

UNIVERZITA PARDUBICE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

BAKALÁŘSKÁ PRÁCE

2009

Raur Martin

Univerzita Pardubice

Fakulta elektrotechniky a informatiky

Systém pro sledování podnikových projektů softwarové firmy

Raur Martin

Bakalářská práce

2009

University of Pardubice
Faculty of Electrical Engineering and Informatics

The system for the software company's business projects monitoring.

Raur Martin

Bachelor Work

2009

Prohlašuji:

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

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

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně Univerzity Pardubice.

V Pardubicích dne 14. 05. 2009

.....
(vlastnoruční podpis)

SOUHRN

Implementace projektového řízení, s použitím technologií pro tvorbu webových aplikací a databáze.

KLÍČOVÁ SLOVA

Projektové řízení, Web, PHP, MySQL

TITLE

The system for monitoring enterprise software company projects.

ABSTRACT

Implementation of project management, using technology for creating Web applications and databases.

KEYWORDS

Project management, Web, PHP, MySQL

PODĚKOVÁNÍ

Největší díky patří mému vedoucímu bakalářské práce panu Ing. Lukáši Čeganovi za pomoc, velikou trpělivost při řešení problémů spojené s touto prací. A samozřejmě rodině a přátelům za jejich rady a podporu

Seznam obrázku	8
1 Úvod.....	9
2 Motivace.....	10
2.1 Cíle práce	10
2.2 Konkrétní podoba výsledku práce.....	10
2.3 Existující implementace.....	11
2.4 Zhodnocení současných systémů na trhu.....	13
3 Teorie projektového řízení.....	14
3.1 Projektový management.....	14
3.2 Cyklus projektového řízení	15
3.3 Softwarová podpora pro řízení projektu	17
3.4 Vytyčení teoretických východisek pro aplikaci	17
4 Vývojové technologie	18
4.1 HTML/XHTML	18
4.2 CSS.....	21
4.3 PHP	22
5 Analýza a návrh řešení.....	24
5.1 Základní charakteristika aplikace.....	24
5.2 Adresářová struktura aplikace.....	24
5.3 Práva v aplikaci	25
5.4 UML diagramy.....	26
5.4.1 Zjednodušené aktivity diagram.....	26
5.4.2 UML use case diagram.....	29
5.5 Návrh databáze aplikace	29
6 Instalace aplikace	35
7 Uživatelská dokumentace.....	36
7.1 Programátor.....	36
7.2 Manažer.....	38
7.3 Administrátor	40
8 Vývoj a možné rozšíření aplikace.....	42
9 Závěr	43
10 Bibliografie	44
Obsah příloženého CD	45

Seznam obrázku

- [1] Obr. 1 Cyklus projektového řízení
- [2] Obr. 2 Adresářová struktura
- [3] Obr. 3 Dílčí ER diagram systému oprávnění
- [4] Obr. 4 Aktivita diagram role programátora
- [5] Obr. 5 Aktivita diagram role manažera
- [6] Obr. 6 Aktivita diagram role administrátora
- [7] Obr. 7 Use case diagram
- [8] Obr. 8 Struktura databáze
- [9] Obr. 9 Přihlašovací dialog
- [10] Obr. 10 Výpis projektů pro programátora
- [11] Obr. 11 Výpis projektů manažera
- [12] Obr. 12 Karta úkolu
- [13] Obr. 13 Výpis uživatelů
- [14] Obr. 14 Nový úkol
- [15] Obr. 15 Přřadit dovednost

1 Úvod

Každý podnik, který chce správu nad podnikovými projekty vykonávat na vyšší úrovni, po nějaké době začne, pociťovat potřebu vhodného programu, který mu pomůže udržet potřebný přehled. S přibývajícím rozsáhlostí a složitostí projektů stane nezbytností možnost, rozdělit složitější projekt na jednodušší úkoly a ty poté přidělit více pracovníkům a tím zajistit, že daný závazek bude splněn ve stanoveném termínu. Další nezbytností je bezpochyby poskytnutí pracovníku důležité informace potřebné k jasné představě o náplni projektu nebo úkolu, který mu byl přidělen.

Pro tyto účely jsem vytvořil aplikaci pro správu podnikových projektů, která bude napomáhat pracovníkům ve vedoucích pozicích efektivně využívat všech zdrojů, které podnik má a to jak lidských, technologických tak informačních.

Jako první je třeba si uvědomit, že aplikace tohoto typu bude sloužit především pro monitorování podnikových projektů a činností, které na nich probíhají. S její pomocí bude možné, jakkoliv složitý projekt dekomponovat na elementární úkoly, vhodné pro rozdělení mezi jednotlivé pracovníky. Kompletní delegování pracovní náplně zaměstnanců. Bude schopna udržet náležitý přehled vedoucího pracovníka nad projekty či úkoly, které mu byly přiděleny. Umožňovat, aby do skupiny úkolů (projekt) dělalo zásahy více lidí, zabezpečuje důležitou komunikaci jak mezi zaměstnanci, tak mezi podnikem a zákazníkem.

Na druhé straně je třeba brát na vědomí, že primárním úkolem tohoto nástroje je správa projektů a řízení činností. Bylo by zbytečné, zde implementovat moduly umožňující například správu zdrojových kódů projektu, nebo jejich kompilaci, na tuto problematiku existují specializované nástroje. Systém též neprovádí testování či jakoukoliv jinou kontrolu zpracovávaných projektů či úkolů, čímž by nehradil vedoucího projektu či jiný kontrolní článek.

2 Motivace

2.1 Cíle práce

Hlavním cílem práce je vytvořit jednoduchou a snadno rozšiřitelnou webovou aplikaci, která nebude náročná, jak na obsluhu, tak na výkon serveru, na kterém bude spuštěna. Bude určena zejména pro malé až střední podniky, zejména tam kde podnik zaměstnává větší množství jak interních tak externích pracovníků. Pro manažera takového podniku, by bylo velmi obtížné mít představu o činnosti svých podřízených. K tomuto účelu bych chtěl vytvořit aplikaci na správu podnikových projektů, která by přehledně a jednoduše umožnila manažerovi, rozdělovat závazky firmy mezi zaměstnance.

Ovšem problematika správy projektů je však tak obsáhlá, že vytvořit zcela kompletní aplikaci, které by pokryla všechny požadavky malé až střední firmy přesahuje možnosti bakalářské práce, časových možností i znalostí jednoho člověka. Cílem této práce je tedy navrhnout a vytvořit kvalitní jádro aplikace, které bude možné v budoucnu snadno rozšířit a další možnosti dle konkrétní potřeby. Vlastní jádro aplikace bude řešit především:

- ✓ Správu podnikových zaměstnanců
- ✓ Správu projektů, jejich dělení na úkoly
- ✓ Přidělování jednotlivých úkolů zaměstnancům
- ✓ Plánování činnosti
- ✓ Hlídání docházky a činností zaměstnanců
- ✓ Správu podnikových dokumentů
- ✓ Možnosti komunikace

2.2 Konkrétní podoba výsledku práce

Je to běžná webová aplikace, napsaná převážně ve skriptovacím jazyce PHP, která běží na intranetové serveru, přistupuje se k ní pomocí kteréhokoliv webového prohlížeče. Ke své funkčnosti nezbytně potřebuje, aby na serveru byla spuštěna databáze MySQL.

Aplikace se velmi snadno obsluhuje a to následovně:

1. Přihlášení uživatele
2. Načtení práv k souborů a možnostem aplikace
3. Další činnost se odvíjí od zobrazeného menu (například zkontrolovat dokončení projektu, začít pracovat na zadaném úkolu, přidělit úkol apod.)
4. Případné naspaní zprávy či komentáře
5. Odhlášení

2.3 Existující implementace

V současné době je velice široký výběr, aplikací tohoto zaměření, většinou se ale jedná o komerční produkty. Pokud najdeme nějaký open-source, velice často má omezené možnosti, převážně se jedná o plánovací systémy k ukládání předem definovaných události do kalendáře. Další velice rozšířeným systémem jsou souborové systémy odsluhující datové uložení. Tyto aplikace velice dobře zvládají práci se soubory, zálohování či ukládání více verzí projektu.

Z těchto produktů zde u vedu jen některé vybrané příklady:

- ✓ **xWORK** poskytuje modulární intranetové řešení pro středně velké i malé firmy. Jeho nasazení pomůže zlepšit koordinaci pracovních procesů uvnitř firmy. Systém umožňuje sdílený přístup k informacím přes intranet i Internet, což přináší společnosti on-line přístup odkudkoliv. Aplikace disponuje moduly pro projekty a úkoly, diář a plánování, správu uživatelů, celofiremní adresář, kontakty, docházku, správu dokumentů, tvorba statistik a reportů. Aplikace je poskytována formou pronájmu na internetovém serveru výrobce, kterým je SOVA NET s.r.o. Výše toho pronájmu závisí na modulech, které si zákazník vybere, základní cena je 990,-Kč měsíčně.
- ✓ **Elaine MC** je výkonným nástrojem pro správu projektů po celou dobu jejich životního cyklu, dále obsahuje grafický nástroj pro plánování činnosti pracovníků na projektech - při plánování delších projektů je možno pracovat s celými týdny místo dní. Systém porovnává zadané hodnoty s celkovým časovým plánem a zabraňuje tak chybám v plánování kapacit. K vytvořenému plánu

je možno připojit komentář, který se pak objeví kdykoliv po najetí kursoru nad příslušné políčko plánovací tabulky. Dalším nástrojem je správa pracovních výkazů . Ke každé činnosti je nutné vyplnit formulář popisující pracovní náplň, dle informací na stránkách toho produktu může vykazání celodenní práce, trvat doslova pár vteřin. Nástroj pro správu uživatelů zajišťuje v rámci systému Elaine MC evidenci všech důležitých údajů o zaměstnancích. Na jejich základe je možno, spravovat uživatelské účty ve smyslu přidělování různých práv nad projekty nebo jejich částí jednomu konkrétnímu uživateli. Systém, dále poskytuje množství grafů a srovnávacích tabulek formou informativních reportů. Základní verze tohoto produktu stojí 59400,- Kč a je poskytnuta roční záruka na celý systém.

- ✓ **Microsoft Office Project Professional** je aplikace společnosti Microsoft pro správu podnikových projektů, která se používá v kombinaci s portálem Microsoft Office Project Web Access a serverem Microsoft Office Project Server. Aplikace Project Professional je určena pro vedoucí projektů, kteří potřebují plánovat projekty, přidělovat na úkoly členy týmu, centrálně ukládat informace na server pro sdílení s ostatními, zlepšování koordinace jednotlivých týmů a efektivnější využití pracovních sil. Aplikace Project Professional poskytuje všechny základní plánovací nástroje jakými jsou, přidělování pracovníků na projekty na základě schopností, sledování plnění úkolů a jejich spravování. Cena standardní verze tohoto produktu je 17152,- Kč a cena aktualizace je 9948,- Kč.
- ✓ **Easy software** je snadno nastavitelnou a uživatelsky jednoduchou webovou aplikaci, umožňující řídit a komunikovat se svými zaměstnanci přes Internet. Poskytuje přehled nad zadanými projekty odkudkoliv. Aplikace disponuje funkcemi pro přidělování úkolů jednotlivým zaměstnancům nebo členům projektového týmu pomocí jednoduchého dialogu, oznamování stavu nebo změnu stavu úkolů manažerovi, zobrazení časových diagramů úkolů, kalendář úkolů pro zaměstnance, kontrola docházky a zaznamenávání hodin strávených na jednotlivých úkolech, definice pracovních rolí a reporty pro manažery (vyřeše-

né/otevřené úkoly, odpracované hodiny atd.). Základní verze systému stojí od 19990,-Kč. Možnost požádat o ukázkovou verzi na 1 týden zdarma.

2.4 Zhodnocení současných systémů na trhu

Velice zajímavým řešením je produkt xWORK, který je distribuován formou pronájmu. Zákazník si nemusíte pořizovat hardwarové vybavení či se potýkat s instalací aplikace. Další výhody jsou, že aplikace je lépe zabezpečena proti zneužití, všechny data aplikace jsou automaticky zálohována jednou za 24 hod, v cena pronájmu jsou i všechny aktualizace a opravy, které jsou implementovány na jednom místě. Nevýhodou je, že všechny soubory a dokumenty týkající se zadání projektu, se musí přenášet přes Internet až ke klientovi. Rychlost tohoto přenosu je velice závislá na vytíženosti serveru a rychlosti Internetového připojení.

Elaine MC skvěle propracovaný systém s mnoha užitečnými funkcemi, má velice dobře provedený grafický plánovací kalendář a systém docházky. Jediný mínus této aplikace na vysoký cena.

Easy software vyniká svou jednoduchostí ovládání a velice dobrým graficky provedeným plánovačem. Je vhodný pro menší firmy, kde není potřebný tak robustný systém, jako je Elaine MC.

Microsoft Office Project Professional nabízí sice velice pokročilé funkce a řadu vylepšení a usnadnění administrativní činnosti, ovšem převážná část těchto výhod, oproti ostatním produktům, je závislá na spolupráci s dalšími produkty firmy Microsoft, jako jsou Microsoft Office Project Web Access a Microsoft Office Project Server, které jsou samozřejmě také komerční.

3 Teorie projektového řízení

V této kapitole se budu zabývat stěžejními principy projektového managementu a elementárními pojmy. Budu se zde snažit, představit základní filozofii tohoto velice komplikovaného oboru na základě, kterých budou zformulována východiska navrhované aplikace.

3.1 Projektový management

Hledáme-li rozdíly mezi běžně chápaným pojetím managementu, jaký prezentuje např. Mintzberg a projektovým managementem, vyjdeme z obecných definic managementu. První z nich podle jednoho z předních světových teoretiků projektového managementu profesora Herolda Kerznera, druhá vychází z teorií největšího a nejuznávanějšího světového profesionálního sdružení projektových managerů Project Management Institute, PMI®.

Projektový management

Projektový management je souhrn aktivit spočívající v plánování, organizování, řízení a kontrole zdrojů společnosti a relativně krátkodobým cílem, který byl stanoven po realizaci dříve specifikovaných cílů a záměrů.¹

Projektový management

Projektovým managementem je aplikace znalostí, schopností, nástrojů, a technologií na aktivity projektu tak, aby splnili požadavky projektu.

Přestože se svém doslovném znění obě definice značně odlišují, je jejich podstata velice podobná: projekt je jisté krátkodobé vynaložené úsilí podpořené aplikací znalostí a technologických metod, jehož účelem je přeměna materiálních a nemateriálních zdrojů na soubor výrobků či služeb nebo jejich kombinace tak, aby bylo dosaženo dříve vytyčených cílů.¹

Vynaložené úsilí společně s aplikovanými znalostmi a technologiemi je prezentováno jako organizované působení pěti základních elementů projektového managementu, kterými jsou:

¹ SVOZILOVÁ, Alena. *Projektový management : Systémový přístup k řízení projektu*. Praha 7 : Grada, 2006. 360 s. Expert. ISBN 80-247-1501-5.

- projektová komunikace - prostředí sloužící k efektivnímu dorozumění všech účastníků projektu,
- týmová spolupráce – principy náležité kooperace a důvěry ve smyslu dosažení společně stanovených cílů,
- životní cyklus projektu – představuje logický sled základních úseků a fází projektu, včetně definovaných stavů a podmínek pro přechod ze současné fáze na fázi následující,
- vlastní součásti projektového managementu – mezi techniky a nástroje pro řízení projektů aplikovatelných v průběhu životního cyklu jednoznačně patří:
 - ✓ požadavky projektu, koncepty, předpisy a omezení zadání
 - ✓ projektový tým – komunikace a spolupráce členů
 - ✓ projektová přehlednost – průběžná kontrola a informovanost
 - ✓ okamžitý stav projektu – soubor metod a postupů pro měření a kontrolu stavu projektu a informovanost a případné potřebě nasazení opatření
 - ✓ příležitosti a rizika – statistiky a hodnocení vlivů a dopadů ostatních a na ostatní subjekty
 - ✓ oprávněná opatření, kterými je možno napravit zjištěné chyby a odchylky
- organizační závazek obsahující pověření manažera řízením projektu, podporu založenou na organizační struktuře, finančních a jiných zdrojích určené k realizaci projektu

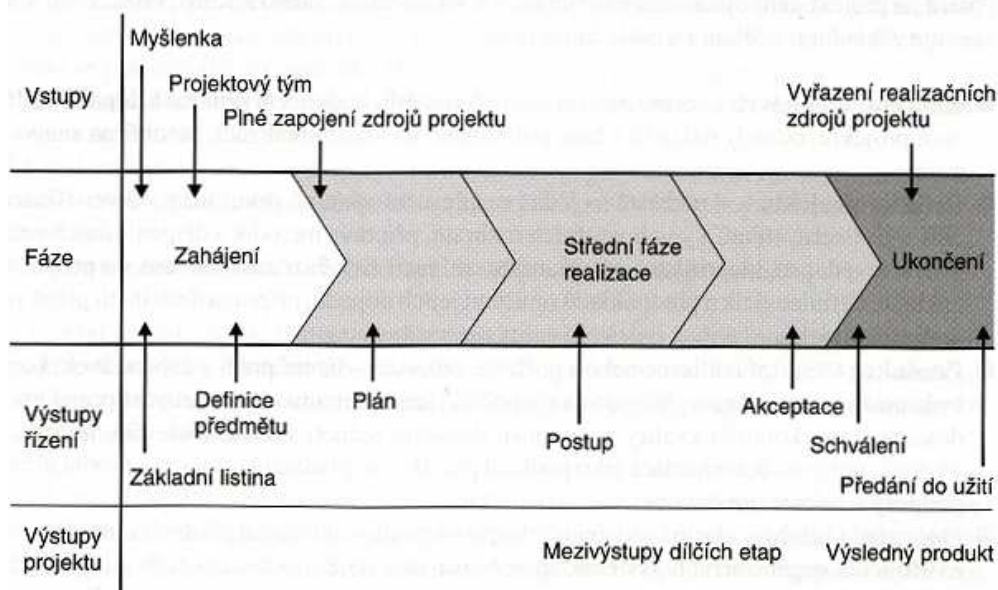
3.2 Cyklus projektového řízení

Projekt je prvek, který má charakter procesu, v době svého vývoje se nachází v různých fázích, které souhrnně můžeme nazvat životním cyklem každého projektu. Existuje mnoho různých definic životního cyklu, ale na jednotné definici se neshodli světový teoretici, hospodářské sektory ani jednotlivé společnosti. Zde se přikloním definici cyklu jak ji demonstrují Cleland a King (viz. níže).

Obecný popis životního cyklu projektu

Pokud zde aplikujeme teorii, jak ji popsali Cleland a King. Pak existuje základní rozdělení na následující fáze:

- Konceptuální návrh – formulace základních záměrů, hodnocení přínosů a dopadů realizace projektu, odhady nákladů a času potřebného na vlastní realizaci, předběžná analýza rizik.
- Definice projektu – v podstatě se jedná o zpřesnění výstupu první fáze, užší definování cílů, výčet podsystémů. Příprava metodik, potřebných znalostí a dovedností, identifikace zdrojů, sestavení reálného časového horizontu, rozpočtu nákladů a zvážení možných rizik.
- Produkce – vlastní realizace, řízení prací a subdodávek. Kontrola postupu podle časového harmonogramu a rozpočtu, řízení komunikace, kontrola kvality a účinnosti jednotlivých dosažených dílčích cílů, testování výstupu a pořízení nezbytné dokumentace
- Operační období – vlastní užívání produktu projektu. Zahrnuje jeho začlenění do již existujících organizačních systému, hodnocení technologických a ekonomických dopadů realizovaného projektu, zpětná vazba pro plánování dalších projektů a hodnocení spolupracujících systémů.
- Vyřazení projektu – převedení produktu do stádia podpory a do případné odpovědnosti organizace, která podporu poskytuje, předání firemních zdrojů (např. pracovníků nebo technologií) na jiné projekty, zpracování poučení a získaných zkušeností z řízení.[2]



Obr. 1

Cyklus projektového řízení

3.3 Softwarová podpora pro řízení projektu

Předchozích kapitolách jsem velice stručně popsal základní pojmy a vazby projektového managementu. Nyní se zaměřím na vhodnou volbu softwarové podpory pro projektové řízení.[2]

Pro řízení projektu se samozřejmě dají použít i obecné matematické a statistické nástroje, ovšem jejich aplikace na složitější projekty je spojena se značnou administrativní zátěží řídicího subjektu. Většina výše zmiňovaných metod, ať už v základní podobě nebo v obohacené formě, tvoří základní funkcionalitu programů, které mohou pomoci manažerovi projektu při výkonu jeho funkce.

Správná volba softwarového nástroje je velice důležitá a jedná se o nelehký úkol. Je třeba se rozhodnout na základě rozsahu softwarové podpory a jejího začlenění do podnikového informačního systému. Z tohoto podle se setkáváme s jednoduchými programy, které fungují většinou izolovaně a jsou určeny pro pokrytí základní potřeb pro řízení projektu menších rozsahů. Proti kladem jsou komplexní nástroje, které jsou součástí podnikového informačního systému, jsou vybaveny programovými balíky s rozsáhlou funkcionalitou potřebnou k plánování, kontrolu a optimalizace mnoha projektů různé úrovně složitosti.

3.4 Vytyčení teoretických východisek pro aplikaci

Zvýše uvedené teorie vyplývá, že pro správnou funkci aplikace spravující podnikové projekty je nutné, aby umožňovala základní prvky projektového managementu. Musí poskytnout určitou úroveň komunikace, jak mezi pracovníky, tak mezi pracovníky a managementem. Důležitá je možnost dekompozice složitého projektu, jako množinu několika jednodušších úkolů. Musí poskytovat možnost sledování realizace jednotlivých částí projektu a její řízení. Pro zajištění úspěšné realizace je nezbytné, aby systém poskytoval možnost dostatečně popsat zadání projektu nebo jeho částí, vytvoření určitého plánu pro splnění zadání a závěrečné zhodnocení celé realizace nebo jejích samostatné části. Při plánování musí aplikace přehledně definovat pět základních faktů a to, jaký úkol má být realizován, jaké technologie mají být použity, kdo má daný úkol vypracovat a kolik finančních prostředků bude realizace stát.

4 Vývojové technologie

Primárním cílem této kapitoly je stručné seznámení s vývojovými technologiemi použitých při realizaci této práce. V kapitole 2.1 Cíle práce jsem stanovil, že výsledným produktem má být jednoduchá a snadno ovladatelná webová aplikace. Tudíž technologie, které zde představím, zaměřené právě na tvorbu webů. Toto stručné seznámení s vývojovými technologiemi určené, pro vytvoření určité představy struktury této práce. Zaměřím se proto na stěžejní programovací jazyky, které byly použity k vytvoření tohoto systému.

4.1 HTML/XHTML

Je základním kamenem webových stránek. Soustředí se na strukturu zobrazovaných dat. Neobsahuje žádnou dynamiku stránek, ani se nestará o vzhled zobrazených údajů. Je to základní jazyk mezi prohlížečem a serverem.

HTML

HyperText Markup Language, označovaný zkratkou HTML, je značkový jazyk pro hypertext. Je jedním z jazyků pro vytváření stránek v systému World Wide Web, který umožňuje publikaci dokumentů na Internetu.[3]

První definici HTML vytvořil Tim Berners-Lee v roce 1991. Tato verze umožňovala vkládat do textu obrázky, hypertextové odkazy, vytvořit několik logických úrovní a několik druhů zvýraznění. Byla označena jako *HTML 0.9*. Požadavky uživatelů se zvyšovaly a producenti prohlížečů začali vytvářet HTML nové prvky. Tim Berners-Lee všechny používané prvky shrnul do standardu *HTML 2.0* (specifikace v RFC 1866), který již plně vyhovuje standardu SGML (ISO 8879, r. 1986). V roce 1996 vzniklo *HTML 3.0*. V roce 1997 *HTML 4.0* a o dva roky později opravná verze *HTML 4.01*, která je také poslední vývojovou verzí HTML.²

```
<html>
  <!-- toto je komentář -->
  <head>
    <title>Titulek stránky</title>
  </head>
  <!-- tělo dokumentu -->
```

² Hlavenka, Jiří. *Vytváříme WWW stránky*. Praha : Computer Press, 2001. 437 s.

```
<body>
  <h1>Nadpis stránky</h1>
  <p>Toto je tělo dokumentu</p>
</body>
</html>
```

HTML je tedy nepatentovaný formát založený na SGML. SGML (Standard Generalized Markup Language) je univerzální značkovací metajazyk, který umožňuje definovat značkovací jazyky jako své vlastní podmnožiny. SGML je komplexní jazyk, který poskytuje mnoho značkovacích syntaxí, ale jeho složitost brání většímu rozšíření. HTML využívá tagy ke strukturování textu do nadpisů, odstavců atd. Interpretem HTML tagů je tedy internetový prohlížeč. Jenže internetový prohlížeč neexistuje jenom jeden jediný. Je jich hned několik druhů a každý vývojář do svého internetového prohlížeče může vložit i svoje značky "navíc". Aby se zabránilo různému zobrazování stránek v jednotlivých prohlížečích existuje konsorcium W3C, která tyto tagy schvaluje. Tyto schválené tagy by se měli zobrazovat ve všech prohlížečích správně a hlavně stejně. Vývoj HTML sice již skončil, ale HTML má samozřejmě své následníky.[6]

XHTML

XHTML je právě tím následovníkem, někdy je označován za novou vylepšenou verzi HTML. XHTML není revolučně novým jazyk, který by se kodér musel od začátku učit. V zásadě se jedná o HTML, ovšem s přísnějšími pravidly.[3] Jedná se o striktnější jazyk než HTML, sám již zahrnuje pravidla zajišťující správné fungování stránky (například prvek Doctype není v HTML nezbytně nutný, ovšem v XHTML musí být vždy). Do vývoje toho to jazyka proudí značné úsilí. Aby však byl soubor XHTML kompatibilní s XML je nezbytné dodržet několik zásadních pravidel:

- ✓ Stránku je nutné začínat deklarací typu dokumentu. Jedná se o již zmiňovaný prvek Doctype. Deklarace typu dokumentu je nejdůležitější součástí stránky XHTML.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

- ✓ Značky a jejich atributy musejí být zadány malými písmeny. Na rozdíl od HTML nelze použít například zápis <BODY>.
- ✓ V attributech značek musejí být uvozovky. Kupříkladu odkaz zadaný jako je v kódu HTML platný, ovšem v XHTML nebude fungovat. Hodnotu vlastnosti je třeba uzavřít do uvozovek.
- ✓ Všechny značky, dokonce prázdné, musejí být uzavřeny. Odstavec <p> vytvořený v XHTML musí být ukončený značkou </p>. Trochu problém je v tom, že některé značky nejsou párové. Takové prázdné značky nemají uzavírací podobu. Příkladem je značka ukončení řádku. Chceme-li uzavřít prázdnou značku, je nutné vložit lomítko na její konec tímto způsobem:
.

Proč tedy o XHTML vůbec uvažovat a proč se ho učit? Stojí to za to? Většina expertů se shoduje, že XHTML patří budoucnost. Důvodů je několik, mj.:

- ✓ Díky přísným a zároveň jednoduchým pravidlům, mohou počítače XHTML velmi snadno automatizovaně zpracovávat. Kdyby prohlížeči stačilo "umět" XHTML, byl by mnohem jednodušší (a tedy menší a rychlejší), než když musí zvládat veškeré "nevyzpytatelnosti" HTML.
- ✓ Dá se očekávat, že právě díky vyšší srozumitelnosti počítačům, budou časem stránky vytvořené v XHTML "oblíbenější" u vyhledávačů, katalogů stránek, výměnných reklamních systémů a dalších automatizovaných služeb. Dříve nebo později začnou prohlížeče podporovat pouze XHTML a neuškodí, budete-li na to připraveni.
- ✓ První specifikace, jejíž cílem bylo převedení staršího jazyka HTML tak, aby vyhovoval podmínkám tvorby XML dokumentů a přitom byla zachována zpětná kompatibilita.

Rozlišujeme 3 druhy XHTML:

- ✓ XHTML 1.0 Strict je čistě strukturální značkování, neobsahuje žádné značky spojené s formátováním vzhledu.

- ✓ XHTML 1.0 Transitional povoluje atributy pro formátování textu a odkazů v elementu body a některé další atributy.
- ✓ XHTML 1.0 Frameset používá se při použití rámců pro rozdělení okna prohlížeče na dvě nebo více částí.[7]

4.2 CSS

Kaskádové šablony stylů (Cascading style sheets – zkráceně CSS) umožňují tvořivě ovládat rozvržení a návrh webových stránek. Před příchodem CSS museli weboví návrháři používat omezené možnosti rozvržení a zavádění stylů podporované v HTML.³

Jazyk HTML stále představuje základ webových stránek na síti World Wide Web, ovšem v žádném případě se nejedná o nástroj designu. Je samozřejmé, že HTML zajišťuje základní formátování prvky pro text, obrázky, tabulky a další elementy webových stránek a že pečliví weboví tvůrci mohou dosáhnout dobrého vzhledu stránek jen pomocí HTML. Výsledkem jsou však pomalé webové stránky s nepředvídatelným chováním a se spoustou ošklivého kódu. Styly CSS naproti tomu nabízejí následující výhody:

- ✓ Šablony stylů nabízejí mnohem více formátovacích možností než HTML. Pomocí CSS formátovat lze odstavce stejně jako v knihách nebo v časopisech (například kdy je první řádek odsazený a mezi odstavci nejsou žádné mezery) a řídit prokládání (mezeru mezi řádky v odstavci).[4]
- ✓ Když je použito k přidání obrázku na poradí stránky CSS, je možné určit, jak a zda vůbec se dlaždicově poskládá. HTML něco podobného vůbec nezvládne.
- ✓ Styly CSS zabírají mnohem méně místa než formátovací prvky jazyka HTML, jako je například ona neslavná značka . Po aplikování stylu CSS lze obvykle převážně textové webové stránky zmenšit o mnoho kilobajtů. V konečném důsledku vytvořené stránky vypadají výborně a navíc se rychleji zavádějí.
- ✓ Šablony stylů rovněž zjednodušují aktualizaci sídla. Všechny použité styly je možné shrnout do jediné externí šablony stylů navázané na všechny stránky síd-

³ <http://www.jakpsatweb.cz/css/>

la. Když pak bude nějaký styl upraven, projde tato změna okamžitě celým sídlem, ať už se daný styl objevuje kdekoli. Vzhled sídla tak lze úplně změnit prostou úpravou jediné šablony stylů.⁴

Příklad přímého stylu `<p style="color: red;">text červeného odstavce</p>`

Příklad stylopisu `<style> p {color: red} body {background-color: yellow;} </style>`

Příklad jednoduchého zápisu vlastností `color: red` a složitějšího zápisu vlastností `font-family: Arial, Arial CE, sans-serif; color: red;`

4.3 PHP

PHP je hypertextový preprocesor, který na serveru interpretuje stránky HTML s vlastními příkazy před jejich odesláním ke klientovi. Klientem obvykle bývá webový prohlížeč. To znamená že PHP umožňuje vkládat vlastní krátké úseky kódu ale i celé programy přímo do hypertextových stránek.[1]

```
<?php
/*
víceřádkový
komentář
*/

echo ("vypíše text <br />");

// jednořádkový komentář
?>
```

Na rozdíl od jazyků interpretovaných teprve na straně klienta, tedy až ve webovém prohlížeči, jako je například JavaScript, je PHP interpretováno na serveru, což přináší především tyto výhody:

- ✓ Snadná interakce s dalšími aplikacemi na serveru, není třeba přenášet data po internetu, snáze se zajišťuje bezpečnost celého systému.

⁴ Sawyer Mcfarland, David. *CSS chybějící manuál*. Karel Voráček. Praha : Grada Publishing a.s., 2006. 432 s.

- ✓ Nenáročnost na hardware či software klienta, jelikož výstupem je obvykle čistý HTML kód, který dokáže interpretovat i starší prohlížeč. Není potřeba, aby na klientovy byly prováděny skripty či rozsáhlé výpočty, nebo aby podporoval zásuvné moduly atd.
- ✓ Menší objem přenesených dat, protože serverové skripty, jenž jsou do kódu vloženy, jsou ještě na serveru interpretovány a na výstupu se již neobjeví, zatímco skripty určené pro klienta ano.
- ✓ Výrazně vyšší ochrana zdrojových textů programů, díky interpretaci kódu na serveru se klient nemůže dostat ke zdrojovým textům, zatímco v případě klientských technologií jsou mu zdrojové texty přímo zaslány.

Naopak provádění kódu na straně klienta má také několik výhod:

- ✓ Menší zátěž serveru. Server pouze dostane požadavek a odešle stránku, zatímco v případě serverových technologií musí ještě spustit interpret jazyka, předat mu stránku a klientovi zasílá až výsledek. PHP může běžet také jako modul webového serveru, čímž se výrazně snižuje reakční doba i paměťové nároky, takže vliv interpretace PHP kódu na rychlost odezvy serveru je minimální.
- ✓ Možnost interakce s objekty na webové stránce. Klientské technologie vznikají především z tohoto důvodu. Jen u klienta je možné reagovat na stisk kláves, pohyb či klepnutí myší atd.

Z výše uvedených výhod a nevýhod PHP jasně vyplývá, kdy a které technologie použít. Zatímco pro manipulaci s objekty uvnitř webové stránky je vhodné použít nějakou technologii interpretující se na straně klienta, například JavaScript, PHP se velmi dobře hodí pro spolupráci s databázemi, zpracování formulářů a náročnější úlohy jako manipulace s grafikou nebo se soubory PDF.

5 Analýza a návrh řešení

5.1 Základní charakteristika aplikace

Primárním úkolem aplikace Maxim je umožnit a co možná nejvíce zjednodušit správu podnikových projektů. Je zaměřena zejména na rozsáhlejší projekty, kdy jeden zaměstnanec realizaci nemůže stihnout ve stanovené čase a je nutná spolupráce více zaměstnanců. Aplikace tento problém řeší rozložení jednoho složitého projektu na více jednodušších úkolů a následné rozdělení úkolů mezi jednotlivé zaměstnance. Lze ji používat i jako jednoduchý informační systém spojující výhody intranetu (komunikace uvnitř firmy) a extranetu komunikace s externími subjekty. Aplikace Maxim je prezentována na síti jako standardní hostovaná aplikace.

Datová základna aplikace je tvořena databází **MySQL**, která musí být nebytně spuštěna na serveru. Pro aplikaci je primární zdroj dat, kterými jsou definovány jednotlivé projekty a jejich elementární úkoly, zákazníci, profily zaměstnanců a interní logika systému.

V databázi jsou především soustředěny všechny základní údaje týkající se podnikových činností a konfigurační tabulky aplikace, pomocí kterých se nastavují parametry podnikových závazcích. V aplikaci jsou i informace pro základní personální evidenci. Již zavedení zaměstnance do systému vede uživatele formou implicitních nastavení tak, aby nebylo možné dopustit se zásadní personální chyby, jako například přiřazení úkolu na který daný zaměstnanec nemá dostatečné vědomosti nebo dovednosti. Neumožňujeme zpracovávat pro jednoho zaměstnance souběžně více pracovních činností. Současně s výběrem druhu činnosti uživatel automaticky nastavuje i jednoznačné algoritmy pro zpracování tohoto vztahu. Pracovní doba není nijak parametrizována standardními rozvrhy, ale je možno vytvářet i naprosto individuální rozvrh pracovních činností.

5.2 Adresářová struktura aplikace

Adresářovou strukturu názorně představuje obrázek Obr. 3. Všechny základní skripty a informace nezbytné pro správný chod aplikace jsou uloženy v několika strukturovaných adresářích. Z nichž každý obsahuje zdrojové texty nebo důležité informace určitého dru-

hu. Bylo velmi nepřehledné, kdyby všechny zdrojové soubory , dokumenty i výstupy aplikace nebyly dostatečně tříděny nebo dokonce uloženy všechny v jednom adresáři.



Obr. 2 Adresářová struktura

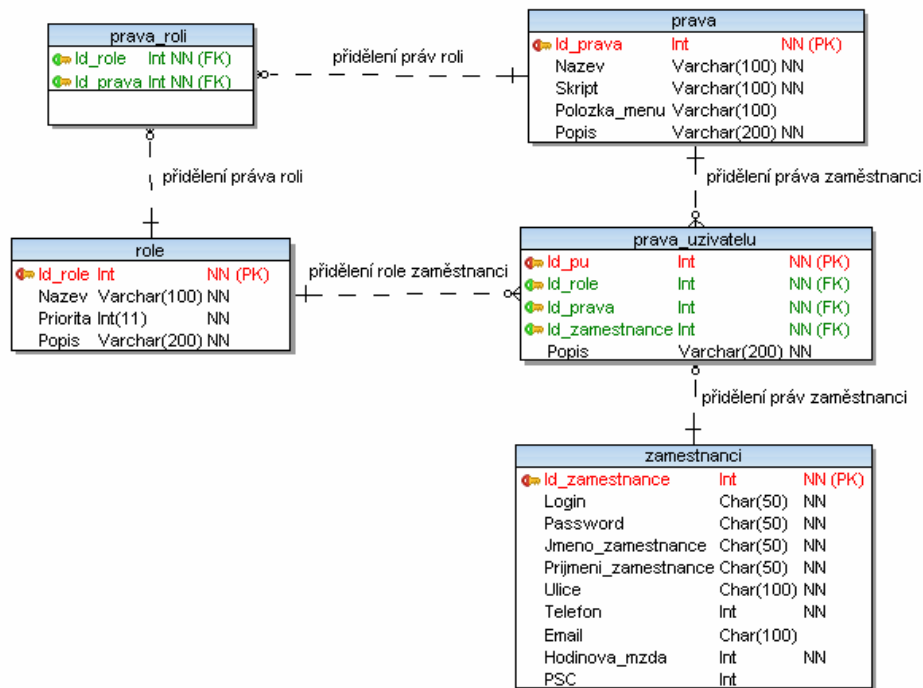
Popis a účel podadresářů

Definice objektů, které aplikace hojně využívá, jsou uloženy v podadresáři *classes*. Tyto objekty definují reálné objekty v systému jakými jsou například projekt, uživatel, úkol nebo objekt zabezpečující komunikace z databází. *CSS* je podsložka pro externí soubory kaskádových stylů, které definují vzhled jednotlivých částí aplikace, ať už to je menu, formulář tabulka či obrázek. Skripty, které se starají o jejich zobrazení a obsluhu jsou uloženy v podadresáři *form*. Složka *images* obsahuje, jak už její název napovídá, všechny obrázky, které jsou v aplikaci použité, převážně jsou využívány jako součásti designu. Složky *menu* a *include* jsou ve velmi úzkém vztahu. V *menu* se nacházejí jmenné seznamy možností aplikace pro každou zavedenou roli v systému. Každá taková položka ve jmenném seznamu role odkazuje na příslušný skript ve složce *include*, který obsluhuje danou možnost aplikace vybranou z *menu*. Všechny pomocné skripty, většinou vytvořené v jazyce JavaScript, jsou umístěny ve složce *scripts*. Poslední adresát *temp* slouží jako odkládací místo pro dočasné soubory.

5.3 Práva v aplikaci

Základním kamenem celého systému oprávnění je tabulka „prava“, ta shromažďuje všechny práva, která jsou v aplikaci definována. Z skupiny těchto práv se tvoří modifikovatelné role v aplikaci. Množiny oprávnění jednotlivých rolí jsou uložena v tabulce „prava_rolí“. Dále, zde existuje možnost udělovat individuální práva libovolným uživatelům aplikace. Uživateli může být přiděleno více rolí, oprávnění přidělených rolí se

sloučí. Záznamy o přidělených rolích a individuálních právech jsou uloženy v tabulce „prava_uzivatele“. Níže uvedený dílčí ER diagram znázorňuje vazby jednotlivých tabulek figurující v systému oprávnění.



Obr. 3 Dílčí ER diagram systému oprávnění

5.4 UML diagramy

Níže uvedené UML aktivní diagramy velice přehledně zachycují chování aplikace u předdefinovaných rolí v jejich jednotlivých stavech, ve formě sekvence činností řízených převážně interními událostmi aplikace.

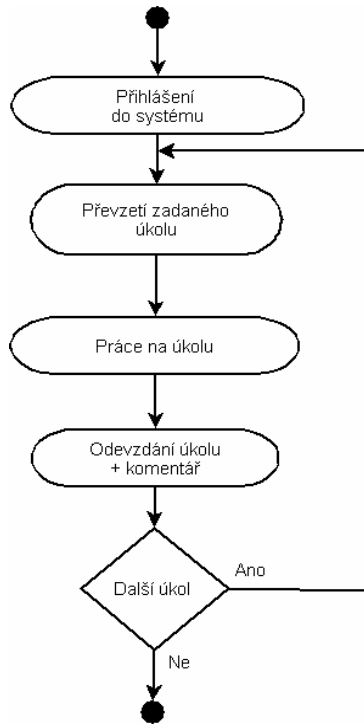
Rozhodovacími bloky diagram znázorňuje implementaci interní logiky systému.

V rozhodovacích procesech mají důležitou roli, jak data zadaná uživatelem, tak konfigurační údaje aplikace vložené administrátorem. Na diagramu aktivit jsou zřetelně znázorněny činnosti interní logiky aplikace při zakládání projektu a definování jeho úkolů.

5.4.1 Zjednodušené aktivní diagram

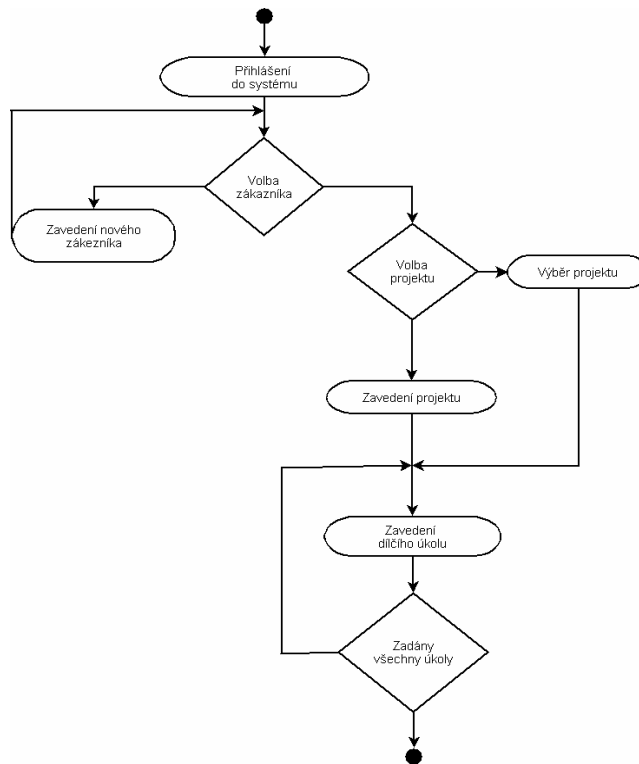
Aktivní diagram pro programátorskou roli je velice jednoduchý. Řadový pracovník, v tomto případě programátor, nemá žádné možnosti, jak zasahovat do dat, se kterými

aplikace pracuje. Může vykonávat pouze velice omezené množství úkonů, jak ukazuje Obr. 4.

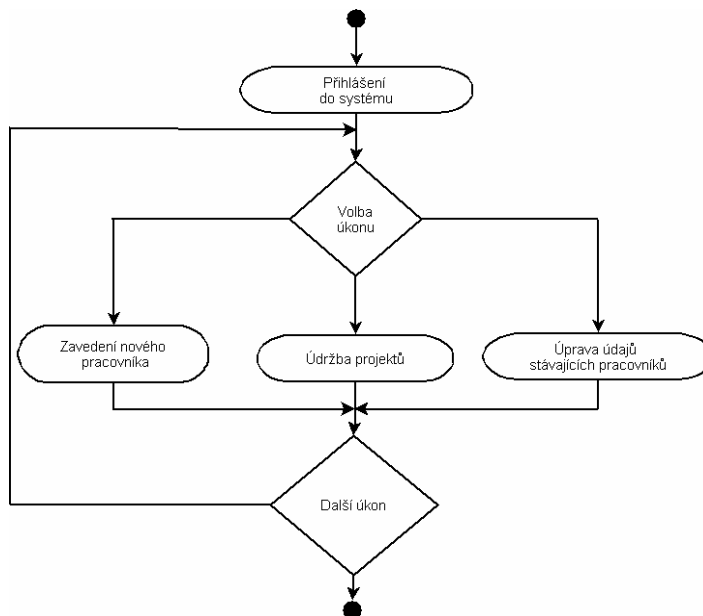


Obr. 4 Aktivitní diagram role programátora

Role administrátora a manažera již mají přístupné funkcionality pro vkládání a modifikování dat zavedených projektů a v případě administrátora i možnosti nastavovat parametry aplikace a tím ovlivnit její chování. Zatím co manažer má oprávnění manipulovat s daty, které aplikace spravuje (definování zákazníků, projektů, úkolů, ...) jak je znázorněno na obrázku Obr. 5. Hlavní funkcí administrátora je spravovat aplikaci z hlediska přístupů, oprávnění uživatelů a správa nastavení uživatelských účtů (hodnocení, mzdy, dovednosti, ...). Aktivitní diagram této role jsou znázorněny na obrázku Obr. 6



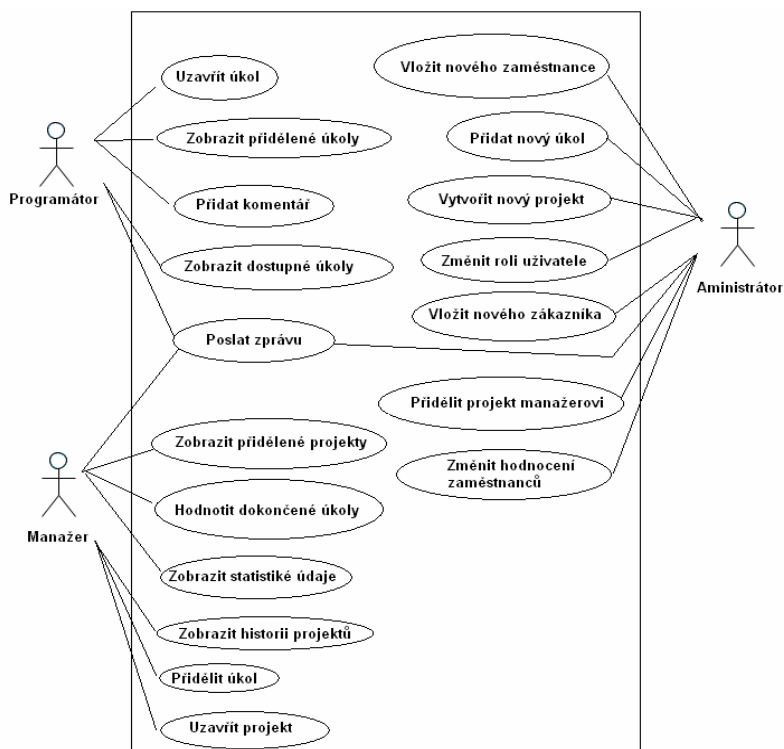
Obr. 5 Aktivitní diagram role manažera



Obr. 6 Aktivitní diagram role administrátora

5.4.2 UML use case diagram

Uživatelské role jsou důležitým článkem hierarchického uspořádání uživatelů aplikace. Jednotlivé role, zavedené v systém a jejich práva, jsou názorně zobrazena na obrázku Obr. 7.



Obr. 7 use case diagram

5.5 Návrh databáze aplikace

Datovou základnu celého systému tvoří databáze. Pro správný chod aplikace je nezbytně nutné, aby měla k dispozici spuštěnou databázi na serveru. Aplikace je navržena pro databázi MySQL. Struktura databáze je znázorněna na obrázku Obr. 4 Struktura databáze.

MySQL je zkratka z anglického názvu My Structured Query Language, což by se dalo volně přeložit jako *systém pro řízení databází*. MySQL je relační databáze typu DBMS (database management system) a vychází z deklarativního programovacího jazyka SQL (Structured Query Language). Je šířen jako Open Source.

Důvodů, proč jsem zvolil zrovna tuto databázi je hned několik. Díky své licenci a rychlosti je velmi rozšířeným a v poslední době velmi oblíbeným systémem. MySQL je ma-

Struktura a funkce tabulek

Zamestnanci

```
Id_zamestnance Int PRIMARY KEY AUTO_INCREMENT,  
Login Varchar(50) NOT NULL,  
Password Varchar(50) NOT NULL,  
Jmeno_zamestnance Varchar(50) NOT NULL,  
Prijmeni_zamestnance Varchar(100) NOT NULL,  
Mesto Char(100) NOT NULL,  
Ulice Char(100) NOT NULL,  
PSC Int NOT NULL,  
Telefon Int NOT NULL,  
Email Varchar(100),  
Hodinova_mzda Int NOT NULL  
Hodnoceni Int NOT NULL
```

Tabulka zaměstnanci neshromažďuje pouze osobní data o jednotlivých zaměstnancích, jako jsou jméno, příjmení, atd., ale slouží také k Autentizaci uživatelů, kteří se přihlašují do systému. Tuto schopnost zajišťují sloupce Login a Password. K osobním datům mimo hodinové mzdy a hodnocení, mají přístup pouze osoby, jejichž se přímo týkají a administrátoři, kteří mají navíc možnost měnit hodnoty ve všech sloupcích bez výjimky.

Další důležitou částí tabulky je sloupec Hodnoceni. Toto číslo má inicializační hodnotu 1 a limitující hranici 10. Číslo ve sloupci hodnocení odráží schopnosti a zkušenosti zaměstnance. Velikost bodového ohodnocení každého zaměstnance může měnit pouze administrátor aplikace. Tyto body se také projevují ve výši platového ohodnocení. Jsou jakousi směrnicí ke konkrétní výši platu.

Zakaznici

```
Id_zakaznika Int PRIMARY KEY AUTO_INCREMENT,  
Jmeno_zakaznika Varchar(50) NOT NULL,  
Prijmeni_zakaznika Varchar (50) NOT NULL,  
Mesto Varchar (100) NOT NULL,  
Ulice Varchar (100) NOT NULL,  
Telefon Int NOT NULL,  
Email Char(100)
```

Tato tabulka uchovává informace o zákaznících, se kterými má, nebo měl podnik nějaký obchodní vztah. Pro vytvoření projektu je nutný aby zákazník byl již definován v této tabulce. Jsou zde uvedeny také kontaktní informace pro řešení případných problémů. Jednoznačným identifikátorem zákazníka je sloupec Id_zakaznika.

Projekty

```
Id_projektu Int PRIMARY KEY AUTO_INCREMENT,  
Id_zakaznika Int,  
Nazev_projektu Varchar(100) NOT NULL,  
Cena_projektu Varchar(100) NOT NULL,  
Datum_zadani Date NOT NULL,  
Datum_ukonceni Date
```

Tabulka projektů má úzký vztah s tabulkou úkolů, protože každý projekt je složen z libovolného počtu úkolů. V tabulce projektů jsou uloženy hlavičky jednotlivých zadaných projektů. Každá hlavička nese souhrnné informace projektu. Id_projektu je unikátní identifikátor každého záznamu. Id_zakaznika cizí klíč z tabulky zákazníků, tvoří zpětnou vazbu na zadavatele projektu. Celkovou cenu projektu tvoří součet cen všech úkolů, které k danému projektu patří. Datum_ukonceni obsahuje datum, do kdy musí být všechny úkoly splněny.

Ukoly

```
Id_ukolu Int PRIMARY KEY AUTO_INCREMENT,  
Id_projektu Int,  
Id_zamestnance Int,  
Jmeno Varchar(100) NOT NULL,  
Popis_ukolu Varchar(1000),  
Cena_ukolu Int NOT NULL,  
Datum_zadani Date NOT NULL,  
Datum_dokonceni Date,  
Termin_dokonceni Date NOT NULL
```

Každý řádek v této tabulce definuje jeden úkol projektu, vazbu na daný projekt, zabezpečuje cizí klíč z tabulky projektů Id_projektu. Druhým cizím klíčem je Id_zamestnance. Ten představuje vazbu na tabulku zaměstnanců, slouží k přiřazování úkolů jednotlivým programátorům. Další důležitým prvkem je Termin_dokonceni, tímto stanovuje zákazník do kdy má být úkol hotový.

Dovednosti

```
Id_dovednosti Int PRIMARY KEY AUTO_INCREMENT,  
Nazev_dovednosti Varchar(100) NOT NULL
```

Tabulka Dovednosti shromažďuje všechny definice dovedností a technologií, které zaměstnanci ovládají a které může podnik nabídnout k realizaci svých projektů. Jednoznačným identifikátorem dovednosti je Id_dovednosti. Do této tabulky smí zapisovat pouze administrátor aplikace.

Cinnosti

```
Id_cinnosti Int PRIMARY KEY AUTO_INCREMENT,  
Id_zamestnance Int,  
Id_ukolu Int,  
Zacatek Datetime NOT NULL,  
Konec Datetime NOT NULL,  
Komentar Varchar(1000)
```

Pravděpodobně nejpoužívanější tabulka v databázovém systému. Do tabulky činností se zapisují všechny vstupy do systému a aktivity všech zaměstnanců nad jednotlivými úkoly. Záznam se vytvoří v momentě, kdy se programátor přihlásí k nějakému projektu. Zaznamená se čas, kdy činnost začala a čas, kdy došlo k odhlášení nebo ukončení úkolu. Mzdy každého zaměstnance se odvíjí od odpracovaných hodin, konečná mzda zaměstnance je vypočítávána na základě záznamů v této tabulce.

Dovaednosti_zamestnance

```
Id_dovednosti Int,  
Id_zamestnance Int
```

Dovaednosti_zamestnance obsahuje dva cizí klíče z tabulek dovedností a zaměstnanců, obsahuje informace o tom, jaké dovednosti ovládají jednotliví zaměstnanci. Přidávat záznamy do této tabulky má právo pouze administrátor. Odebírat může libovolný uživatel, ovšem pouze záznamy, které se ho bezprostředně týkají.

Prava_rolí

```
Fk_role Int NOT NULL,  
Fk_prava Int NOT NULL
```

Tabulka Prava_rolí páruje záznamy s tabulky Prava a Role. Každý záznam v této tabulce přiřazuje jedné roli právě jedno právo. Práva k této tabulce by měl mít pouze administrátor.

Prava

```
Id_prava int PRIMARY KEY AUTO_INCREMENT,  
Nazev Varchar(100) NOT NULL,  
Skript Varchar(100) NOT NULL,  
Polozka_menu Varchar(100) NOT NULL,  
Popis Varchar(100) NOT NULL
```

Tato tabulka obsahuje informace o všech oprávněních, jaké jsou v aplikaci definovány. Tato tabulka tvoří strukturu systému práv uživatelů v aplikaci..

Role

```
Id_role PRIMARY KEY AUTO_INCREMENT,  
Nazev Varchar(100) NOT NULL,  
Priorita Int(11) NOT NULL,  
Popis Varchar(200) NOT NULL
```

Tabulka rolí nese záznamy o jednotlivých rolích, které jsou v systému zavedeny. Práva k této tabulce by opět měl mít pouze administrátor.

Prava_uzivatelu

```
Id_pu Int PRIMARY KEY AUTO_INCREMENT,  
Id_role Int NOT NULL,  
Id_prava Int NOT NULL,  
Id_zamestnance Int NOT NULL,  
Popis Varchar(200) NOT NULL
```

V této tabulce se páruje primární klíč tabulky uživatelů s primárním klíčem role, jedná-li se o přidělení role uživateli, nebo s primárním klíčem oprávnění, jedná-li se o přidělení individuálního oprávnění. Práva k této tabulce by měl mít pouze administrátor.

6 Instalace aplikace

Jelikož se jedná o webovou aplikaci, nepotřebuje nijak složitou instalaci. Prakticky se jedná pouze o uložení celé aplikace na webový server do vámi zvoleného veřejného adresáře. Dále je nutné upravit několik konstant ve zdrojovém kódu. V adresáři `inc` najdeme soubor `config.php` zde je třeba změnit ty konstanty:

- ✓ Host na adresu k databázi,
- ✓ Log zde uložíme login uživatele k databázi,
- ✓ Pass hodnotu změníme na heslo uživatele databáze.
- ✓ `defaultDatabase` na název databáze, kterou bude aplikace používat

Administrátor serveru musí povolit zápis do složky `temp`, `dokuments` a `export`. Pro zavedení databáze je třeba mít na serveru na instalovanou databázi MySQL a importovat data, která jsou exportována do souboru `Databáze.sql`, jenž se nachází v adresáři `Databáze`. Po těchto změnách, bude aplikace plně funkční a připravena k používání.

7 Uživatelská dokumentace

K aplikaci se přistupuje pomocí webového prohlížeče. Po zadání příslušné webové adresy je uživateli zobrazena úvodní stránka aplikace. Na této stránce je vyzván, aby se přihlásil. Přihlašovací dialog se nachází vpravo nahoře, hned pod hlavičkou aplikace.



Obr. 9 Přihlašovací dialog

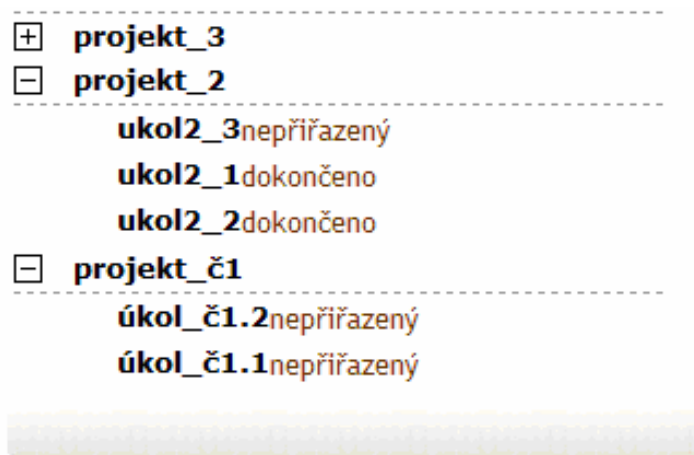
Po zadání přihlašovacích údajů aplikace ověří, zda má uživatel přístup do aplikace. Pro umožnění přístupu musí, být uživatel zaveden administrátorem a jsou mu přiděleny přihlašovací údaje. Pokud přihlášení proběhlo v pořádku, tj. byly zadány správné údaje, přihlašovací dialog zmizí a je nahrazen celým jménem a názvem role uživatele. Dále je zobrazeno menu, které obsahuje možnosti aplikace, ke kterým má uživatel přístup. To, jaké možnosti se bude moci využívat, závisí na roli, která mu byla přidělena.

Práva v aplikaci jsou řešena pomocí modifikovatelných rolí a individuálních práv. Definovat nové, či upravovat již zavedené role, smí pouze administrátor, nebo uživatel jemuž bylo přiděleno individuální oprávnění pro manipulaci s právy uživatelů. Tímto systémem jsou v aplikaci předdefinované tři základní role, které jsou popsány níže.

7.1 Programátor

Tato role je určena pro zaměstnance, kteří nemají žádnou vedoucí pozici v podniku. Neumožňuje nijak významně zasahovat do dat a chování aplikace. Po přihlášení jsou programátorovi nabídnuty úkoly, které mu byly přiděleny vedoucím projektu. Pokud nemá zadáný žádný úkol, může si vybrat některý z volných. Programátor má přístup pouze

k úkolům projektů, který odpovídají jeho dovednostem. Úkoly, pro které je zapotřebí jiných dovedností a schopností mu aplikace nezobrazí.



Obr. 10 Výpis projektů pro programátora

Po přihlášení ke zvolenému úkolu, může začít s vypracováním jeho náplně. Od momentu zvolení úkolu se zaměstnanci zaznamenává čas strávený na vybrané úloze. Pokud má programátor úkol vypracovaný, označí jej jako dokončení a může začít pracovat na jiném. Ke své činnosti může přidat libovolný komentář, kterým by mohl vyzdvihnout hlavní charakteristické body své činnosti, nebo například zdůvodnit, proč mu úkol trval méně, nebo naopak více času, než bylo při zadávání odhadnuto administrátorem. Programátor má také přístup ke svým osobním údajům, se kterými aplikace pracuje, jako jsou přihlašovací údaje, jméno, příjmení, adresa, atd.. Tyto údaje může libovolně měnit. Má rovněž možnost nahlédnout na seznam dovedností, které má zavedené v systému a podle kterých se mu nabízejí úkoly. Je mu dovoleno se nějaké své dovednosti vzdát, ale o přidání nějaké dovednosti si musí požádat administrátora systému.

7.2 Manažer

Role manažera je z pohledu programu mnohem rozmanitější. Tato role má rozhodovací možnosti, které předchozí role postrádá. Po přihlášení do role manažera se nezobrazí přidělené úkoly, jako tomu bylo u programátora, nýbrž celé projekty, které bude mít daný manažer na starosti a zajistí jeho včasné vypracování. Výpis přidělených projektů zobrazuje obrázek Obr. 11 Výpis projektů manažera.



Obr. 11 Výpis projektů manažera

K obyčejnému výpisu přibyla možnost přidělení daného úkolu nějakému programátorovi. Po zvolení volby „přiřadit“ se dostaneme do karty zvoleného úkolu, aplikace vybere programátory, kteří mají potřebnou dovednost pro vypracování dané úlohy. Karty úkolu je zobrazena na obrázku Obr. 12 Karta úkolu.

Úkol

Název úkolu :

Cena úkolu

termín dokončení

popis

Zaměstnanec :

Výber vhodných zaměstnanců

Zaměstnanec	Body hodnocení	Nejbližší termín nedokončeného úkolu
Martin Raur	1	2008-08-12
Jan Prchal	1	2008-08-20
Petr Dipold	1	2008-08-20

Obr. 12 Karta úkolu

Manažerovi může usnadnit výběr vhodného zaměstnance zobrazená tabulka vhodných programátorů včetně jejich bodového ohodnocení a pokud zaměstnanec má již přidělený určitý počet úkolů, které ještě nejsou dokončeny, tak i nejbližší termín nedokončeného úkolu. Poskytuje manažerovi přehled o tom, který programátor je pro tento úkol nejvhodnější z hlediska jeho času a kvality.

U úkolů svých přidělených projektů má možnost napsat posudek, na jejichž základě se může administrátor rozhodnout, jak o výši jeho bodového hodnocení, tak o velikosti hodinové mzdy. Manažer má dále přístup ke všem informacím o zaměstnancích i k jejich osobním kartám. Obrázek Obr. 13 zobrazuje výpis uživatelů, kliknutím na jeho jméno může zobrazit jeho osobní kartu. Zde může v případě nutnosti uživateli změnit osobní či přihlašovací údaje. V této roli je také možno přistupovat ke všem statistickým údajům, které aplikace nabízí.

Zaměstnanec	Dovednosti	Mzda	Hodnocení
Martin Raur	php,java mobile	2	1/10
Jan Prchal	konfigurace serveru,java aplet	1	1/10
Petr Dipold	konfigurace serveru,java aplet	180	1/10
Martin Mazuch		222	1/10
Radek Myslivec		211	1/10
Ivo Farka	java aplet,java serverlet	250	1/10
Petr Hlava	javascript,php,konfigurace serveru,java serverlet,java serverlet	199	1/10
Jan Havel	php	210	2/10

Obr. 13 Výpis uživatelů

7.3 Administrátor

Administrátor je nejvyšší role jaká je v aplikaci zavedena. Jenom administrátor má právo zakládat nové projekty. Při tvorbě projektu musí rozdělit projekt, jako celek na menší úkoly. Každý úkol do systému zadat a v několika bodech ho charakterizovat do pole popis, určit cenu za jeho zpracování, odhadnout dobu potřebnou na jeho správné vypracování. Formulář pro vkládání úkolu k projektu je na obrázku Obr. 14 Nový úkol.

Aktivní projekt

Název projektu : Jméno zadavatele :

Cena projektu : termín dokončení :

Přidat k projektu nový úkol

Název úkolu : Technologie :

Cena úkolu : Termín dokončení :

Odhadovaný čas : hodin

Popis :

Obr. 14 Nový úkol

Vkládání nových zákazníků a zaměstnanců také smí provádět pouze administrátor. Stejně tak veškeré změny v ohodnocení programátorů, ať už bodové nebo finanční, jsou v povolené pouze v administrátorské roli. Definování uživatelských rolí a jejich přidělování, je umožněno také pouze administrátorovi. Další možností udělování práv má prostřednictvím individuálních oprávnění, které může přidělit libovolnému uživateli aplikace. Podmět pro změnu ohodnocení může přijít od manažera, nebo si může sám administrátor zobrazit historii úkolů libovolného zaměstnance a prohlédnout si reporty od manažerů a na jejich základě může provést změnu v hodnocení uživatelů. Administrátor může také definovat nové technologie, kterými podnik disponuje a může jimi realizovat zadané projekty svých zákazníků. Tyto technologie může poté přidělovat jako dovednosti svým zaměstnancům, pomocí formuláře na obrázku Obr. 15 Přidělení dovednosti.



The image shows a web form with two dropdown menus and a button. The first dropdown menu is labeled 'Zaměstnanec:' and contains the name 'Martin Raur'. The second dropdown menu is labeled 'přidat dovednost :' and contains the text 'php'. Below the dropdowns is a button labeled 'Potvrdit'.

Obr. 15 Přidělení dovednosti

8 Vývoj a možné rozšíření aplikace

Aplikace je navržena tak, abych případné rozšíření nevyžadovalo složitější zásahy do zdrojových kódů. Všechny moduly aplikace, které zabezpečují služby aplikace jsou uloženy v adresářích include a form. Pro přidání nějakého modulu postačí, když ho uložíme do jednoho ze jmenovaných adresářů a v adresáři menu připsáme odkaz na nový soubor do příslušného souboru se seznamem možností jednotlivých rolí.

Možnosti pro rozšíření této aplikace jsou velice široké, prakticky to závisí na konkrétních požadavcích každého podniku. Mezi ty nejdůležitější určitě patří:

- ✓ Možnost zpřístupnit aplikaci zákazníkům, a tím jim umožnit se aktivně účastnit na vypracování svého projektu.
- ✓ Přidání souborového manageru pro přehlednější správu souborů.
- ✓ Implementace systému odměn zaměstnanců, kteří dodržují termíny odevzdání vypracovaného projektu, nebo extra odměny za předčasné vypracování.
- ✓ Aplikovat atraktivnější design systému.
- ✓ Rozšíření možnosti komunikace, jak mezi jednotlivými zaměstnanci podniku, tak i mezi manažerem a zákazníkem.
- ✓ Implementace široké škály zpracování informací a získání statistických ukazatelů o činnosti podniku, které by napomáhaly při projektovém řízení.
- ✓ Přidání dokonalejších filtrů, pro snadnější rozhodování manažerů nad obsazováním přidělených úkolů.
- ✓ Modifikovatelné uživatelské účty, ve kterých by si mohl uživatel měnit vzhled aplikace.

9 Závěr

Rozsah projektového řízení, obdobně jako další manažerské obory, je velice široký a je těžké ho obsáhnout jedinou aplikací. Z tohoto důvodu jsem se soustředil hlavně na základní prvky tohoto rozsáhlého oboru. Vytvořil jsem snadno ovladatelnou webovou aplikaci, poskytující základní úkony na správ projektů, která sleduje závazky firmy po celou dobu jejich existence. Z historie projektů stanovuje statistické ukazatele, které usnadňují projektové řízení.

Aplikace není vytvořena pro jednorázové použití. V případě potřeby, je možné, jí v budoucnu jednoduše rozšířit o další funkce a zvětšit tak možnosti uplatnění tohoto systému v praxi. Díky otevřenosti systému je možno zaimplementovat další funkce vytvořené přímo na míru daného podniku, ať už by se jednalo například, rozšíření možnosti komunikace mezi pracovníky či další možnosti ve správě a sdílení souborů. Obzvlášť veliké možnosti rozšíření má manažerská role, kde by bylo příhodné implementovat víc nástrojů usnadňující důležitá řídicí rozhodnutí, například více statistických ukazatelů nebo implementace vhodnějších metod statistického zpracování informací. Lepší orientaci manažera ve větším počtu podnikových projektů by zjistilo, přidání grafického plánovacího kalendáře.

Tento systém je vhodné uplatnit všude tam, kde je nezbytné dosáhnout určité úrovně organizace lidských zdrojů a vnitropodnikové komunikace. Je primárně určen pro menších podniky, které zaměstnávají jak interní tak externí pracovníky.

10 Bibliografie

- [1] Bráza, Jiří. *CSS PHP 5 Začínáme prograovat*. Praha : Grada Publishing a.s., 2005. 244s.
- [2] Řízení projektů [online]. [cit. 2009-04-1]. URL: <<http://www.systemonline.cz/clanky/rizeni-projektu.htm> >
- [3] Hlavenka, Jiří. *Vytváříme WWW stránky*. Praha : Computer Press, 2001. 437 s.
- [4] Sawyer Mcfarland, David. *CSS chybějící manuál*. Karel Voráček. Praha : Grada Publishing a.s., 2006. 432 s.
- [5] Ullman, Larry. *PHP pokročilé Programování pro WWW*. Bartel Martin. Praha : SoftPress s.r.o, 2003. 512 s.
- [6] *Standard Generalized Markup Language* [online]. 2008 [cit. 2009-04-01]. Dostupný z WWW: <<http://cs.wikipedia.org/wiki/SGML>>.
- [7] *Co je to XHTML* [online]. 2008 [cit. 2009-04-02]. Dostupný z WWW: <<http://www.sovavsiti.cz/c01242.html>>.
- [8] *Co to vlastně HTML je, základy WWW stránek* [online]. 2008 [cit. 2009-04-02]. Dostupný z WWW: < <http://www.jirinovak.com/zobrazit.php?id=13>>.
- [9] Pokorná, Gabriela. *Projekty –jejich tvorba a řízení* . Univerzita Palackého Olomouc, 2006. 45s.
- [10] SVOZILOVÁ, Alena. *Projektový management : Systémový přístup k řízení projektu*. Praha 7 : Grada, 2006. 360 s. Expert. ISBN 80-247-1501-5.

11 Obsah příloženého CD

Aplikace – obsahuje všechny podadresáře a soubory aplikace, které jsou popsány v kapitole 5.2 Adresářová struktura.

Databáze – v této složce je uložen skript pro import databázové struktury a předdefinovaných dat.