druhé skupina uživatelů se zkušenostmi a schopnostmi problémy řešit. Takovéto systémy mají hlavní využití například jako helpdesky firem a společností, kde se zaměstnanci snaží pomoci vyřešit nějaké problémy svým zákazníkům s jejich produkty. Další možností

využití jsou například telefonická centra, kde se na svém počítači operátor v podobném programu proklikává a snaží se telefonicky pomoci volajícímu. Některé ze zde uvedených systémů lze využít i jako issue tracking systémy, ale nejsou pro to primárně určeny.

Dalším odvětvím, kterým se zde nezabývám, ale je řešenému tématu velice blízké, jsou takzvané source code repository nebo také jiným názvem hosted bug tracking systémy, které jsou obdobou zde uváděných systémů, s tím rozdílem že jsou zprovozněny na hostovaných serverech a kdokoliv se může přes svůj internetový prohlížeč na tyto servery připojit a pracovat, některé umožňují i vytváření vlastních uživatelských projektů a skupin a můžou velmi ulehčit práci. Mezi nejpoužívanější zástupce patří velmi známý Sourceforge a Google Code.

4.5 Kvalita volně dostupných testovaných systémů

Existuje jen velmi málo placených systémů pro bugtracking, jejich kvalita je srovnatelná s volně dostupnými, nemá tedy velký význam nějaký z takových systémů zakupovat a mohu vše doporučit použití nějakého z volně dostupných systémů, kterých je celá řada a jistě nebude pro nikoho problémem najít některý, který bude plně vyhovovat. Výjimku tvoří některé systémy, které jsou tvořeny většinou na zakázku velkých firem a obsahují další aplikace tak, aby nejlépe vyhovovaly potřebám firmy. V takovém případě je potřeba zvážit, zda li nebude systém šitý na míru pro potřeby Vaší firmy vhodnější.

4.6 Srovnání porovnávaných systémů

Jako poslední část této kapitoly uvádím tabulku (Tabulka 4) pro rychlý přehled zde porovnávaných systémů, vzhledem k analýze na požadavky pro takovýto systém z druhé kapitoly. K vyšší přehlednosti jsem každému testovanému odvětví přidělil maximální možný počet bodů, které představují míru důležitosti v rámci požadavků na Bug Tracking systém. Detaily k jednotlivým položkám jsou již uvedeny v kapitolách konkrétních systémů, nebudu je zde tedy znovu rozepisovat. Pro upřesnění uvedu způsob, jakým jsem testované systémy hodnotil.

Způsob hodnocení:

- Databáze chyb:
 - Možnosti práce s reporty, stavy, hledání a procházení reportů, přidávání reportů a jejich organizace v aplikaci i v databázi

- Uživatelské role:
 - Množství rolí, editace rolí, přidávání vlastních, možnosti práv a jejich změn, vytváření uživatelských skupin
- Databáze projektů:
 - Možnosti práce s projekty, hledání v projektech, organizace projektů a jejich reportů v aplikaci i v databázi
- Správa aplikace:
 - Administrátorské rozhraní, jeho vzhled a možnosti, správa uživatelů a nastavení aplikace
- Přehlednost:
 - Vzhled aplikace, možnosti vzhledu a jeho přizpůsobení dle potřeby
- Složitost
 - Orientace začátečníka i znalce v systému, celkové možnosti systému vzhledem k přehlednosti
- Registr změn:
 - Kvalita registru, jeho přehlednost, možnosti vyhledávání, způsob ukládání změn a jeho úprava
- Komunikační systém:
 - Kvalita a možnosti komunikačních prostředků v aplikaci, chat, fórum, redakční systém
- Plug-iny:
 - Možnosti dodatečného rozšíření aplikace, správa a vyhledávání plug-inů, obtížnost jejich instalace a zprovoznění
- Lokalizace:
 - Kvalita lokalizace do češtiny, způsob zobrazování českých znaků, obtížnost lokalizace

Tabulka 4 – Porovnání sledovaných systémů, zdroj vlastní

-	Max bodů	Mantis	Bugzilla	Redmine
Databáze chyb	10	10	8	5
Uživatelské role	8	8	8	6
Databáze projektů	6	6	5	3
Správa aplikace	8	4	5	4
Přehlednost	6	2	5	6
Složitost	5	2	4	5
Registr změn	8	7	5	4
Komunikační systém	5	0	0	3
Plug-iny	3	1	3	0
Lokalizace	3	3	2	1
celkem	62	43	40	37

4.7 Shrnutí a závěr

V hodnocení nejlépe vyšel Mantis, především zásluhou jeho širokých možností, mnoho bodů však ztratil na přehlednosti a složitosti, které souvisí s jeho velkou komplexností. Bugzilla trápí tři body, jedná se o jakýsi průměr v možnostech aplikace a jejího vzhledu, naopak Redmine trápí mnoho bodů na základních systémech pro správu chyb a změn, ale naopak vyčnívá v přehlednosti a malé složitosti. Pro upřesnění zde uvádím tabulku (Tabulka 5) procentuálního splnění požadavků na dokonalý bug tracking systém založené na výsledcích z tabulky porovnání systémů (Tabulka 4). Pro výpočet byl použit vzorec:

$$\text{procentuální úspěšnost} = \frac{\text{počet získaných bodů}}{\text{maximální počet bodů}} \times 100$$

Tabulka 5 - Procentuální výsledky porovnání systémů, zdroj Tabulka 4

	Mantis	Bugzilla	Redmine
Procentuální splnění požadavků na dokonalý Bug tracking systém*	69,4%	64,5%	59,6%

* získaný počet bodů/ maximální počet bodů

Z mého pohledu nejlepší volbou je Mantis či Bugzilla s podobným bodovým průměrem, pro zkušeného informatika bych volil spíše Mantis, který je složitější, ale skýtá více možností z pohledu bug tracking systémů. Pro méně zkušené uživatele lze spíše doporučit Bugzillu, která svůj účel také dosáhne a obsahuje přívětivější grafické rozhraní. Redmine lze doporučit spíše pro začátečníky v problematice bug tracking systémů a to především díky přehlednosti, což začátečník zajistě ocení.

5 Úvod k vlastní aplikaci nazvané Golem project manager

5.1 Motivace

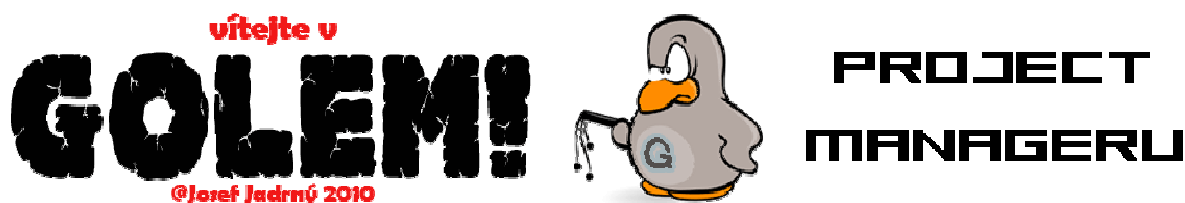
Hlavním důvodem, proč jsem se rozhodl zpracovat vlastní systém pro správu projektů, jsou mé dosavadní zkušenosti s používáním některých bug tracking systémů. Několik měsíců jsem se účastnil jako programátor vývoje jednoho nejmenovaného herního serveru. Vývoj probíhal především v jazyce Java a mimo jiných, byl využíván i Mantis, jako program pro správu a evidenci chyb. Pro programátory to byla neocenitelná pomůcka, často jsem na něm trávil celý den a ani se při práci neodhlašoval. Stále jsem aktualizoval postup mé práce a sledoval, na čem zrovna pracují ostatní členové teamu a jak se jim daří. Jelikož ale většina programátorů o samotné hře moc nevěděla, byla zde další skupina uživatelů, hráčů, kteří hru výborně znali a jejich úkolem bylo vyhledávat chyby, testovat je a následně veškeré poznatky evidovat v Mantisu. Většina z nich neměla žádné zkušenosti s podobnými systémy a ani valné zkušenosti s používáním internetu a to bylo jádrem celého problému. Většina z nich neměla ponětí, jak chyby do systému vložit a jak je spravovat. Pokud někdo našel chybu, kontaktoval raději některého z programátorů s detaily a požádal ho, aby chybu do systému vložil sám. Po delší době jsme si začali uvědomovat, že vlastně většina testerů a reportérů své účty v Mantisu ani nepoužívá, či je vůbec nemá a systém se tím pádem stával málo efektivním, jelikož přidával práci programátorům, kteří strávili mnoho času komunikací s testery mimo systém a následným testováním nalezených chyb a jejich vkládáním do systému. Částečným řešením bylo doprogramování jednoduchého redakčního systému do jádra Mantisu, který na hlavní stránce po přihlášení zobrazoval mnoho námi vytvořených a vložených návodů jak s ním pracovat. I tak si ale mnoho hráčů uchovalo svůj prvotní odpor k tomuto systému a nikdy jej nezačali používat a my stále ztráceli čas. Právě tyto zkušenosti mě vedly k tomu, vypracovat jako svou bakalářskou práci systém pro správu projektů, který by nabízel všechny základní funkce kladené na takovýto systém, včetně redakčního systému a poskytoval proměnné uživatelské prostředí, dle toho jaký typ uživatele s aplikací pracuje tak, aby ani v informatice neznalý uživatel neměl problém jej používat a splnit tak svůj úkol v rámci projektu.

5.2 Předmluva k aplikaci Golem project manager

Důvody, které mě vedly k vytvoření takové aplikace, jsem rozepsal v kapitole 5.1, nyní bych řekl pár slov úvodem k jejímu popisu. Golem project manager je jednoduchý program pro správu a evidenci chyb. Nabízí základní možnosti správy chyb, jejich přidávání, editaci a odstraňování, včetně systému priorit, přiřazování a stavů. Dále nabízí podporu třídění chyb do projektů, evidenci změn chyb, evidenci logů, jednoduchou správu uživatelů a sedm uživatelských rolí, až na pár výjimek s pevně stanovenými přístupovými právy. Dále obsahuje jednoduchý redakční systém s publikací článků, s možností komentářů a jejich správu.

Z důvodu omezeného času a prostředků určitě nedosahuje rozsahu zde testovaných systémů, které jsou ve vývoji již několik let a pracuje na nich mnoho lidí, nabízí však všechny klíčové funkce, které by měl takový systém umět, i když často na základní, omezené úrovni. Aplikace byla programována tak, aby se dala případně dále rozšiřovat a přidávat nové funkce. Veškeré prvky pro práci s databází jsou umístěny na jednom místě, kvůli lepší přehlednosti a případným dodatečným úpravám.

Aplikaci jsem nazval Golem project manager, úvodní logo aplikace můžete vidět na dalším obrázku (Obrázek 7). Originální obrázek¹ tučňáka pochází z veřejné obrázkové knihovny Open Clip Art Library a jeho autor se jeho zveřejněním vědomě vzdal všech autorských práv, které mu ze zákona náležely.



Obrázek 7 – Golem project manager logo

¹ Originální obrázek <http://www.clipartist.net/clipart/open-clip-art/manager-mimoooh-01-manager-clip-art/>

6 Popis aplikace

6.1 Základní charakteristika aplikace

Golem project manager je jednoduchý, ale komplexní systém pro správu softwarových projektů a evidenci chyb. Jeho služeb se dá využít i v jiných případech, kdy je potřeba sdílet více informací s více lidmi, ohledně řešení nějakého složitého problému. Aplikace umožňuje správu uživatelů a uživatelských skupin, dále je možno vytvářet, či upravovat projekty a k jednotlivým projektům vytvářet reporty (zprávy) s detailními podrobnostmi, jako je například prioritita, či obtížnost problému. Zároveň má každý report nějaký stav, v jakém stádiu řešení se právě nachází. Aplikace dále obsahuje jednoduchý redakční systém pro publikaci článků a komentářů, pro zjednodušení komunikace mezi uživateli a administrátorské rozhraní pro správu aplikace a uživatelů. Systém také eviduje všechny změny provedené na projektu, které je možné v aplikaci zobrazit tak, aby byl u každého reportu zřetelný průběh jeho řešení.

Základní charakteristika v bodech:

- Správa uživatelských skupin, uživatelů, projektů a reportů
- Jednoduchý redakční systém s podporou komentářů
- Předdefinované stavy a priority, které je možno reportům přiřazovat
- Evidence změn reportů a možnost jejich zobrazení v aplikaci
- Použité technologie: CSS, HTML, PHP, Oracle DB, javascript
- Plně validní HTML a CSS kód dle standartu W3C
- Stejně zobrazení ve většině dostupných prohlížečů, včetně podpory pro tisk
- Zabezpečení hesel kódováním
- Evidence aktivit uživatelů
- Možnost exportu projektů či logů do souboru
- Administrátorské rozhraní pro nastavení aplikace a účtů
- Vzhled aplikace je závislý na typu přihlášeného uživatele
- Veškeré funkce pro přístup a práci s databází na jednom místě
- Funkce pro práci s databází vracejí číselný kód, který představuje úspěšnost prováděného dotazu, který je následně zachycen a zpracován aplikací

6.2 Požadavky na systém

Golem project manager ke zprovoznění vyžaduje následující podpůrné systémy:

- Databázový server Oracle 10
- Softwarový webový server včetně PHP, například Apache

6.3 Instalace

Na databázovém serveru se musí nacházet tabulky se strukturou zobrazenou na následujícím obrázku (Obrázek 8) s E-R diagramem aplikace a některá další data. Potřebný sql kód, generující zmiňovanou tabulkovou strukturu lze nalézt v souboru readme.sql, v kořenovém adresáři s aplikací. Použitím celého kódu ze souboru lze docílit sestavení kompletní tabulkové struktury na serveru, do tabulek však budou přidána i některá dodatečná data jako administrátorský účet a úvodní článek.

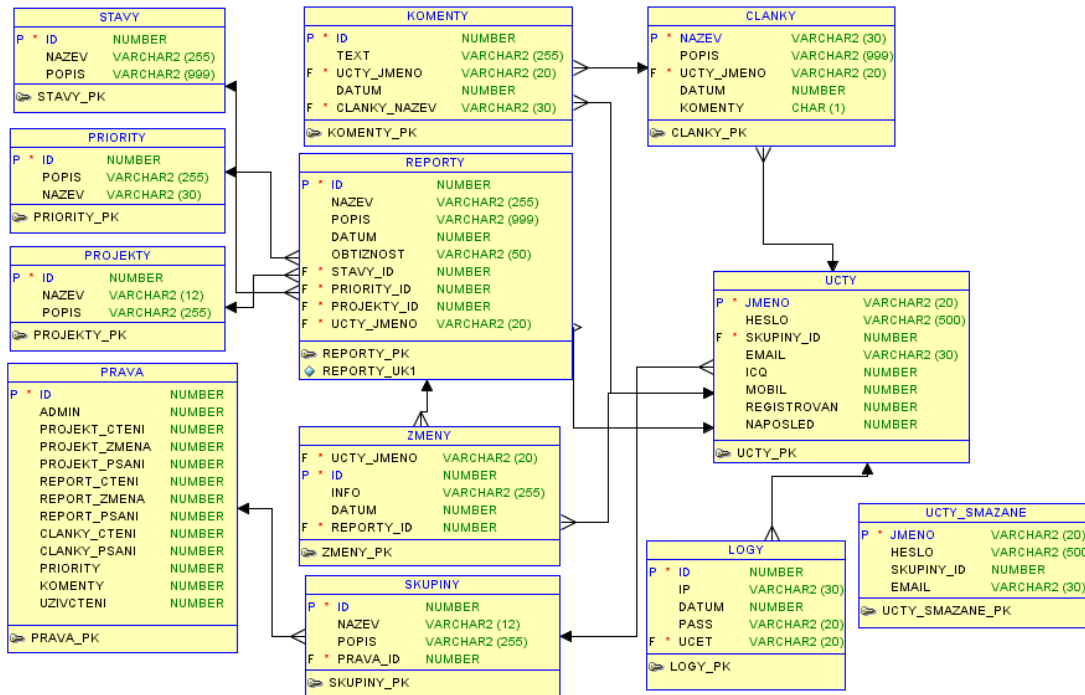
Dále je nutností upravit funkci dbConnect v konfiguračním souboru config.php, která obsahuje údaje pro přístup k databázi dle následujících pokynů. Červeně označené položky je nutno upravit.

```
function dbConnect() {
    PutEnv('ORACLE_SID=VaseSID');
    PutEnv('ORACLE_HOME=VasDomovskyAdresarOracle');
    PutEnv('LD_LIBRARY_PATH=$ORACLE_HOME/CestaKeKnihovnam');
    $con = oci_connect("UzivatelскеJmeno", "Heslo", "Domena",
'AL32UTF8') or die(print_r(oci_error()));
    if (!$con) {
        $e = oci_error();
        trigger_error(htmlentities($e['message'], ENT_QUOTES),
E_USER_ERROR);
    } else {
        return $con;
    }
}
```

Pro první přihlášení bude nutno vytvořit administrátorský účet admin, lze tak učinit třemi způsoby.

1. Vytvořte účet nazvaný admin přímo v databázi a položku heslo nechte prázdnou. Po prvním přihlášení lze heslo změnit.
2. Vytvořte nový účet admin přes registraci v aplikaci a poté mu nastavte v databázi položku SKUPINY_ID na 1. Tím mu přidělíte maximální možná přístupová práva.

3. Použijte ke generování struktury tabulek kompletní sql kód ze souboru readme.sql (výchozí nastavení hesla a jména je admin, admin).



Obrázek 8 - ER diagram

6.4 Tabulková struktura

Na obrázku 8 je zobrazen ER diagram aplikace, který ukazuje tabulkovou strukturu Golem project manageru. Významy tabulek podrobně rozebírám níže.

UCTY

Základní tabulka s uživatelskými účty. Slouží k uchování informací o uživateli, včetně kontaktních údajů. Jako primární klíč je zde zvolen atribut JMENO, protože název účtu musí být jedinečný.

- *Primární klíč: JMENO (Varchar2)*
- *Cizí klíč: SKUPINY_ID (Number)*
- *Další atributy: HESLO (Varchar2), EMAIL (Varchar2), ICQ (Number), MOBIL (Number), REGISTROVAN (Number), NAPOLSED (Number)*

CLANKY

Tabulka obsahující informace o článcích, včetně těla článků. Jako primární klíč je zvolen název článku.

- *Primární klíč: NAZEV (Varchar2)*
- *Cizí klíč: UCTY_JMENO (Varchar2)*
- *Další atributy: POPIS (Varchar2), DATUM (Number), KOMENTY (char)*

KOMENTY

Tabulka komentů obsahuje komentáře k článkům. Jako hlavní klíč je zde atribut ID, který je automaticky generován pomocí sekvence.

- *Primární klíč: ID (Number)*
- *Cizí klíč: UCTY_JMENO(Varchar2), CLANKY_NAZEV (Varchar2)*
- *Další atributy: TEXT (Varchar2), DATUM (Number)*

UCTY_SMAZANE

Tato struktura slouží k uchování dat smazaných účtů, které slouží jako záloha kontaktů na bývalé uživatele.

- *Primární klíč: JMENO (Varchar2r)*
- *Další atributy: HESLO (Varchar2), SKUPINY (Number), EMAIL (Varchar2)*

LOGY

Tabulka logy slouží k uchování informací o přístupu k aplikaci včetně informací o přihlášených uživateli, jako jsou například IP adresy.

- *Primární klíč: ID (Number)*
- *Cizí klíč: UCET (Varchar2)*
- *Další atributy: IP (Varchar2), DATUM (Number), PASS (Varchar2)*

SKUPINY

Vazební tabulka skupiny spojuje uživatelská práva a uživatelské účty v rámci jedné uživatelské skupiny s primárním klíčem ID skupiny.

- *Primární klíč: ID (Number)*
- *Cizí klíč: PRAVA (Number)*
- *Další atributy: NAZEV (Varchar2), POPIS (Varchar2)*

PRAVA

Tabulka přístupových práv, v kombinaci s tabulkou skupiny tvoří uživatelské role přidělované uživatelům.

- *Primární klíč: ID (Number)*
- *Další atributy: ADMIN (Number), PROJEKT_CTENI (Number), PROJEKT_ZMENA (Number), PROJEKT_PSANI (Number), REPORT_CTENI (Number), REPORT_ZMENA (Number), REPORT_PSANI (Number), CLANKY_CTENI (Number), CLANKY_PSANI, PRIORITY (Number), KOMENTY (Number), UZIVCTENI (Number)*

REPORTY

Jedna z klíčových tabulek tvořící jádro aplikace. Ukládá veškeré informace o reportovaných chybách.

- *Primární klíč: ID (Number)*
- *Cizí klíč: STAVY_ID (Number), PRIORITY_ID (Number), PROJEKTY_ID (Number), UCTY_JMENO (Varchar2)*
- *Další atributy: NAZEV (Varchar2), POPIS (Varchar2), DATUM (Number), OBTIZNOST (Varchar2)*

PROJEKTY

Tabulka třídící reporty do projektů.

- *Primární klíč: ID (Number)*
- *Další atributy: NAZEV (Varchar2), POPIS (Varchar2)*

PRIORITY

Tabulka se seznamem možných priorit k reportům.

- *Primární klíč: ID (Number)*
- *Další atributy: NAZEV (Varchar2), POPIS (Varchar2)*

STAVY

Tabulka se seznamem možných stavů k reportům.

- *Primární klíč: ID (Number)*
- *Další atributy: NAZEV (Varchar2), POPIS (Varchar2)*

6.5 Hlavní okno aplikace

Vzhled hlavního okna aplikace je závislý na typu přihlášeného uživatele a jeho práv. Plné zobrazení se všemi možnostmi, přístupné administrátorům a správcům systému lze vidět na dalším obrázku (Obrázek 9). Z obrázku je patrné hlavní rozdělení aplikace. V horní části je zasazeno logo Golem project manageru, pod ním je úzký panel s aktuálním datem a časem a tlačítka pro základní nastavení účtu a aplikace. Na levé straně lze pozorovat hlavní uživatelské menu pro práci se systémem. Na pravé straně je panel s informacemi o uživateli a možností odhlášení. Prostřední okno je rozděleno na dvě části. Vrchní část slouží k zobrazení posledního vloženého článku, spodní část je hlavní pracovní plocha, zde se zobrazují vstupy a výstupy programu, dle toho s jakou částí aplikace uživatel pracuje. Zápatí aplikace je věnováno důležitým odkazům a při práci s aplikací se nemění.

Obrázek 9 - Hlavní okno aplikace

Aplikace byla od začátku vyvíjena v české lokalizaci, a to zcela záměrně, její přeložení do dalšího jazyka by vzhledem k malému počtu textů nebylo složité.

Rozdílné pracovní prostředí dle typu přihlášeného uživatele je dále zmiňováno v další kapitole. Aplikace dále obsahuje další grafické prvky, které se zobrazují pouze jako odpověď na konkrétní akce uživatele, jako například chybové okno a podobně.

6.6 Uživatelské skupiny

Golem project manager obsahuje sedm uživatelských rolí. Každý uživatel musí být členem právě jedné skupiny. Aplikace zatím neumožňuje změnu jednotlivých práv přímo k jednotlivým účtům, lze ale upravovat některá práva uživatelským rolím v nastavení aplikace. Změna se tak promítne na právech všech uživatelů patřících do skupiny. Aplikace a její databázová struktura počítá s případným rozšířením a obsahuje tedy tabulky práv. Většinu práv však zatím nelze v aplikaci změnit, z důvodu časové a implementační náročnosti, pro názornost byla přidána alespoň možnost měnit uživatelským skupinám přístupová práva pro prohlížení ostatních uživatelských účtů. Změnu práv dalších položek

lze naprogramovat stejným použitým algoritmem, pouze s jinými parametry. Význam jednotlivých položek v tabulce práv je uveden v následující tabule (Tabulka 6).

Tabulka 6 – Význam položek z tabulky práv

zkratka	celý název	význam
admin	Administrátor	Jedná se o administrátorský účet?
p_č	Projekt čtení	Může uživatel číst projekty?
p_z	Projekt změna	Může uživatel měnit projekt?
p_p	Projekt psaní	Může uživatel přidávat projekt?
r_č	Report čtení	Může uživatel prohlížet reporty?
r_z	Report změna	Může uživatel měnit reporty?
r_p	Report psaní	Může uživatel přidávat reporty?
č_p	Články psaní	Může uživatel přidávat články?
č_č	Články čtení	Může uživatel články zobrazit?
prior	Priority	Může uživatel měnit priority reportů?
kom	Komentáře	Může uživatel komentovat články?
uživ	Uživatelská tabulka	Jaký typ uživatelské tabulky je uživateli přístupný?

Kompletní tabulka práv v základním nastavení je zobrazena níže (Tabulka 7), jednotlivé zkratky jsou rozepsány v Tabulce 6, nebudu je tedy zde již rozepisovat. Jednička ve většině případů znamená, že je právo roli uděleno. Výjimkou je sloupec **uživ**, který udává, jaký typ uživatelské tabulky je pro skupinu přístupný.

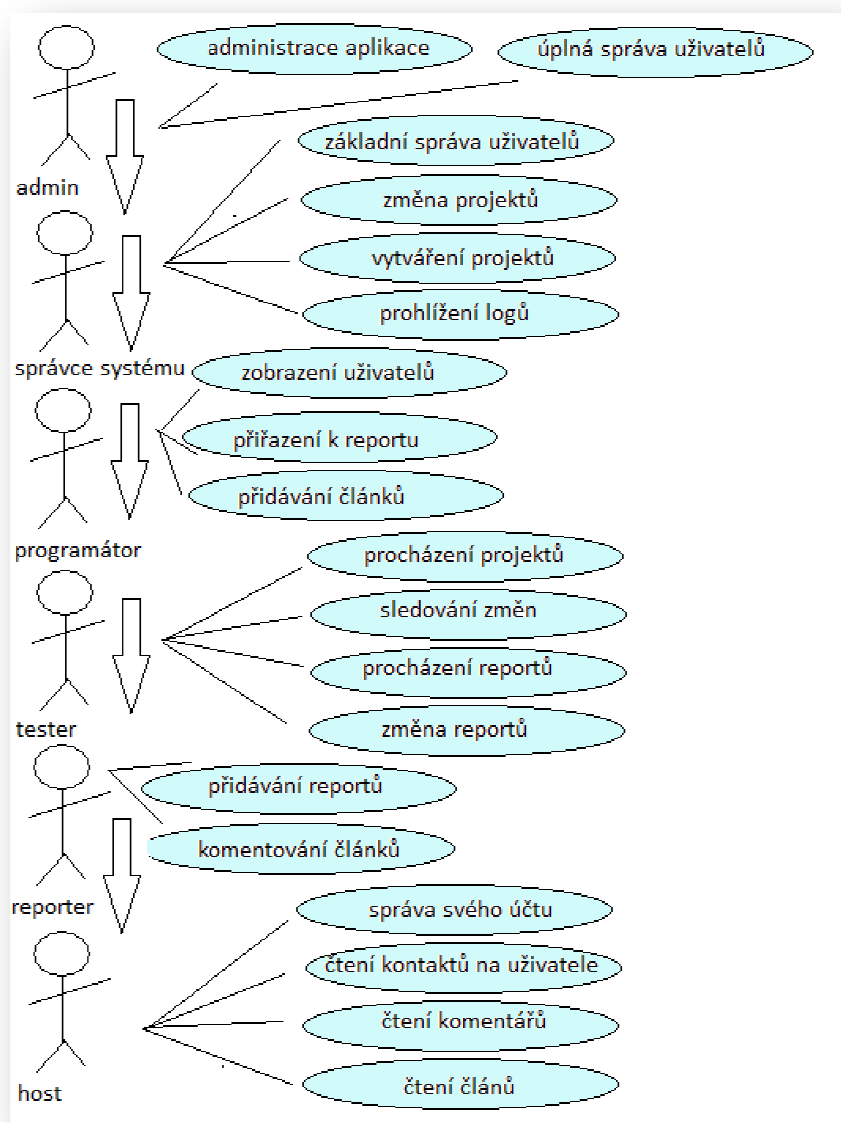
Tabulka 7 - Uživatelské role v Golem project manageru

skupina	id	admin	p_č	p_z	p_p	r_č	r_z	r_p	č_p	č_č	prior	kom	uživ
Admin	1	1	1	1	1	1	1	1	1	1	1	1	2
Správce	2	-	1	1	1	1	1	1	1	1	1	1	2
Programátor	3	-	1	-	-	1	1	1	-	1	1	1	2
Tester	4	-	1	-	-	1	1	1	-	1	1	1	2
Reporter	5	-	-	-	-	1	-	1	-	1	-	1	1
Host	6	-	-	-	-	-	-	-	-	1	-	-	1
Blokace	7	-	-	-	-	-	-	-	-	-	-	-	0

Z tabulky lze zpozorovat, že každá uživatelská skupina má práva předchozí skupiny a nějaké navíc. Rozdíl mezi hostem a blokováným uživatelem je, že blokovánému uživateli není povoleno přihlášení. Rozdíl mezi administrátorem a správcem projektu je především v tom, že správce projektu nemůže dalším uživatelům nastavit vyšší uživatelskou skupinu než má on sám, tudíž práva administrátora může udělit pouze administrátor.

Pro upřesnění uživatelských práv a jejich možností je přiložen Use case diagram na obrázku (Obrázek 10). Diagram je pouze orientační, protože některá přístupová práva

nejsou pevně stanovena a dají se v průběhu práce s aplikací měnit, dle požadavků a potřeb uživatelů. Uživatel s vyšší uživatelskou rolí má vždy všechny možnosti uživatelů pod ním.



Obrázek 10 - Use case diagram

7 Architektura Golem project manageru

7.1 Přihlášení do aplikace

Přihlášení do aplikace je řešeno jako součást hlavní stránky index.php. Pokud systém detektuje, že uživatel není přihlášen, zobrazí okno pro přihlášení do aplikace patrné na obrázku (Obrázek 11).



Obrázek 11 - Přihlašovací okno

Na přihlašovací stránce lze zadat uživatelské jméno a heslo, které budou použity pro přihlášení k systému. Na zápatí je odkaz na vytvoření nového účtu a zobrazení seznamu administrátorů a kontaktů na ně. Hesla jsou kódována algoritmem SHA, a proto jsou pro uživatele s přístupem k databázi nečitelná. Po úspěšném přihlášení jsou vytvořena cookies, která ukládají základní informace o přihlášení do paměti prohlížeče:

```
$_SESSION['uzivatel'] = $JMENO;  
$_SESSION['registrovan'] = $row["SKUPINY_ID"];  
$_SESSION['naposled'] = $row["NAPOSLED"];  
$_SESSION['aktivni'] = $_SESSION['naposled'];
```

Po každém pokusu o přihlášení je kromě funkce login volána také funkce loginlogsave, která ukládá veškeré informace o přihlášení, jako je například jméno uživatele, datum přihlášení, IP adresa či zda přihlášení proběhlo v pořádku. Tyto informace jsou dále pojmenovány jako logy a je s nimi dále pracováno. Logy neukládají heslo uživatele, ale zaznamenávají, zda bylo zadáno heslo správně.

7.2 Registrace nového uživatele register.php

Okno registrace nového uživatele je patrné z dalšího obrázku (Obrázek 12). Každý nově registrovaný uživatel je automaticky zařazen do skupiny návštěvník.

vítejte v
GOLEM!
©Josef Jadrný 2010

PROJECT MANAGERU

REGISTRACE NOVÉHO UŽIVATELE

zadejte vaše přihlašovací údaje

jmeno:

heslo:

heslo ověření:

e-mail:

[zpět](#)

Pro vytvoření nového účtu klikněte na [zaregistrovat](#).
Pro plnou aktivaci účtu kontaktujte administrátora [seznam](#).

Obrázek 12 - Registrace nového uživatele

Po úspěšném zaregistrování uživatele je možné ihned účet využít k přihlášení. Účtu s uživatelskou rolí návštěvník nebo také host je však většina aktivit zakázána, v základním nastavení systému má však přístup k zobrazení tabulky s administrátorskými účty a jejich kontakty. Zde může vyhledat potřebné informace a požádat o přidělení nové role. Právo na zobrazení tabulky uživatelů je však nastavitelné administrátory, takže nemusí být pro návštěvníky nutně k dispozici. V takovém případě nezbyvá, než vyplnit v nastavení účtu svoje kontakty a čekat, než vás některý z administrátorů kontaktuje sám.

K registraci nového uživatele je použita funkce register, její hlavní část:

```
$query = oci_parse($conn, "insert into UCTY values (:JMENO, :HESLO, :SKUPINY_ID, :EMAIL, :ICQ, :MOBIL, :REGISTROVAN, :NAPOSLED)");
oci_bind_by_name($query, ':JMENO', $JMENO);
oci_bind_by_name($query, ':HESLO', $HESLO);
oci_bind_by_name($query, ':SKUPINY_ID', $SKUPINY_ID);
oci_bind_by_name($query, ':EMAIL', $MAIL);
oci_bind_by_name($query, ':ICQ', $null);
oci_bind_by_name($query, ':MOBIL', $null);
oci_bind_by_name($query, ':REGISTROVAN', time());
oci_bind_by_name($query, ':NAPOSLED', $null);
```


Z dotazu lze vidět, že některé kontaktní údaje jsou při registraci nastaveny na nulu. Tyto položky lze pak ve správě účtu dodatečně doplnit.

7.3 Úvodní stránka Index.php

Po úspěšném zadání uživatelského jména a hesla je zobrazena hlavní stránka aplikace s uvítáním a registrem novinek (Obrázek 13).



Obrázek 13 - Regist změn

V registru jsou uvedeny informace o aktivitách uživatelů od posledního nahlédnutí přihlášeného uživatele do tohoto registru. Lze tak zjistit, kolik přibylo nových uživatelů, kolik se jich úspěšně přihlásilo, počet nových změn v projektech, počet nových článků a kolik přibylo komentářů.

Celkový vzhled úvodní obrazovky jsem již předvedl na Obrázku 9 v kapitole 6.4.

7.4 Správa článků clanky.php

V hlavním obsahovém okně se zobrazí seznam všech publikovaných článků, seřazených dle data publikování. U každého článku lze zobrazit jeho obsah a komentáře, jsou-li povoleny. Obsah článku se zobrazuje v okně s posledním psaným článkem.

Dále je dle práv přihlášeného uživatele na této stránce možnost publikovat nový článek, či komentovat již existující, případně mazat již existující článek a komentář. Jednotliví uživatelé bez dostatečných práv k mazání komentářů mohou mazat pouze komentáře vytvořené jejich účtem.

7.5 Správa uživatelů `uzivatele.php`

Zobrazí v hlavním okně seznam uživatelů a jejich přístupových práv a kontaktních údajů, tak jak jsou zaznamenány v databázi. Pro lepší pochopení je přiložena v aplikaci tabulka uživatelských skupin a práv tak, jak je zobrazena v Tabulce 7.

Uživatelům s dostatečným oprávněním dále umožní měnit uživatelům jejich skupinu a tím měnit jejich přístupová práva, administrátoři mají možnost blokace uživatele.

Vzhled a rozsah dat v uživatelské tabulce je závislý na typu přihlášeného uživatele a nastavení aplikace.

7.6 Správa projektů `projekty.php`

Zobrazí v tabulce seznam všech vytvořených projektů, pokud má uživatel práva na prohlížení projektu, bude mít navíc u každého projektu zobrazeno tlačítko umožňující přímý vstup k reportům v projektu.

Dále obsahuje pro správce projektů navíc i okno administrace projektu, kde lze projekt změnit, odstranit, či tisknout do souboru xml. Další změnou oproti ostatním uživatelům, je okno pro vložení nového projektu. Odstraněním projektu budou zároveň odstraněny všechny reporty a změny spojené s daným projektem tak, aby byla splněna referenční integrita.

7.7 Správa reportů `report.php`

V hlavním okně zobrazí nejprve seznam všech projektů s tlačítkem pro vstup do vybraného projektu. Při vybrání konkrétního projektu je navíc v tomto okně zobrazen seznam příslušných reportů, seřazených dle data vložení a obsahujících pouze základní informace. Dále je zde zobrazeno okno posledních změn, které zobrazuje maximálně 5 posledních změn v projektech. Pro uživatele s dostatečným oprávněním obsahuje i editor pro rychlé vložení nového reportu do vybraného projektu.

Jednotlivé reporty jsou v aplikaci vybarveny barvou tak, aby bylo graficky znázorněno, v jakém stavu se nacházejí (Tabulka 8).

Tabulka 8 - Stavy reportů

název	Barva v HTML
nový	#ccccff
znovu otevřený	#ffcccc
přiřazený	#cccccc
vyřešený	#ccffcc
uzavřený	#cccccc

7.8 Detaily vybraného reportu reportdetail.php

Toto okno je z hlavní nabídky nedostupné, obsahuje kompletní výpis jednoho reportu se všemi změnami na něm provedeném, po dobu jeho existence v databázi. Uživatelům s potřebnými právy zároveň zobrazí jednotlivá tlačítka, potřebná k editaci a smazání uložených údajů. Všechny změny zde provedené jsou zapisovány do databáze a zobrazitelné v aplikaci. Toto pravidlo nelze nikterak obejít bez přístupu k databázi.

Pro názornost uvádím tabulku priorit reportů (Tabulka 9).

název	Barva v HTML
minimální	default
nízká	default
normální	default
vysoká	red
blokující	red

7.9 Sledování logů logy.php

Toto okno je přístupné pouze administrátorům a správcům systému a zobrazí maximálně dvacet posledních přihlášení do systému, včetně detailů přihlášení. Příklad výpisu lze vidět na obrázku.

Systém vypíše maximálně posledních 20 přihlášení, kompletní výpis získáte kliknutím na tlačítko **KOMPLETNÍ SEZNAM**

ID logu	Datum	uživatel	IP	login
664	19/04 21:07	chose	10.128.4.184	OK
663	19/04 21:07	chozc	10.128.4.184	BAD
662	19/04 21:07	nevim	10.128.4.184	BAD
661	19/04 21:06	test2	10.128.4.184	OK
642	19/04 20:34	chose	10.128.4.184	OK
641	19/04 20:34	chozc	10.128.4.184	BAD
621	19/04 19:44	test	10.128.4.184	OK
601	19/04 18:20	admin	10.128.4.184	OK
581	19/04 13:19	admin	10.128.4.178	OK
562	06/04 18:53	admin	10.128.4.45	OK
561	06/04 18:52	admin	10.128.4.45	OK
541	06/04 14:29	admin	10.128.4.56	OK
525	01/04 18:46	chose	10.128.4.227	OK
524	01/04 18:46	test2	10.128.4.227	OK
523	01/04 18:46	admin	10.128.4.227	OK
522	01/04 18:43	test2	10.128.4.227	OK
521	01/04 18:39	admin	10.128.4.227	OK
509	31/03 19:12	chose	10.128.4.208	OK

KOMPLETNÍ SEZNAM

Obrázek 14 – Výpis logů

System ukládá pořadí logu, datum přihlášení, uživatelské jméno, IP adresu a zdali proběhlo přihlášení úspěšně. Kompletní seznam logů je možno vypsát do souboru a poté zobrazit nebo uložit do souboru.

7.10 Nastavení účtu nastaveni.php

V tomto okně lze upravovat nastavení přihlášeného uživatele. Lze doplnit nebo změnit kontaktní údaje, uživatelské heslo nebo případně účet odstranit. Okno pro nastavení účtu je zobrazeno na následujícím obrázku (Obrázek 15).

Vítejte v administraci vašeho účtu

KONTAKTNÍ ÚDAJE

uživatelské jméno: admin

uživatelská skupina: Administrator [změnit](#)

váš email:

váše icq:

váš telefonní kontakt:

ODSTRANĚNÍ ÚČTU

heslo:

ZMĚNA HESLA

staré heslo:

nové heslo:

Obrázek 15 - Nastavení účtu

7.11 Nastavení Golema nastavenigolem.php

Jedná se o nastavení dostupné pouze administrátorům. V tomto nastavení lze změnit některé vlastnosti uživatelských rolí a vymazat veškeré logy. Případné další změny, týkající se práv uživatelských rolí, by byly implementovány právě zde.

7.12 Další soubory

Součástí aplikace je mnoho dalších souborů, ať už kvůli zabezpečení nebo k realizaci grafického prostředí pomocí odstavců. Některé z výše uvedených částí systému používají další soubory ke své funkčnosti. Těchto souborů je velice mnoho a nebudu je zde dále rozepisovat.

8 Zabezpečení aplikace

8.1 Systém rolí

Zobrazený obsah stránky je vždy závislý na typu uživatele, který je uložený v session. Při načtení každé stránky je tedy kontrolováno, o jakou uživatelskou roli se jedná a podle toho zobrazí obsah stránky. Takzvané hidden prvky, které jsou jednoduše zneužitelné, systém nezobrazuje, pokud na jejich použití uživatel nemá právo. Příklad:

```
if (isset($_SESSION['registrovan']) && $_SESSION['registrovan'] <= 2)
{
echo
    '<form action="clanky.php" method="post">
    <input type="submit" value="Odstranit">
    <input type="hidden" name="odstranit" value="' . $row["NÁZEV"] . "'>
    </form></TD></TR>' . NL;
}
```

V některých případech je zabezpečeno stejným testem i samotné zachycení události (například kliknutí na odstranit). Dále je použita kontrola, zdali je uživatel stále přihlášen:

```
function test(){
    if(!isset($_SESSION['uzivatel'])) return die("chyba autorizace");
    if(!isset($_SESSION['registrovan'])) return die("chyba autorizace");
}
```

8.2 Práce s hesly

Pro zabezpečení přihlašovacích údajů jsem zvolil hašovací funkci SHA1[5], která vytvoří z hesla otisk (hash) dlouhý 160 bitů, ze kterého už nelze heslo dostat zpět. Proto při ověřování správnosti zadaného hesla, se toto heslo musí funkcí SHA1 taktéž zakódovat a poté porovnat hash ze zadaného hesla s hashem uloženým v databázi. SHA je náhradou za starší MD5.

8.3 Formuláře

Velice nebezpečnou částí aplikace jsou formulářové vstupy. K vyřešení tohoto problému jsem k dotazování na databázi použil interface OCI, který tuto problematiku řeší a dále se tímto problémem nezabýval.

Některé formuláře jsou navíc kontrolovány javascripty na jejich vyplnění, aby nedocházelo k zbytečným chybám:

```

function KontrolaRegister(f) {
    var vysledek = true;
    var strerr = "";
    if (f.login.value=="") strerr += "Jméno nevyplněno\n";
    if (f.password.value=="") strerr += "Heslo nevyplněno\n";
    if (f.password2.value=="") strerr += "Heslo overení
nevyplněno\n";
    if (f.email.value=="") strerr += "Email nevyplněn\n";
    if (f.password.value!=f.password2.value) strerr += "Hesla
nejsou stejná";
    if (" " != strerr) {
        vysledek = false;
        alert("CHYBA:\n\n" + strerr);
    }
}

```

8.4 Porušení referenční integrity

Referenční integrita [6] je nástroj databázového stroje, který pomáhá udržovat vztahy v relačně propojených databázových tabulkách. Referenční integrita se definuje cizím klíčem, a to pro dvojici tabulek, nebo nad jednou tabulkou, která obsahuje na sobě závislá data (například stromové struktury). Tabulka, v níž je pravidlo uvedeno, se nazývá podřízená tabulka (používá se také anglický termín slave). Tabulka, jejíž jméno je v omezení uvedeno je nadřízená tabulka (master). Pravidlo referenční integrity vyžaduje, aby pro každý záznam v podřízené tabulce, pokud tento obsahuje data vztahující se k nadřízené tabulce, odpovídající záznam v nadřízené tabulce existoval. To znamená, že každý záznam v podřízené tabulce musí v cizím klíči obsahovat hodnoty odpovídající primárními klíči nějakého záznamu v nadřízené tabulce, nebo NULL.

K porušení referenční integrity dochází v případě, že se pokoušíme měnit či odstraňovat data, na která se odkazují data jiná. Pokud by příkazem došlo k porušení referenční integrity, vrátí funkce číselný chybový kód, který je aplikací zachycen a zpracován.

Aby nedocházelo k neustálému mizení dat z databáze tím, že se uživatel rozhodne odstranit svůj účet (v takovém případě, je totiž nejdříve nutné odstranit všechny položky, které se na takový účet vztahují, tedy například projekty, které uživatel založil), není účet nikdy doopravdy smazán, ale pouze prohlášen za zablokovaný, tak aby se z databáze nemusely odstraňovat související položky.

8.5 Záloha účtů

Aplikace také obsahuje jednoduchý trigger, který automaticky zálohuje všechny účty smazané přímo v databázi. Tyto data lze později využít pro případné obnovení účtu, nebo také jako adresář kontaktů na bývalé uživatele pro případ, kdyby bylo nezbytné některého z uživatelů později kontaktovat:

```

create or replace
TRIGGER "POTVRZENI"
BEFORE delete ON ucty
FOR EACH ROW
BEGIN
    INSERT INTO UCTY_SMAZANE VALUES (:OLD.JMENO, :OLD.HESLO,
    :OLD.SKUPINY_ID, :OLD.EMAIL);
END;

```

8.6 Systém zachytávání chyb

Do aplikace byl implementován i jednoduchý systém pro zachytávání chyb, kdy většina funkcí vrací nějakou číselnou hodnotu, která představuje úspěšnost jejího splnění. Tento kód je v aplikaci zachycen a zpracován směrem k uživateli tak, aby bylo jasné, zdali nedošlo při provádění k chybám. Příklad zobrazení chybového hlášení je vidět na obrázku (Obrázek 16) a obsah celého souboru pro zpracování chyb, včetně chybových hlášení je jako příloha mé práce.

The image shows a web interface for a project manager. At the top, there is a green banner with the text "vítejte v GOLEM! @Josef Jadrný 2010" and "PROJECT MANAGERU" next to a cartoon penguin logo. Below the banner, a red error message bar reads "Chyba přihlášení, zadali jste správné heslo? (ERROR100)". Underneath is a login form titled "PŘIHLÁŠENÍ DO SYSTÉMU" with the instruction "zadejte vaše uživatelské jméno a heslo". The form contains two input fields and a "Přihlásit" button. At the bottom of the page, there are two links: "Pro vytvoření nového účtu klikněte na [zaregistrovat](#)." and "Pro plnou aktivaci účtu kontaktujte administrátora [seznam](#)".

Obrázek 16 – Ukázka chybového systému

9 Závěr

Mým cílem bylo vytvořit jednoduchou aplikaci pro správu chyb a projektů tak, aby uměla všechny zásadní funkce, které by taková aplikace umět měla, včetně integrovaného redakčního systému a jednoduchého přehledného grafického rozhraní, které se liší dle typu přihlášeného uživatele tak, aby i méně zkušený uživatel neměl problémy při jejím používání.

Tyto cíle byly z větší části splněny. Jistě by se po delším provozování aplikace vyskytly nějaké chyby, které mi unikly. Otázkou je také optimalizace kódu, kde by nebylo na škodu více využívat funkcí, triggerů a procedur, místo neustálého kopírování podobných selectů. Dalším otazníkem je zabezpečení aplikace, kde by dle mého názoru mohlo být několik slabých míst. Aplikace je ale připravena k dalšímu rozšiřování, především díky tomu, že jsem se snažil všechny funkce shromáždit na jednom místě kvůli přehlednosti. Případné další rozšíření bych směřoval do podpory mailování, také do větší rozlehlosti a variability systému tak, aby žádné údaje nebyly v databázi uloženy na pevně, ale daly se libovolně měnit, dle požadavků uživatelů.

Porovnání s jinými bug tracking systémy by bylo zbytečné, jen těžko se dá očekávat, že student třetího ročníku vysoké školy vytvoří za pár měsíců lepší a komplexnější program, než skupiny zkušených programátorů za několik let. Svůj účel ale Golem project manager určitě splní.

10 Použité zdroje:

- [1] *Programování ve velkém*. [online]. Wikipedia. Naposledy editováno 2.9.2010 [cit. 2011-04-07]. Dostupné z WWW: <http://cs.wikipedia.org/wiki/Programování_ve_velkém>.
- [2] *Mantis Bug Tracker*. [online]. Wikipedia. Naposledy editováno 1.4.2011 [cit. 2011-04-07]. Dostupné z WWW: <http://en.wikipedia.org/wiki/Mantis_Bug_Tracker>.
- [3] *Bugzilla*. [online]. Wikipedia. Naposledy editováno 1.4.2011 [cit. 2011-04-07]. Dostupné z WWW: <<http://en.wikipedia.org/wiki/Bugzilla>>.
- [4] *Redmine*. [online]. Wikipedia. Naposledy editováno 1.4.2011 [cit. 2011-04-07]. Dostupné z WWW: <<http://en.wikipedia.org/wiki/Redmine>>.
- [5] LUBOSLAV Lacko. SQL - Hotová řešení. [s.l.] : COMPUTER PRESS, 2003. 298 s. ISBN 80-7226-975-5.
- [6] THOMAS Conolly, CAROLYN Begg, RICHARD Holowczak. Mistrovství Databáze - Profesionální průvodce tvorbou efektivních databází. [s.l.] : COMPUTER PRESS, 2007. 584 s. ISBN 978-80-251-2328-7.
- [7] DRUSKA, Peter. CSS a XHTML : tvorba dokonalých webových stránek krok za krokem. [s.l.] : Grada Publishing, 2006. 200 s. ISBN 80-2471-382-9.
- [8] KOSEK, Jiří. PHP: tvorba interaktivních internetových aplikací. [s.l.] : Grada Publishing, 1999. 490 s. ISBN 80-7169-373-1.
- [9] *Bug tracking system*. [online]. Wikipedia. Naposledy editováno 1.4.2011 [cit. 2011-04-07]. Dostupné z WWW: <http://en.wikipedia.org/wiki/Bug_tracking_system>.

Příloha A zdrojový kód se seznamem chybových hlášení

```
<?php
include 'webcheck.php';
function error($error) {
    global $config;
    echo
        ' <div id="prostredni_ERROR">
          <div class="ERROR_content">' . NL;
    if ($error == 100) echo "Chyba přihlášení, zadali jste správné heslo?
(ERROR100)";
    if ($error == 101) echo "Komentář se nepodařilo vložit (ERROR101)";
    if ($error == 102) echo "Projekt se nepodařilo odstranit (ERROR102)";
    if ($error == 103) echo "Jeden nebo více reportů se nepodařilo odstranit,
možná chyba integrity (ERROR103)";
    if ($error == 104) echo "Uživatel již existuje (ERROR104)";
    if ($error == 105) echo "Uživatel úspěšně zaregistrován <BR>klikněte <a
href=index.php>ZDE</a> pro návrat na přihlašovací stránku<BR>";
    if ($error == 106) echo "Uživatele se nepodařilo registrovat (ERROR106)";
    if ($error == 107) echo "Článek se nepodařilo vložit (ERROR107)";
    if ($error == 108) echo "Nepodařilo se odstranit související komentáře,
možná chyba integrity (ERROR108)";
    if ($error == 109) echo "Článek se nepodařilo odstranit (ERROR109)";
    if ($error == 110) echo "Komentář se nepodařilo odstranit (ERROR110)";
    if ($error == 111) echo "Projekt se nepodařilo aktualizovat (ERROR111)";
    if ($error == 112) echo "Soubor vytvořen.<BR>klikněte <a href=data.xml
target=_blank>ZDE</a> pro stažení<BR>";
    if ($error == 113) echo "Projekt se nepodařilo vložit (ERROR113)";
    if ($error == 114) echo "Report se nepodařilo vložit (ERROR114)";
    if ($error == 115) echo "Report vložen, chyba při zapisování změn
(ERROR115)";
    if ($error == 116) echo "Uživatele se nepodařilo aktualizovat
(ERROR116)";
    if ($error == 117) echo "Nepodařilo se načíst seznam skupin a uživatelů
(ERROR117)";
    if ($error == 118) echo "Report se nepodařilo aktualizovat (ERROR118)";
    if ($error == 119) echo "Report aktualizován, při zápisu změn došlo k
chybě (ERROR119)";
    if ($error == 120) echo "Soubor vytvořen.<BR>klikněte <a href=logs.xml
target=_blank>ZDE</a> pro stažení<BR>";
    if ($error == 121) echo "Chyba při ukládání dat o přihlášení (ERROR121)";
    if ($error == 122) echo "Kritická chyba autorizace (ERROR122)";
    if ($error == 123) echo "Na přidělení administrátora nemáte dostatečná
oprávnění (ERROR123)";
    if ($error == 124) echo "Kritická chyba, nemáte dostatečná oprávnění pro
tuto akci (ERROR124)";
    if ($error == 125) echo "Některé vaše kontaktní údaje nejsou vyplněné,
důrazně doporučujeme jejich doplnění (ERROR125)";
    if ($error == 126) echo "Některé vaše kontaktní údaje nejsou vyplněné,
důrazně doporučujeme jejich doplnění (ERROR126)";
    if ($error == 127) echo "Váš email není nastaven, pro správnou funkčnost
aplikace doporučujeme email doplnit (ERROR127)";
    if ($error == 128) echo "Email nelze odstranit (ERROR128)";
    if ($error == 129) echo "Neplatný účet, pokračujte <a
href=index.php>ZDE</a>(ERROR129)";
    if ($error == 130) echo "Uživatele se nepodařilo odstranit (ERROR130)";
    if ($error == 131) echo "Chyba, nesouhlasí heslo (ERROR131)";
    if ($error == 132) echo "Heslo nelze nechat prázdné (ERROR132)";
```

```
if ($error == 133) echo "Nepodařilo se identifikovat některá práva
uživatele (ERROR133)";
if ($error == 134) echo "Logy se nepodařilo odstranit (ERROR134)";
if ($error == 135) echo "Práva se nepodařilo upravit (ERROR135)";
if ($error == 136) echo "Nastavení pro blokované uživatele bylo
přeskočeno (ERROR136)";
    echo '
</div>
                </div>' . NL;
}
?>
```