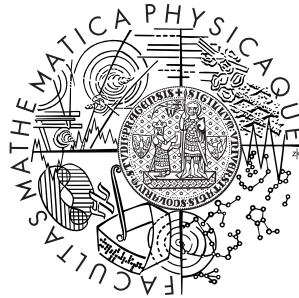


Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

BAKALÁŘSKÁ PRÁCE



Tomáš Kučera

Plánování projektů a evidence práce

Katedra softwarového inženýrství

Vedoucí bakalářské práce: RNDr. Josef Zlomek, Katedra teoretické
informatiky a matematické logiky

Studijní program: Informatika, programování

2006

Děkuji vedoucímu bakalářské práce za podnětné rady a připomínky v průběhu zpracování této práce.

Prohlašuji, že jsem svou bakalářskou práci napsal samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce a jejím zveřejňováním.

V Praze dne 9. srpna 2006

Tomáš Kučera

Obsah

1	Úvod	6
1.1	Plánování softwarového projektu	6
1.2	Trojimperativ	6
1.3	Potřeba nového SW	7
1.4	Požadované vlastnosti	8
2	Analýza problému	9
2.1	Obecný pohled na základy plánování	9
2.2	Potřeba využití softwaru	9
2.3	Využívané vlastnosti systému	10
2.4	Možnosti implementace systému	10
2.4.1	Výběr prostředí	11
2.4.2	Možnosti uchovávání dat	12
2.5	Zaměření na softwarové firmy	12
2.5.1	Zpětná vazba	12
2.5.2	Ochrana přístupových práv	13
3	Požadované vlastnosti nového programu	14
3.1	Příbuzný software na plánování	14
3.1.1	MS Project	14
3.1.2	dotProject	14
3.1.3	PHPprojekt	15
3.2	Naše požadavky	15
3.3	Specifikace požadovaných vlastností	15
3.3.1	Stromová struktura projektu	15
3.3.2	Uživatelé	16
3.3.3	Evidence práce uživatelů	16
3.3.4	Cenová nabídka úkolů	16
3.3.5	Import projektových plánů	17
4	Datové struktury	18
4.1	Projekt	18
4.2	Úkol	19
4.3	Uživatelská práce	20

4.4	Uživatel	20
4.5	Uživatelská funkce	21
5	Implementace	22
5.1	Formát systému	22
5.2	Použitý software	22
5.3	Datové struktury	23
5.3.1	Stromová struktura úkolů	23
5.3.2	Použité tabulky	24
5.4	Zajištění bezpečnosti systému	26
5.4.1	Vstupní data	26
5.4.2	Přístup k databázi	26
5.5	Bezpečný pohyb po systému	26
5.6	Uživatelské funkce	26
5.7	Přístupová práva	28
5.8	Import projektových plánů	28
5.9	Shrnutí	29
6	Pohled na vytvořenou aplikaci	30
6.1	Instalace	30
6.2	Ukázka aplikace	30
6.2.1	Zobrazení stromové struktury	30
7	Závěr	34
7.1	Možnosti rozšíření	34
	Literatura	35
	A Obsah CD-ROM	36

Název práce: Plánování projektů a evidence práce
Autor: Tomáš Kučera
Katedra (ústav): Katedra softwarového inženýrství
Vedoucí bakalářské práce: RNDr. Josef Zlomek, Katedra teoretické informatiky
a matematické logiky
e-mail vedoucího: Josef.Zlomek@mff.cuni.cz

Abstrakt: Hladký průběh vývoje softwaru se neobejde bez korektně vytvořeného projektového plánu. Tato práce se zabývá návrhem a implementací webového systému pro vytváření projektových plánů. Uživatelé systému evidují svou vykonanou práci a zaznamenávají průběh vývoje dílčích úkolů v jednotlivých projektech. Důraz je kladen na přehledné zobrazení stromové struktury projektů včetně informací týkající se průběhu vývoje projektů a jejich aktuální ceny. Systém je naprogramován ve skriptovacím jazyce PHP, data se uchovávají v databázi MySQL.

Klíčová slova: plánování projektů, webový systém, PHP

Title: Project Planning and Time Tracking
Author: Tomáš Kučera
Department: Department of Software Engineering
Supervisor: RNDr. Josef Zlomek, Department of Theoretical Computer Science
and Mathematical Logic
Supervisor's e-mail address: Josef.Zlomek@mff.cuni.cz

Abstract: Software development is much easier with a correct project plan. In our work we study design and implementation of the web system for creating project plans. Users of the system make records about their work and they enter development progress of the tasks in the projects. Emphasis is placed on clearly organized project tree structure with the information about project progress and current costs. The system is programmed in the PHP scripting language. Data are stored in the MySQL database.

Keywords: project planning, web system, PHP

Kapitola 1

Úvod

1.1 Plánování softwarového projektu

Vývoj softwarového projektu bezesporu není jednoduchou záležitostí. Je to komplexní problém. Na počátku probíhají dlouhé konzultace se zákazníkem s cílem zjistit, co zákazník od nového programu očekává, jaké vlastnosti by měl daný program mít. Poté dojde k vytvoření projektového plánu, ve kterém se definují základní úkoly, jejich dělení na dílčí podúkoly a přidělení dostupných zdrojů k daným úkolům. Podle takového projektového plánu se koordinuje programátorská činnost, jednotlivé části nového programu se řádně testují na bezproblémovou funkčnost.

Vytvoření korektního projektového plánu je nejen nezbytné pro hladký průběh vývoje, umožňuje nám již na počátku určit přibližný termín dodání nového programu a jeho odhadovanou cenu. Tyto údaje jsou pro zákazníka určující, je tedy vhodné je dobře odhadnout.

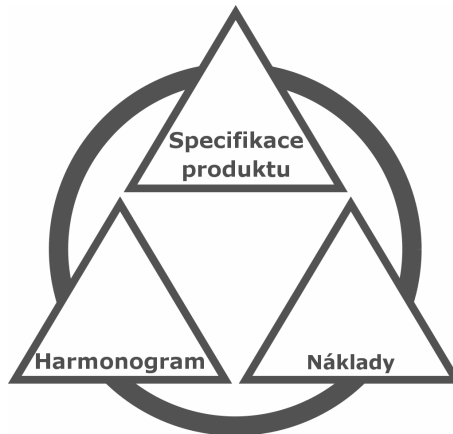
A právě softwarem umožňujícím vytváření projektových plánů se dále budu zabývat.

1.2 Trojimperativ

Projekt můžeme charakterizovat jako tzv. trojimperativ. Trojimperativ tvoří tři základní vzájemně propojené parametry projektu — specifikace produktu, časový harmonogram a náklady.

Tyto tři parametry jednoznačně definují projekt z hlediska jeho výsledku v několika rovinách:

- v rovině kvalitativní — specifikace produktu (popis produktu, rozsah díla)
- v rovině časové — harmonogram (termíny)
- v rovině finanční — náklady (náklady na práci a potřebné materiální zdroje)



Obrázek 1.1: Trojimperativ charakterizující projekt

Parametry jsou na sobě závislé, při změně jednoho parametru se automaticky změní i ostatní parametry. Například při snížení rozpočtu projektu by se měla zjednodušit specifikace projektu a tím dojde i ke změně jeho harmonogramu.

Za výsledek procesu plánování projektu lze tedy považovat určení parametrů trojimperativu. Podle těchto parametrů lze následně vyhodnotit úspěšnost projektu. Pokud projekt splňuje specifikaci svého produktu, je dokončen včas a v rámci předpokládaných nákladů, lze ho považovat za velmi úspěšný.

Více informací o trojimperativu lze získat v článku [6].

1.3 Potřeba nového SW

Na vytváření projektových plánů již nějaký software existuje, a to v různých variantách. Mohou to být desktopové aplikace, které vyžadují instalaci programu na Váš počítač, či webové služby, u kterých žádný software instalovat nemusíme, pouze využíváme prostředků jiné společnosti přes webový prohlížeč. Zajímavá varianta je webový systém, který se instaluje na server a rovněž je přístupný přes webový prohlížeč. Oproti webové službě je zde možnost instalovat systém na server dostupný pouze v lokální síti.

Dostupný software na vytváření projektových plánů umožňuje vytvářet nové projekty, definovat v nich úkoly a podúkoly a definovat zdroje¹ řešící dané úkoly. Někdy se však pozapomíná na možnost zpětné vazby zdrojů v průběhu zpracování projektu a právě na tuto vlastnost bych se chtěl zaměřit.

¹V tomto případě se jedná o uživatele (programátory, testery ...)

1.4 Požadované vlastnosti

Cílem mého snažení je vytvořit program, který by umožňoval definování projektových plánů a kde by bylo možné sledovat zpětnou vazbu zdrojů. Cílem je vytvořit systém, který bude dostupný z jakéhokoliv počítače připojeného k Internetu či k místní síti, ve kterém každý uživatel může zaznamenat svůj postup při řešení svého úkolu. Systém, ve kterém je kdykoliv a odkudkoliv možné sledovat aktuální vývoj projektu, systém, ve kterém nebude problém říci, zda-li se projekt stihne v plánovaném termínu a jaká bude jeho výsledná cena.

Kapitola 2

Analýza problému

2.1 Obecný pohled na základy plánování

Před započítím projektu je nutné určit cíle a úkoly, které musí být provedeny, aby bylo cíle dosaženo. Dalším krokem je určení, kdo bude jednotlivé úkoly provádět, kdy jednotlivé úkoly začnou a jak dlouho bude trvat jejich provedení. Navíc se musí během plánovací fáze určit, jaká bude cena projektu.

Po spuštění projektu je vhodné sledovat pokrok jednotlivých úkolů. Může se například sledovat, kdy zpracování úkolu začalo, kdy byl úkol dokončen, jaká práce byla vykonána, a zda v průběhu zpracování nenastaly nějaké problémy. V každém úkolu je pak možné provádět úpravy podle nepředpokládaných okolností, jako je například onemocnění pracovníka nebo nedodání materiálu v termínu (například při plánování stavby domu). Po úpravě plánu je nutné informovat všechny osoby, kterých se případná změna týká.

2.2 Potřeba využití softwaru

Správa projektu vyžaduje využití mnoha různých správních a koordinačních schopností. Někdy může být sledování všech aspektů projektu a udržování pokroku směrem k cíli i velmi obtížné.

Zároveň informování všech osob (přičemž každé osobě náleží pouze určitá část informací) je bez softwarové pomoci také velice pracné. Pracovníci musí vědět, co a do kdy mají udělat, management musí být informován o průběžném pokroku projektu.

Je proto vhodné využít software, který by zefektivnil plánování, správu a koordinaci projektů, a díky kterému by byl management i pracovníci průběžně informováni.

2.3 Využívané vlastnosti systému

Pokud budeme využívat systém pro plánování projektů, bude bezesporu jednou z nejdůležitějších vlastností možnost sledovat aktuální postup zpracování projektů. Je třeba informovat o možném blížícím se problému, aby ho bylo možné včas vyřešit a aby tudíž nebyl ohrožen plánovaný cíl projektu.

Data o průběžném postupu zpracování projektu vkládá do systému projektový manažer. Lze také využít potenciálu ostatních pracovníků, kterým se umožní vytvářet záznamy o vykonané práci na projektu. Poté je však potřeba vytvořit systém přístupových práv, který by rozlišoval funkce jednotlivých uživatelů.

Od systému bychom mohli požadovat také dostupnost těchto funkcí:

datový sklad S plánováním projektů nebo s jejich plněním často souvisí mnoho dokumentů či jiných souborů, které by bylo vhodné uchovávat na jednom místě (nejlépe tedy u samotného projektového plánu). Bylo by tedy možné vytvořit datový sklad, který by mohl fungovat na principu jednoduššího souborového systému.

Pokud bude systém využíván pro vývoj softwaru, tak by bylo vhodné uchovávat veškerý zdrojový kód na jednom dostupném místě. Tento problém lze jednoduše vyřešit využitím programů pro správu verzí, jako jsou CVS či Subversion. Tyto programy také umožňují zpětné získávání starších verzí zdrojového kódu.

komunikace Kvalita a způsob komunikace ovlivňuje projekt od jeho začátku až do samotného konce a má zásadní vliv na úspěšnost projektu. Je třeba zajistit hladkou a efektivní komunikaci uvnitř projektového týmu a také mezi týmem a ostatními účastníky projektu.

Komunikaci mezi uživateli lze v systému zajistit pomocí funkce jednoduchého zaslání zpráv. Jednodušší metodou je však využití emailových služeb. Pro interaktivní komunikaci lze využít chatovacích programů, jako je například ICQ.

Při návrhu systému pro plánování projektů a evidenci práce, se kterým budou pracovat mimo projektových manažerů i ostatní pracovníci, je třeba pamatovat na fakt, že systém má uživatelům pomáhat. Hlavní pracovní náplní uživatelů systému bude pravděpodobně programátorská či testovací činnost na novém softwaru. Bylo by tedy nevhodné, kdyby uživatelé museli trávit více času evidováním své práce než prací samotnou. Aplikace by tudíž měla mít intuitivní ovládání a chování.

2.4 Možnosti implementace systému

Systém je možné řešit dvěma základními způsoby, a to buď jako lokální aplikaci nebo jako aplikaci využívající síť.

lokální aplikace Aplikace je nainstalována na lokálním počítači, nevyžaduje připojení k síti. K samotné aplikaci má proto přístup malý okruh lidí, zpravidla se jedná o projektové manažery. Ti vytváří projektové plány, k jednotlivým úkolům přiřazují dostupné zdroje a evidují průběh plnění jednotlivých úkolů.

Tento typ aplikace může být vhodný například pro plánování výstavby budov nebo řízení výroby v podniku. Projektový manažer má v těchto oblastech obecně přehled o vykonané práci, kterou následně může evidovat. Ze systému získává požadavky na plnění nových úkolů či na nové zdroje a ty následně rozesílá na příslušná místa. Ostatní pracovníci, kteří plní zadané úkoly, nepotřebují přistupovat k samotné aplikaci.

síťová aplikace Aplikace je typu klient-server, tudíž vyžaduje přístup k funkční síti. Výhodou je možnost centralizovaného uchovávání dat, ke kterým je možné přistupovat odkudkoliv ze sítě.

Síťová aplikace je vhodná například pro plánování softwarových projektů. K aplikaci mají přístup jak projektoví manažeři, tak samotní pracovníci (programátoři, testeři). Pokud projektový manažer vytvoří nový úkol a zadá ho k vyřešení určitému pracovníkovi, nemusí již pracovníka informovat o novém úkolu, jelikož to systém udělá sám. Všichni pracovníci mohou navíc přímo evidovat svou vykonanou práci na projektu, sledování jeho průběhu je tedy snazší.

2.4.1 Výběr prostředí

Pokud se rozhodneme pro síťové řešení systému, nabízí se nám dvě možnosti implementace. Systém lze implementovat jako desktopovou nebo jako webovou aplikaci. Obě řešení mají své výhody i nevýhody, k jejich naprogramování lze využít různých programovacích jazyků.

desktopová aplikace Vytvořená aplikace může mít důmyslný vzhled i ovládání, jehož implementace je snazší než u webových aplikací. Při vytváření aplikace je však třeba se rozhodnout pro cílovou platformu, na které bude program fungovat (Windows/Linux).

webová aplikace Výhodou webové aplikace je fakt, že jako svůj klientský program používá internetový prohlížeč. Není tudíž třeba vytvářet nového klienta či řešit jeho platformovou závislost. Navíc dostupnost systému může být prakticky z jakékoliv internetové kavárny.

Nevýhodou webové aplikace může být méně příjemné ovládání, je tedy vyžadováno vyšší úsilí při návrhu vzhledu a ovládání aplikace.

2.4.2 Možnosti uchovávání dat

Stejně jako je velký výběr programovacích jazyků, ve kterých lze systém implementovat, existuje také více databázových produktů, ve kterých lze uchovávat data. Tyto produkty lze obecně rozdělit do dvou typů, a to souborové databáze a databázové servery.

souborové databáze Tyto databáze mají jednodušší implementaci než databázové servery. K ukládání dat využívají prostředků souborového systému (data se ukládají do souborů). Tyto databáze lze spolehlivě využívat v případech, kdy k databázi po většinu času přistupuje malý počet uživatelů. Při přístupu více uživatelů najednou může dojít ke snížení výkonnosti databáze.

databázové servery Tyto databáze vytváří komplexní systémy pro uchovávání dat. Umožňují paralelní přístup k datům pro více uživatelů současně. Jsou zde implementovány algoritmy pro zamezení vzniku konfliktů, které by vedly k poškození dat. Proto jsou tyto databáze vhodné pro náročné využívání více uživateli najednou. Navíc tyto systémy umožňují jednoduše provádět automatické zálohy dat, případně se z daných záloh opět rekonstruovat.

Právě z této kategorie je vhodné vybrat databázi pro uchovávání dat v nové aplikaci.

2.5 Zaměření na softwarové firmy

V předchozí části jsem se zaměřil na analýzu vlastností a možné řešení systému pro vytváření obecných projektových plánů. Nyní bych se chtěl více zaměřit na vlastnosti systému, které budou využívány při vytváření projektových plánů v softwarových firmách.

2.5.1 Zpětná vazba

V softwarové firmě může pracovat na vývoji nového programu mnoho lidí, jejichž funkce je rozlišena na projektové manažery, programátory a testery. Pro zvýšení produktivity uživatelů je vhodné, aby každý uživatel evidoval svou vykonanou práci, nebo aby zaznamenal nové připomínky či postřehy při řešení požadavků. Pokud budou mít i ostatní uživatelé řešící stejný požadavek umožněn přístup k těmto záznamům, bude reakce při řešení požadavku mnohem rychlejší.

Programátor například dokončí programování jistého modulu a vytvoří o tom záznam. Tohoto záznamu si všimne tester, a proto začne modul řádně testovat. Pokud objeví konkrétní chybu, nebo pokud se mu nebude zdát nějaká funkčnost testovaného modulu, upozorní na problém vytvořením nového záznamu. Programátorovi se následně tento záznam zobrazí, a proto se může na konkrétní chybu zaměřit a odstranit ji.

2.5.2 Ochrana přístupových práv

Se systémem bude pracovat hodně lidí, jejichž funkce na projektech je odlišná. Každý uživatel tedy nemůže vykonávat veškeré akce v systému, nemůže mít přístup ke všem informacím, protože některé informace mohou být důvěrné a určené třeba jen pro úzké vedení firmy. Je proto nezbytné definovat systém přístupových práv, který bude omezovat funkčnost systému pro jednotlivé pracovníky.

Projektový manažer by měl vytvářet projektové plány, přidělovat požadavky k řešení jednotlivým pracovníkům, sledovat a koordinovat průběh zpracování jednotlivých úkolů. Kdežto programátor by měl mít přístup ke svým zadaným úkolům, u kterých bude vytvářet záznamy o průběhu a výsledcích své práce. Bylo by nevhodné, aby samotný programátor mohl měnit klíčové vlastnosti projektového plánu.

Systém bude pravděpodobně dostupný také z Internetu, je tedy nezbytné ověřit identitu uživatele před vstupem do systému.

Kapitola 3

Požadované vlastnosti nového programu

3.1 Příbuzný software na plánování

3.1.1 MS Project

Mezi známé komerční programy na vytváření projektových plánů patří profesionální desktopová aplikace MS Project od firmy Microsoft [2]. Program umožňuje poměrně snadné vytváření úkolů, podúkolů, definování zdrojů a jejich přidělování k jednotlivým úkolům. Umožňuje přehledné znázornění dostupnosti zdrojů, aktuálního vývoje a ceny projektu. Samozřejmostí je možnost exportu dat do různých formátů, tvorba prezentací atp.

Tento program je však určen jen pro projektové manažery, kteří mají plnou kontrolu nad plánováním projektu. Chybí zde totiž zpětná vazba se samotnými uživateli, kteří jsou k daným úkolům přiděleni.

Právě na zpětnou vazbu uživatelů bych se chtěl v mém programu zaměřit.

3.1.2 dotProject

dotProject [1] je open-source¹ webová aplikace, zdarma dostupná. Slouží k vytváření projektových plánů, je postavená na PHP a databázi MySQL. Má příjemné uživatelské prostředí a díky modulárnímu zpracování nabízí spoustu vlastností: definování projektů a nových úkolů (s možností připojení poznámky či souboru), vytváření událostí v kalendáři, správa uživatelských kontaktů, diskusní fórum . . .

Navíc zde klienti mohou vkládat záznamy o svých problémech, stížnostech či návrzích, což umožňuje rychlou a účinnou klientskou podporu.

¹aplikace s dostupným zdrojovým kódem

3.1.3 PHProjekt

PHProjekt [5] je obdobně jako dotProject open-source webová aplikace, primárně určena pro koordinaci skupinových aktivit a ke sdílení informací a dokumentů. Díky modulární architektuře lze využít právě takové moduly, které budeme opravdu potřebovat.

Systém obsahuje velice propracovaný modul kalendáře umožňující zaslání upozornění emailem nebo i pomocí SMS (pokud je taková služba dostupná). Kalendář také obsahuje rezervační systém zdrojů (například pro rezervaci místností pro nějaké události).

Mezi další moduly patří emailový klient (možný přístup přes *pop3* či *imap*), diskusní fórum, chat, hlasovací zařízení . . .

Zajímavou vlastností je zaznamenávání času stráveného prací konkrétního uživatele pomocí tlačítek „Start“ a „End“. Díky této funkci lze také zobrazit všechny uživatele, kteří jsou právě „v práci“.

3.2 Naše požadavky

Jak je vidět z předchozích příkladů, náš problém lze pojmout různými způsoby, a proto existují různě komplexní možnosti zpracování.

Cílem mého snažení je vytvořit jednoduchý systém pro projektování a evidenci práce uživatelů.

Od programu jsou požadovány tyto základní vlastnosti:

- vytváření projektů
- vytváření úkolů a možnost jejich dělení na menší podúkoly
- vytvoření hierarchie uživatelů
- jednoduchý přístup uživatelů do systému
- přidělování uživatelů k projektům a úkolům
- evidence práce jednotlivých uživatelů
- zobrazení cenové nabídky jednotlivých úkolů
- import projektových plánů z programu MS Project

3.3 Specifikace požadovaných vlastností

3.3.1 Stromová struktura projektu

U nového projektu je třeba definovat hlavní úkoly. Tyto úkoly je potřeba dále dělit na menší a menší podúkoly do doby, kdy budeme schopni říct, že řešení daného požadavku je zřejmé. Tím nám vznikne stromová struktura projektu.

U jednotlivých dílčích úkolů je třeba odhadnout práci potřebnou k vyřešení daného problému. Tato práce by se měla zadávat v člověkohodinách (případně člověkodnech). Pokud práci odhadneme u všech úkolů, zjistíme potřebnou dobu k vypracování celého projektu.

Zobrazení projektu v systému by mělo mít stromovou strukturu s možností zobrazení či schování jednotlivých podúkolů. Zobrazení dílčích informací týkajících se nedodržení termínů, špatně odhadnuté doby potřebné ke zpracování úkolů či překročení odhadované ceny úkolů by mělo být jasné a zřetelné.

3.3.2 Uživatelé

V systému je třeba vytvářet uživatele, kteří budou mít do systému přístup (nejlépe po přihlášení pod svým jménem a heslem).

U uživatelů je třeba definovat jejich typ, dle kterého jim je umožněna (nebo znemožněna) práce v systému. Je třeba vytvořit systém přístupových práv na provádění dílčích akcí v systému pro jednotlivé typy uživatelů.

Je třeba rozlišovat tyto typy uživatelů:

administrátor spravuje celý systém

projektový manažer vytváří stromové struktury projektů, k jednotlivým úkolům přiřazuje uživatele

programátor, tester řeší přidělené úkoly, o své průběžné práci a pokroku vede záznamy

3.3.3 Evidence práce uživatelů

Uživatel přidělený k jednotlivým úkolům musí být schopen evidovat průběh své vykonané práce. Každý záznam o této práci by měl obsahovat časový údaj definující dobu strávenou danou prací a datum, ke kterému se daná práce vztahuje. Mělo by být umožněno tyto záznamy vkládat zpětně (uživatel nemusí mít v průběhu své práce přístup do systému).

3.3.4 Cenová nabídka úkolů

Abychom mohli odhadnout cenu daného úkolu, musíme nejprve u jednotlivých uživatelů definovat finanční ohodnocení (v korunách za hodinu), které bude zákazníkovi účtováno. Poté podle přidělené doby ke zpracování úkolu a podle přidělených uživatelů k danému úkolu můžeme odhadnout cenu úkolu.

Zároveň díky evidované odpracované době jednotlivých uživatelů lze snadno vypočítat aktuální cenu úkolu.

3.3.5 Import projektových plánů

Import projektových plánů je obzvláště vhodný v případě, kdy už máme plán vytvořený v jiné aplikaci. Je nutné nalézt správné mapování takových datových položek cizí aplikace, které bude možné v našem systému využít.

Aplikace MS Project nabízí silné prostředky pro vytváření projektových plánů, správu zdrojů a vytváření kalendářů dostupnosti jednotlivých zdrojů. Vytváření projektových plánů v této aplikaci je poměrně snadné, rychlé a velice efektivní². Navíc je aplikace poměrně hojně rozšířena, je proto vhodné, aby náš systém podporoval import projektů právě z MS Project.

²O základech plánování v MS Project 2003 se můžete dočíst v [3].

Kapitola 4

Datové struktury

Před návrhem datových struktur je třeba definovat základní pojmy, se kterými budeme pracovat. Pojmy jsou seskupeny podle příslušnosti do jednotlivých kategorií:

- vytváření projektových plánů

požadavek je zadání, které pochází buď od zákazníka nebo manažera projektu. Většinou neobsahuje technické informace, pouze je podnětem pro bližší analýzu a další zpracování projektu.

projekt reprezentuje soubor činností pro zajištění naplnění požadavku (plánovaného cíle)

úkol základní jednotka projektu, reprezentuje a popisuje část problému, který bude řešen. Komplexní úkoly lze dále dělit na jednodušší podúkoly.

- evidence zpracování projektu

uživatelská práce týká se konkrétního úkolu, slouží k zaznamenání postupu při řešení úkolu konkrétním uživatelem

- hierarchie uživatelů

uživatel pohybuje se v systému, plní úkoly a projektové plány

uživatelská funkce definuje možnosti uživatele v systému

Následuje popis datových struktur jednotlivých položek.

4.1 Projekt

Projekt reprezentuje soubor činností a informací týkající se jedné obchodní zakázky (požadavku).

Slouží k reprezentaci celé struktury úkolů, které se pod ním vytvářejí. Přiděluje se k němu skupina uživatelů, kterým je poté možné zadávat jednotlivé úkoly k řešení.

U projektu je třeba definovat tyto základní položky:

- název
- popis — může obsahovat jakékoliv informace týkající se daného projektu
- datum začátku práce na projektu
- datum očekávaného dokončení projektu
- přibližná doba potřebná ke zpracování projektu
- procentuální zpracování projektu — slouží k zaznamenání odhadovaného postupu vývoje projektu

4.2 Úkol

Úkoly jsou pevně spjaty s konkrétním projektem. Popisují požadavek, který má být vyřešen. Díky možnosti dělení větších úkolů na menší vzniká celá stromová struktura úkolů.

K jednotlivým úkolům lze přiřazovat jejich řešitele (ze skupiny lidí přidělených k projektu), kteří poté daný úkol řeší.

Úkolu je třeba definovat tyto základní položky:

- odkaz na nadřazený úkol
- projekt, ke kterému úkol patří
- název úkolu
- popis — může obsahovat podrobnější informace týkající se zpracování daného úkolu
- datum začátku práce
- datum předpokládaného dokončení
- přibližná doba potřebná ke zpracování daného úkolu
- procentuální zpracování úkolu — slouží k zaznamenání odhadovaného postupu zpracování úkolu

4.3 Uživatelská práce

Uživatelská práce¹ je záznam uživatele o vykonané práci u konkrétního úkolu za kratší časový úsek. Na základě těchto záznamů lze zjistit odpracovanou dobu úkolu a aktuální cenu.

Jelikož systém je převážně určený pro vývoj softwaru, bylo by vhodné rozlišovat práci programátorskou (samotné psaní programu) a práci testovací (testování již hotových částí).

U záznamů je třeba definovat tyto základní položky:

- odkaz na úkol, ke kterému záznam patří
- odkaz na uživatele, který práci vykonal
- stručný název práce
- podrobnější popis vykonané práce
- datum, ke kterému se daná práce vztahuje
- datum poslední editace záznamu
- celková doba spotřebovaná vykonáním popisované práce
- typ práce:
 - programátorská
 - testovací

Pro správné chování výpočtu ceny za vykonanou práci je třeba zaznamenat pevně ohodnocení uživatele v době vzniku záznamu.

4.4 Uživatel

Uživatel představuje v systému výkonnou složku — vytváří projektové plány, zaznamenává svou vykonanou práci. Je třeba umožnit uživateli bezpečný vstup do systému.

U uživatele jsou nezbytné tyto položky:

- login — unikátní jméno, pod kterým se uživatel přihlašuje do systému
- heslo — slouží k autentikaci uživatele
- uživatelská funkce — definuje možnosti uživatele v systému

Dále je u uživatele vhodné uchovávat osobní informace: jméno, příjmení, telefon, email, URL domovské stránky, případně poznámku o uživateli.

¹v systému též označováno jako *uživatelská poznámka*, či jen *poznámka*

4.5 Uživatelská funkce

Definováním uživatelských funkcí je možné vytvořit hierarchii uživatelů, která rozlišuje uživatele podle jejich možností vykonávat různé akce v systému, nebo která rozlišuje jejich platové ohodnocení účtované zákazníkovi.

Uživatelská funkce specifikuje míru oprávnění uživatele v systému, čili vlastní možnosti uživatelů. V základní verzi by se mohlo jednat například o administrátora, projektového manažera a programátora.

U uživatelské funkce je třeba definovat:

- název funkce
- popis
- stupeň oprávnění
- možnost přiřazení
- hodinová sazba účtovaná zákazníkovi u uživatele s touto funkcí

O uživatelských funkcích je blíže popsáno v kapitole 5.6.

Kapitola 5

Implementace

5.1 Formát systému

Zásadní otázka zní, jaký bude mít naše aplikace formát — zda-li se bude jednat o desktopovou aplikaci či webový systém.

Desktopová aplikace může být platformově závislá¹ nebo multiplatformní². Uživatel pro přístup do systému musí používat speciální klientský program, který ovšem může mít propracované uživatelské rozhraní či jiné zvláštní vlastnosti.

Webová aplikace nevyžaduje na uživatelském počítači žádný speciální klientský program, pro přístup do systému nám postačí obyčejný webový prohlížeč. Tím je systém dostupný prakticky z jakéhokoliv počítače. Navíc se uživatel nemusí učit ovládání nového programu, jelikož používá právě svůj dobře známý webový prohlížeč.

Dle požadavků na náš systém (hlavně kvůli možnosti přístupu „odkudkoliv“) jsem se rozhodl pro webovou aplikaci.

5.2 Použitý software

Další otázka zní, jaký programovací jazyk zvolíme a v jaké databázi budeme uchovávat data.

Implementaci systému jsem se rozhodl dělat v programovacím jazyku PHP [7], data uchovávám v databázi MySQL [4]. Výběr tohoto programovacího jazyka a databáze byl podpořen faktem, že mnoho poskytovatelů webového prostoru s oblibou používá servery Apache, automaticky již s podporou PHP a MySQL. Systém je tedy možné zprovoznit na velkém počtu serverů, některé z nich dokonce poskytují své služby zdarma.

¹na operačním systému Windows či Linux

²například při implementaci systému v Javě

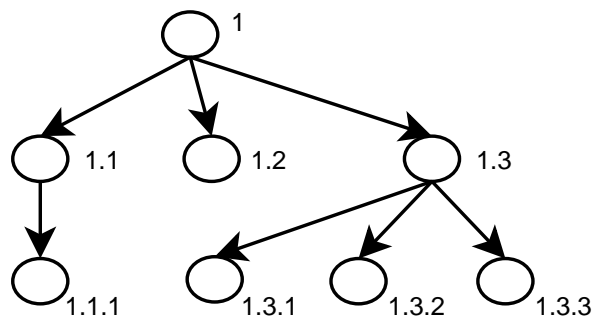
5.3 Datové struktury

Výběr správných datových struktur je nezbytný pro korektní běh celého systému. Jelikož komunikace mezi webovým prohlížečem a serverem má bezstavový charakter, musí se všechny informace uchovávat v databázi. Z databáze se poté informace získávají při každé žádosti o zobrazení systémové stránky.

5.3.1 Stromová struktura úkolů

Uchovat stromovou strukturu úkolů v relační databázi není jednoduché, jelikož stromová struktura má rekurzivní charakter. Jednotlivé podúkoly tedy musí obsahovat odkaz na nadřazený úkol. Problém nastává například při zjišťování závislosti mezi dvěma úkoly (zda-li je jeden úkol nadřazený druhému), nebo při zjišťování některých údajů v rámci celého podstromu.

Poměrně snadné zjišťování celé množiny úkolů patřící do konkrétního podstromu lze definováním nové položky úkolu, a to řetězec reprezentující cestu od kořene k danému úkolu. Tento řetězec je strukturovaný, jednotlivé části mohou být odděleny třeba tečkami. Cesta poduzlu obsahuje cestu nadřazeného uzlu plus identifikaci aktuálního uzlu (lze použít pořadí poduzlu v rámci nadřazeného uzlu, viz obrázek 5.1). Takto definovaná cesta je mimochodem použita k zaznamenání stromové struktury v programu MS Project [2].



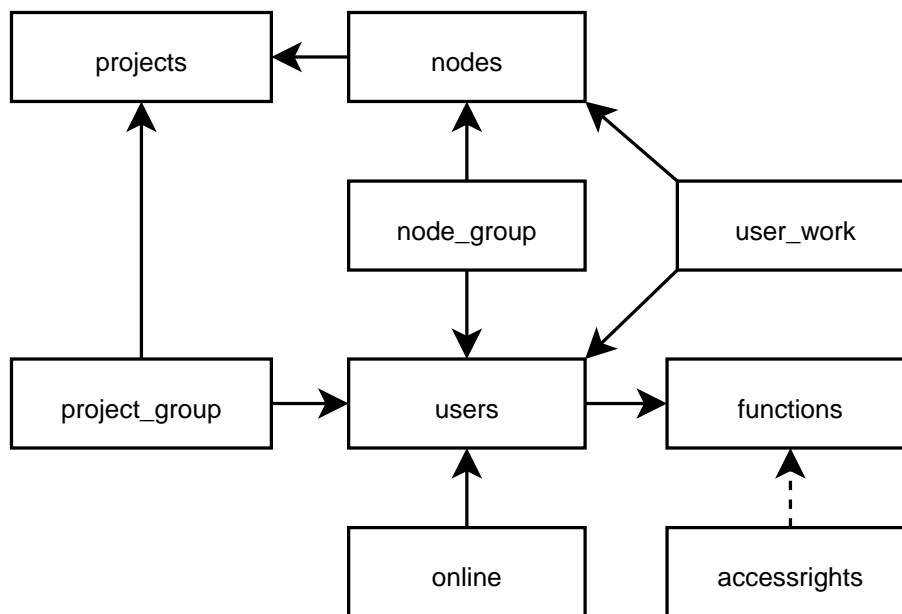
Obrázek 5.1: Popis stromové struktury pomocí cesty

Definování položky cesty je pěkné, leč její délka je neomezená. Pokud délku cesty omezíme, omezíme tím i maximální hloubku stromové struktury.

V mé implementaci stromové struktury úkolů jsem položky cesty nepoužil. Místo ní jsem u uzlu definoval další položky obsahující informace týkající se podstromu daného úkolu. V praxi to znamená, že získávání těchto položek z databáze je rychlejší (nemusíme provádět navíc další dotazy), nicméně vkládání či mazání podúkolu znamená úpravu těchto položek u všech úkolů na cestě ke kořeni.

5.3.2 Použité tabulky

V databázi jsou vytvořeny tabulky s datovými položkami, které jsou identické položkám popsanych v kapitole 4. Schéma struktury tabulek lze vidět na obrázku 5.2.



Obrázek 5.2: Schéma tabulek v databázi

Tabulka *projects* obsahuje informace o projektech — název, popis, začátek a konec práce, přibližnou dobu potřebnou ke zpracování projektu, odpracovanou dobu, procentuální zpracování projektu, aktuální cenu projektu.

Tabulka *nodes* obsahuje veškeré informace týkající se jednotlivých úkolů — název, popis, začátek a konec práce, přibližnou dobu potřebnou ke zpracování úkolu, odpracovanou dobu, procentuální zpracování úkolu, aktuální a odhadovanou cenu úkolu, hodinovou sazbu použitou pro výpočet ceny úkolu (sazba nijak neovlivňuje odhadovanou cenu podúkolů). Úkoly obsahují odkazy do tabulky *projects*, podle kterých se rozlišuje příslušnost úkolů k jednotlivým projektům. Každý úkol obsahuje odkaz na nadřazený úkol. Tyto odkazy vytvářejí stromovou strukturu projektů.

Tabulka *users* obsahuje veškeré informace o uživatelích v systému — jméno, příjmení, telefon, email, URL domovské stránky, poznámku. Možnosti uživatelů v systému jsou definovány pomocí uživatelské funkce, která je určena odkazem do tabulky *functions*. Pro možnost přihlášení do systému tu jsou položky login a heslo (uchováva se zašifrované). Pro možnost zobrazení informací o posledním přihlášení do systému se zde ukládá stará a nová IP adresa počítače, ze které uživatel přistupuje k systému, a také datum a čas posledního přihlášení do systému.

Tabulka *functions* obsahuje informace o uživatelských funkcích — název, po-

pis, stupeň oprávnění, možnost přiřazení, hodinovou sazbu účtovanou zákazníkovi. Blíže jsou uživatelské funkce popsány v kapitole 5.6.

Tabulka *accessrights* slouží k nastavení práv pro vykonávání jednotlivých akcí v systému. Záznamy obsahují tyto informace: název přístupového práva, popis, stupeň oprávnění minimálně nutný k vykonání dané akce. Položky tabulky jsou statické, byly vytvářeny v průběhu vývoje systému (jednotlivé názvy přístupových práv se přímo používají v kódu aplikace). Tato tabulka je nepřímo svázaná s tabulkou *functions*. Společně tvoří systém přístupových práv, který je blíže popsán v kapitole 5.7.

Tabulka *project_group* slouží k zaznamenání přidělení uživatelů k projektům. Každý záznam obsahuje odkaz na uživatele (tabulka *users*) a odkaz na projekt (tabulka *projects*), ke kterému je daný uživatel přidělen.

Tabulka *node_group* slouží k zaznamenání přidělení uživatelů k jednotlivým úkolům. Každý záznam obsahuje odkaz na uživatele (tabulka *users*) a odkaz na úkol (tabulka *nodes*), ke kterému je daný uživatel přidělen.

Tabulka *user_work* obsahuje informace o poznámkách (vykonané práci uživatelů). Tyto záznamy o vykonané práci jsou vázány na konkrétní úkol (pomocí odkazu do tabulky *nodes*) a na uživatele, který danou práci vykonal (odkaz do tabulky *users*). Dále záznamy obsahují tyto položky: název, popis, datum a čas poslední editace záznamu, datum a čas, ke kterému se poznámka vztahuje, spotřebovanou dobu, typ práce (programátorská, testovací), cenu za vykonanou práci, sazbu použitou pro výpočet ceny této poznámky (peněžní ohodnocení uživatele v době vzniku poznámky).

Tabulka *online* obsahuje záznamy o aktuálně přihlášených uživateli do systému. Každý záznam obsahuje hash-kód (řetězec, podle kterého je uživatel identifikován), odkaz na záznam daného uživatele (v tabulce *users*) a čas automatického odhlášení uživatele při jeho delší nečinnosti v systému.

Pro optimalizaci častých dotazů do databáze a také pro zajištění správné funkčnosti stromové struktury úkolů (viz sekce 5.3.1) jsou v některých tabulkách definovány redundantní datové položky, které lze z ostatních položek odvodit (viz tabulka 5.1).

tabulka	datové položky
projects	spotřebovaná doba prací na projektu aktuální cena projektu
nodes	přibližná doba potřebná ke zpracování všech podúkolů spotřebovaná doba prací na úkolu aktuální cena úkolu (včetně všech podúkolů) odhadovaná cena úkolu (včetně všech podúkolů)
user_work	sazba použitá pro výpočet přibližné ceny úkolu cena za vykonanou práci

Tabulka 5.1: Datové položky sloužící k optimalizaci dotazů do databáze

5.4 Zajištění bezpečnosti systému

5.4.1 Vstupní data

Veškerá data od uživatelů posílaná na server představují bezpečnostní riziko. Zpracování těchto dat musí být řádně ošetřeno, aby nedošlo k případnému zneužití systému.

Systém zpracovává vstupní data poslaná metodou POST a GET nebo poslaná v COOKIE. Pokud je aktivovaná funkce *Magic-quotes*, automaticky se všechny vstupní řetězce quotují pro bezpečné vkládání do dotazů do databáze. Toto chování je nevhodné, pokud takový řetězec nevkládáme přímo do databáze, ale pracujeme s ním jinak. Proto, pokud je aktivovaná funkce *Magic-quotes*, se nejprve všechny vstupní řetězce odquotují pomocí funkce *stripslashes*. Následně se pomocí funkce *addslashes* quotují takové řetězce, které se vkládají do dotazů do databáze. Pro bezpečný výpis proměnných do HTML se používá funkce *htmlspecialchars*, při vkládání hodnoty do URL se využívá funkce *urlencode*.

Proměnné reprezentující číselné hodnoty jsou konvertovány na čísla a následně jsou testovány na přípustné meze.

5.4.2 Přístup k databázi

Při instalaci systému dojde k vytvoření nové databáze v databázovém systému MySQL a zároveň k vytvoření nového databázového uživatele, který má povolený přístup pouze k této nové databázi. Při případném bezpečnostním incidentu, při kterém by došlo k vyzrazení přístupových informací k databázi, nemůže útočník ohrozit jiné databáze vytvořené v systému MySQL.

5.5 Bezpečný pohyb po systému

Uživatel se do systému přihlašuje pod svým uživatelským jménem a heslem. Následně je uživateli přidělen jednoznačný řetězec „hash-kód“, který se generuje v době přihlášení a je závislý na loginu uživatele a času přihlášení. Tento hash-kód se uvádí v URL adresách při přechodu na jiné stránky systému a díky němu je uživatel v systému identifikován.

Po odhlášení ze systému je hash-kód zneplatněn. Pokud se uživatel neodhlásí sám, bude z bezpečnostních důvodů po určité době odhlášen automaticky.

5.6 Uživatelské funkce

Mezi hlavní položky definované u uživatelské funkce patří přiřazení, přístupová práva a hodinová sazba, podle které se zákazníkovi účtuje vykonaná práce.

Typ přiřazení určuje, zda se provede kontrola přidělení uživatele k projektu/úkolů při pokusu o přístup k danému projektu/úkolů. Celkem jsou definovány tři typy přiřazení, blíže jsou popsány v tabulce 5.2.

typ přiřazení	popis
system	Uživatel „žije“ v celém systému, nepřirazuje se k žádnému projektu či dílčímu úkolů. Svou činnost může vykonávat všude.
projekt	Uživatel se přiřazuje k určitému projektu, ve kterém může vykonávat svou činnost. Mimo přiřazený projekt nemá povolený přístup.
uzel	Uživatel se přiřazuje k projektu a následně ke konkrétním uzlům (úkolům) daného projektu, kde může vykonávat svou činnost. K nepřiděleným projektům či uzlům nemá přístup.

Tabulka 5.2: Přiřazení uživatelských funkcí

Po instalaci systému máme vytvořené následující základní uživatelské funkce:

nobody jde pouze o teoretickou funkci v systému, nemá žádná práva a tedy nemůže v systému nic dělat (má přístupová práva jako nepřihlášený uživatel). Tato uživatelská funkce by měla sloužit pro bývalé pracovníky, jejichž vykonaná práce se stále uchovává v systému.

návštěvník jde také o teoretickou funkci v systému, přiřazuje se k projektu. Může zobrazovat pouze stromové výpisy projektů. Funkce je určena pro osoby „zvenčí“, kteří mají právo sledovat postup vývoje daného projektu.

tester přiřazuje se k úkolům, oproti předchozím funkcím může navíc zobrazovat informace o úkolech a vkládat k nim testovací poznámky.

programátor přiřazuje se k úkolům, navíc k nim může vkládat programátorské poznámky a editovat procentuální stav zpracování úkolů.

projektový manažer přiřazuje se k projektům, navíc může zobrazovat stromové výpisy nákladů, editovat úkoly, vytvářet nové podúkoly či rušit staré úkoly, upravovat poznámky ostatních uživatelů a k jednotlivým úkolům může přiřazovat uživatele.

administrátor systémová funkce. Navíc (oproti předchozím funkcím) vytváří uživatele v systému, může měnit jejich hesla, vytváří nové projekty, přiděluje k nim lidi, může vytvářet či editovat nové uživatelské funkce, importovat projekty z CSV souborů, přepočítávat jejich odhadované ceny.

V systému lze však vytvářet uživatelské funkce nové, což zvětšuje flexibilitu systému. Díky této možnosti můžeme jemněji rozlišovat přístupová práva k vykonávání jednotlivých akcí v systému, nebo můžeme definovat různé hodinové sazby. Lze tak rozlišovat různě zkušené uživatele či rozdílné kvalifikace osob.

5.7 Přístupová práva

Přístupová práva uživatelů jsou definována v uživatelských funkcích a mají charakter jednoduchého čísla z intervalu $[0, 255]$. Nižší číslo značí vyšší přístupová práva, 0 tedy reprezentuje práva administrátorská.

Každá akce v systému (např. zobrazení stromové struktury projektu, vytvoření nového úkolu . . .) je reprezentována jednoznačným tokenem. Tento token má definován minimální stupeň přístupového oprávnění potřebný k vykonání dané akce.

Výsledný efekt je takový, že funkce s vyšším přístupovým oprávněním mohou vykonávat všechny akce, které mohou vykonávat funkce s nižším oprávněním. K tomu mají možnost vykonávat některé akce navíc.

Tento systém oprávnění je jednoduchý a lehce implementovatelný. Jeho jedinou slabinou je nemožnost definovat uživateli přístupové oprávnění k právě jedné činnosti, která nemá definováno nejmenší přístupové oprávnění. V našem systému však tato nevýhoda nevádí.

Některé systémové akce mohou být vázány na konkrétní projekt či úkol, možnost vykonání takovéto akce poté závisí na přidělení uživatele k danému úkolu/projektu (např. zobrazení informací o úkolu vyžaduje přidělení uživatele k danému úkolu a zároveň k projektu, ke kterému úkol patří). Kontrola přidělení uživatele je závislá na typu přiřazení uživatelské funkce, kterou uživatel disponuje (blíže je přiřazení uživatelských funkcí popsáno v sekci 5.6).

5.8 Import projektových plánů

Jedním z požadavků na systém byl import projektových plánů z aplikace MS Project. Z této aplikace lze exportovat projektové plány do mnoha různých formátů. Pro náš systém je vhodné uvažovat dva formáty, a to XML a CSV.

MS Project definuje formát souboru XML a exportuje do něj veškeré informace týkající se projektového plánu a použitých zdrojů. Tento formát je velmi vhodný pro přenos dat mezi aplikacemi, jeho nevýhodou je značná velikost daná charakterem XML souborů.

Formát CSV je oproti XML velice jednoduchý, nenabízí však možnost ukládat více strukturovaná data. Z aplikace MS Project nelze tudíž exportovat veškeré informace do jediného CSV souboru (např. informace týkající se projektového plánu a podrobné informace o použitých zdrojích).

Jednotlivé položky CSV souboru jsou od sebe odděleny pomocí středníku.

Soubory lze tudíž editovat přímo či pomocí jakéhokoliv tabulkového editoru (například v MS Excel).

Do mého systému jsem implementoval import projektových plánů uložených ve formátu CSV. Díky jednoduchosti formátu CSV není import vázán jen na projektové plány z programu MS Project, můžeme si udělat vlastní plán v jakémkoliv tabulkovém editoru.

5.9 Shrnutí

Implementace systému pro plánování projektů a evidenci práce splňuje veškeré požadavky uvedené v kapitole 3.2. Důraz byl kladen především na dostupnost systému z Internetu (či z lokální sítě) a na přehledné zobrazení stromové struktury projektu včetně informací o aktuálním vývoji projektu.

Veškerý zdrojový kód je napsán v angličtině, není tudíž problém při případném budoucím rozšiřování aplikace využít mezinárodní spolupráce.

Kapitola 6

Pohled na vytvořenou aplikaci

6.1 Instalace

Před instalací aplikace již musíme mít funkční webový server s podporou PHP a dostupnou databází MySQL. Samotná instalace systému se provede překopírováním všech zdrojových kódů aplikace na server a spuštěním instalačního skriptu *instal.php* (blíže popsáno v uživatelské dokumentaci).

6.2 Ukázka aplikace

Pro jednoduchou orientaci v systému mají všechny stránky jednotný vzhled. Každá zobrazovaná stránka se skládá ze tří částí: záhlaví, menu a tělo zobrazované stránky.

Záhlaví zobrazuje titulek stránky (případně i doplňující podtitulek), login a funkci aktuálně přihlášeného uživatele s možností odhlásit (či přihlásit) se.

Pomocí menu se zobrazují jednotlivé stránky systému. Zobrazení jednotlivých položek menu je závislé na aktuálně zobrazené stránce a na stupni oprávnění přihlášeného uživatele. Pokud uživatel nemá právo některou akci provést, nebude se mu daná položka menu zobrazovat.

Tělo zobrazované stránky je pro každou stránku specifické. Na obrázku 6.1 je ukázkové zobrazení jedné systémové stránky.

6.2.1 Zobrazení stromové struktury

Jedním z cílů aplikace bylo přehledné zobrazení stromové struktury projektu s možností rozbalování a schovávání jednotlivých částí. Na obrázku 6.2 je ukázka takového zobrazení s informacemi o aktuálním zpracování daného projektu. Položky jsou zvýrazněny v závislosti na stavu zpracování jednotlivých úkolů — zda-li je možné úkoly dokončit v předepsaném termínu a zda byla korektně odhadnuta složitost zpracování jednotlivých úkolů.

Informace o úkolu
důkladnější specifikace

uživatel: **admin** [administrátor] [odhlásit](#)

Úvodní stránka [Zobrazit..](#)

Projekty Úkoly **Informace o úkolu** Strom Uživatelé Funkce

Info Přirazení lidí Změnit stav Editovat Nová poznámka Nový podúkol

název úkolu	důkladnější specifikace		
stav zpracování	100 %	aktuální cena	22525.00 Kč
začátek práce	09.06.2006 08:00	odhadovaná cena	22400.00 Kč
očekávaný konec	19.06.2006 17:00	výdaje	100.56 %
	(dny:hodiny:minuty)		
přibližná doba k vyřešení úkolu	14:00:00		
odpracováno	12:00:30		
spotřebovaná doba	86.16 %		
	Počet hodin na jeden den: 8		

Poznámky

typ	název	uživatel	doba	datum	akce
P	rozšíření základní specifikace	tomas	04:00:00	13.06.06 17:00	editovat
P	doplnění nezbytných údajů	boss	08:00:00	17.06.06 17:00	editovat
T	kontrola specifikace	boss	00:00:30	19.06.06 16:00	editovat

[zobrazit popisy všech poznámek](#)

Obrázek 6.1: Náhled na stránku systému

novinka

[-] Specifikace	100%	05.06.06	[22.06.06]	83.5%	20:07:00	[25:00:00]
[-] nástin problému	100%	05.06.06	[05.06.06]	118.75%	02:03:00	[02:00:00]
[-] rozvaha nad problémem	100%	06.06.06	[08.06.06]	93.75%	02:06:30	[03:00:00]
[-] důkladnější specifikace	100%	09.06.06	[19.06.06]	86.16%	12:00:30	[14:00:00]
[-] konečná specifikace	100%	20.06.06	[22.06.06]	60.42%	03:05:00	[06:00:00]
[-] Implementace	80%	23.06.06	[01.08.06]	90.44%	124:06:30	[138:00:00]
[-] gui	50%	23.06.06	[06.07.06]	60%	18:00:00	[30:00:00]
[-] základní jádro	100%	23.06.06	[20.07.06]	123.33%	74:00:00	[60:00:00]
[-] rozšiřující moduly	40%	21.07.06	[01.08.06]	67.19%	16:01:00	[24:00:00]
[-] síťové prostředí	60%	21.07.06	[01.08.06]	69.53%	16:05:30	[24:00:00]
[-] Testing	0%	21.07.06	[09.08.06]	7.95%	01:06:00	[22:00:00]
[-] test jádra	7%	21.07.06	[27.07.06]	17.5%	01:06:00	[10:00:00]
[-] test sítě	0%	02.08.06	[07.08.06]	0%	00:00:00	[08:00:00]
[-] test výkonnosti	0%	08.08.06	[09.08.06]	0%	00:00:00	[04:00:00]
[-] Revize	0%	10.08.06	[11.08.06]	0%	00:00:00	[06:00:00]
[-] shodnocení výkonnosti	0%	10.08.06	[10.08.06]	0%	00:00:00	[03:00:00]
[-] shodnocení produktu	0%	11.08.06	[11.08.06]	0%	00:00:00	[03:00:00]

Obrázek 6.2: Zobrazení stromové struktury projektu se zvýrazněním plnění časového plánu. Popis položek zleva: název úkolu; stav zpracování úkolu (100% značí splnění úkolu); datum zahájení práce; datum předpokládaného dokončení; dosud odpracovaná doba v procentech; odpracovaná doba (člověkodny:hodiny:minuty); předpokládaná doba pro zpracování úkolu (člověkodny:hodiny:minuty)

Význam podbarvení zobrazovaných položek:

datum zahájení práce úkolu položka je v čase před plánovaným počátkem zpracování úkolu podbarvena stříbrnou barvou.

odpracovaná doba položka se podbarvuje, pokud je úkol dokončen (stav zpracování je 100%). Podbarvení závisí na odpracované době, použití barev znamená:

- žlutá — na dokončení úkolu bylo potřeba méně než 80% předpokládané doby
- zelená — skutečná doba práce je v rozmezí 80–120% předpokládané doby
- červená — skutečná doba práce je vyšší než 120% předpokládané doby k vyřešení úkolu

očekávaný konec práce pole se podbarvuje po zahájení práce a podbarvení znázorňuje stav zpracování úkolu. Porovnává se stav zpracování úkolu s plánovaným časem vymezeným pro úkol. Po dokončení úkolu je pole bez podbarvení.

- žlutá — plnění úkolu je v předstihu
- zelená — práce probíhají v souladu s plánem
- červená — stav zpracování úkolu neodpovídá plánu, hrozí nebezpečí nesplnění očekávaného termínu
- hnědá — byl překročen plánovaný termín pro dokončení úkolu

dosud odpracovaná doba v procentech pole se podbarvuje, pokud ještě není úkol dokončen (stav zpracování není 100%). Porovnává se stav zpracování úkolu se skutečně odpracovanou dobou.

- žlutá — stav zpracování úkolu vykazuje o více než 20% nižší spotřebu času určeného ke zpracování úkolu, než bylo plánováno
- zelená — doba odpracovaná na úkolu odpovídá stavu zpracování úkolu
- červená — odpracovaný čas na úkolu je vyšší než plánovaný, celkový čas určený pro zpracování úkolu nebyl správně odhadnut

Na obrázku 6.3 je opět zobrazena stromová struktura projektu, tentokrát s informacemi o aktuální a odhadované ceně úkolů. Jednotlivé položky týkající se ceny mohou být zvýrazněny.

Podbarvení položek je následující:

dosud odpracovaná doba v procentech podbarvení je shodné s podbarvením popsaným u předchozího obrázku

novinka						
[-] Specifikace	100%	83.5%	92.56%	37025.00Kč	[40000.00Kč]
[-] nástin problému	100%	118.75%	118.75%	3800.00Kč	[3200.00Kč]
[-] rozvaha nad problémem	100%	93.75%	93.75%	4500.00Kč	[4800.00Kč]
[-] důkladnější specifikace	100%	86.16%	100.56%	22525.00Kč	[22400.00Kč]
[-] konečná specifikace	100%	60.42%	64.58%	6200.00Kč	[9600.00Kč]
[-] Implementace	80%	90.44%	96.42%	212900.00Kč	[220800.00Kč]
[-] gui	50%	60%	75%	36000.00Kč	[48000.00Kč]
[-] základní jádro	100%	123.33%	123.33%	118400.00Kč	[96000.00Kč]
[-] rozšiřující moduly	40%	67.19%	67.19%	25800.00Kč	[38400.00Kč]
[-] síťové prostředí	60%	69.53%	85.16%	32700.00Kč	[38400.00Kč]
[-] Testing	0%	7.95%	7.95%	2100.00Kč	[26400.00Kč]
[-] test jádra	7%	17.5%	17.5%	2100.00Kč	[12000.00Kč]
[-] test sítě	0%	0%	0%	0.00Kč	[9600.00Kč]
[-] test výkonnosti	0%	0%	0%	0.00Kč	[4800.00Kč]
[-] Revize	0%	0%	0%	0.00Kč	[9600.00Kč]
[-] shodnocení výkonnosti	0%	0%	0%	0.00Kč	[4800.00Kč]
[-] shodnocení produktu	0%	0%	0%	0.00Kč	[4800.00Kč]

Obrázek 6.3: Zobrazení stromové struktury projektu s cenovým rozpisem jednotlivých úkolů. Popis položek zleva: název úkolu; stav zpracování úkolu (100% značí splnění úkolu); dosud odpracovaná doba v procentech; aktuální výdaje v procentech; aktuální cena úkolu; odhadovaná cena dokončeného úkolu

aktuální cena pole se podbarvuje po dokončení úkolu (stav zpracování je 100%). Podbarvení závisí na konečné ceně úkolu.

- žlutá — na dokončení úkolu se spotřebovalo méně než 80% odhadovaných nákladů
- zelená — celkové náklady jsou v rozmezí 80–120% plánovaných nákladů
- červená — celkové náklady byly překročeny o více než 20%

aktuální výdaje v procentech pole se podbarvuje, pokud ještě není úkol dokončen (stav zpracování není 100%). Sledují se aktuální výdaje za odpracovanou dobu.

- žlutá — výdaje jsou nižší než plánované (minimálně o 20%)
- zelená — výdaje na zpracování úkolu odpovídají plánovaným
- červená — aktuální výdaje jsou vyšší oproti plánovaným (o více než 10%)

Kapitola 7

Závěr

Implementace webového systému pro plánování projektů a evidenci práce splňuje veškeré požadavky uvedené v kapitole 3.2. Systém je určen především pro menší a středně velké vývojové týmy či firmy zabývající se vývojem softwaru, kde by měl zvýšit produktivitu práce a zjednodušit sledování stavu vývoje projektu.

Mezi hlavní výhody programu patří možnost jednoduchého sledování postupu zpracování projektů. Je možné průběžně sledovat aktuální časovou i cenovou náročnost projektových prací. Aktuální informace o rozpracování projektů se porovnávají s naplánovanými údaji. Pomocí barevného zvýraznění jednotlivých položek systém upozorňuje na možné vznikající problémy časové i finanční, které je potřeba řešit pro korektní a včasné naplnění požadovaného cíle.

Další výhodou ulehčující práci je možnost importu již vytvořených projektových plánů ze souboru CSV do tohoto systému.

Systém není předurčen pouze pro plánování softwarových projektů, byl však vyvíjen s tímto záměrem.

7.1 Možnosti rozšíření

V kapitole 3.1 jsem popsal vlastnosti jiných softwarových produktů určených k plánování projektů, což mě přivedlo k myšlence možného rozšíření webového systému o tyto funkce:

- kalendář událostí — zobrazoval by začátky a konce jednotlivých úkolů, dále by se mohly definovat schůzky členů týmu. Upozornění na jednotlivé termíny by se mohla posílat emailovými zprávami nebo pomocí SMS.
- grafické znázornění postupu vývoje
- export projektových plánů — možnost exportu do formátu TXT, CSV či do definovaného XML souboru.

Literatura

- [1] dotProject: *Project Management Software*,
<http://www.dotproject.net/>, 2006.
- [2] Microsoft Corporation: *Microsoft Office Project 2003*,
<http://www.microsoft.com/cze/office/project/>, 2006.
- [3] Microsoft Corporation: *Seznámení se základy plánování projektů*,
<http://office.microsoft.com/cs-cz/assistance/HA010211711029.aspx>, 2006.
- [4] MySQL AB: *MySQL Reference Manual*,
<http://dev.mysql.com/doc/>.
- [5] PHPProject: *PHPProjekt - an open source groupware suite*,
<http://www.phpprojekt.com/>, 2006.
- [6] Soukeník J., Petráň M.: *Jak pomoci projektu*, Business World (červen 2005),
24–26.
- [7] The PHP Documentation Group: *PHP Manual*,
<http://www.php.net/manual/>, 2004.

Příloha A

Obsah CD-ROM

K této práci je přiložen CD-ROM, na němž se nachází zdrojový kód aplikace společně s uživatelskou a programátorskou dokumentací.

Instalace aplikace je popsána v uživatelské dokumentaci.

adresář/soubor	obsah
/projekt/	adresář obsahuje zdrojový kód aplikace
/dokumentace/uziv_doc.html	uživatelská dokumentace
/dokumentace/prog_doc.html	programátorská dokumentace
/prace/	tento dokument v elektronické verzi