

UNIVERZITA PARDUBICE

FAKULTA ELEKTROTECHNIKY A INFORMATIKY

DIPLOMOVÁ PRÁCE

2012

Bc. Radek Bitala

UNIVERZITA PARDUBICE

Fakulta elektrotechniky a informatiky

Návrh a implementace aplikace pro podporu řízení projektů

Bc. Radek Bitala

Diplomová práce

2012

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2011/2012

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Radek Bitala**
Osobní číslo: **I09353**
Studijní program: **N2646 Informační technologie**
Studijní obor: **Informační technologie**
Název tématu: **Návrh a implementace aplikace pro podporu řízení projektů**
Zadávací katedra: **Katedra informačních technologií**

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

Hlavním cílem diplomové práce je návrh a implementace systému pro správu uživatelských požadavků a řízení úloh. Úvodní část bude řešeršní s důrazem na porovnání současných řešení. V rámci práce bude implementováno vlastní řešení běžící jako webová aplikace. Součástí implementace bude analýza požadavků prostřednictvím UML diagramů pomocí softwarového nástroje Enterprise Architect, návrh datových tříd a jejich grafické vizualizace. Systém bude implementován s podporou databázového serveru MySQL a bude naprogramován v jazyce PHP s podporou frameworku Nette. Při implementaci budou dodržovány principy a zásady tvorby webových aplikací a budou dodrženy základní požadavky na bezpečnost systému.

Rozsah grafických prací:

Rozsah pracovní zprávy:

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

Seznam odborné literatury:

ARLOW, J., NEUSTADT, I. UML 2 a unifikovaný proces vývoje aplikací.

Computer Press, 2007;

SEDLÁČEK, Jiří, 1001 tipů a triků pro WWW stránky. 1. vyd. Brno :

Computer Press, 2003;

RIORDAN, R. Vytváříme relační databázové aplikace. Praha:Computer Press, 2000;

KOFLER, Michael; ÖGGL, Bernd. PHP 5 a MySQL 5 : Průvodce webového programátora. Praha : Computer Press, 2007;

James Taylor, Začínáme řídit projekty Praha:Computer Press, 2007;

Kathy Schwalbe, Řízení projektů v IT. Praha:Computer Press, 2011.

Vedoucí diplomové práce:

Ing. Jana Holá, Ph.D.

Katedra informatiky, managementu a radiologie

Datum zadání diplomové práce:

31. října 2011

Termín odevzdání diplomové práce:

18. května 2012

prof. Ing. Simeon Karamazov, Dr.
děkan



L.S.

prof. Ing. Antonín Kavička, Ph.D.
vedoucí katedry

V Pardubicích dne 15. listopadu 2011

Prohlášení autora

Prohlašuji, že jsem tuto práci 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.

V Pardubicích dne 18. 5. 2011

Bc. Radek Bitala

Poděkování

Na tomto místě bych rád poděkoval paní Ing. Janě Holé, Ph.D. za vedení, cenné rady a připomínky při tvorbě této práce. Dále také všem, kteří jakoukoli měrou přispěli k jejímu dokončení.

Anotace

Cílem této práce je návrh a implementace aplikace pro podporu řízení projektů. Teoretická rešeršní část je věnována prostudování a porovnáním současných nástrojů pro řízení projektů. Na základě zjištěných poznatků je navržen vlastní systém jako webová aplikace.

Klíčová slova

Nette Framework, MySQL, projekt, projektové řízení

Title

Design and implementation of applications to support project management

Annotation

The aim of this thesis is the design and implementation of applications to support project management. The theoretical part is devoted to search and study tools for comparing the current project management. According to findings is designed own system as a web application.

Keywords

Nette Framework, MySQL, project, project management

Obsah

1	Úvod	12
2	Problematika projektového řízení	13
2.1	Projekt	13
2.2	Charakteristika projektu.....	14
2.3	Projektové řízení	15
3	Současná řešení	17
3.1	Základní obecné funkcionality.....	17
3.2	Model implementace	19
3.2.1	Počítačové aplikace	19
3.2.2	Webové aplikace	20
3.3	Přehled vybraných komerčních softwarů pro projektové řízení.....	20
3.3.1	Microsoft Project.....	21
3.3.2	OnTime	21
3.3.3	ProjectManager.com.....	23
3.3.1	Clarizen	24
3.4	Shrnutí softwarových produktů	25
4	Implementace aplikace PROJECT MANAGEMENT EASY.....	27
4.1	Definice požadavků	27
4.1.1	Funkční požadavky	28
4.1.2	Nefunkční požadavky	29
4.2	Vybrané technologie pro návrh a implementaci systému	30
4.2.1	Nástroj pro systémovou analýzu a návrh Enterprise Architect.....	30
4.2.2	Nástroj pro grafické modelování a správu databáze MySQL Workbench.....	31
4.2.3	Vývojové prostředí NetBeans IDE	31
4.2.4	Balík předkonfigurovaných instalací XAMPP Lite	32

4.2.5	Databáze MySQL	32
4.2.6	Jazyk SQL	33
4.2.7	Jazyk PHP	33
4.2.8	Nástroj pro vytváření webových aplikací Nette Framework.....	34
4.2.9	Jazyk UML.....	37
4.3	Návrh Use-Case diagramů	38
4.3.1	Use-Case digram Projekty	39
4.3.1	Use-Case digram Úkoly.....	40
4.3.1	Use-Case digram Zprávy.....	42
4.3.1	Use-Case digram Kalendář	42
4.3.2	Use-Case diagram Administrace	43
4.4	Návrhový model tříd	44
4.5	Tvorba datového modelu	46
4.5.1	Datový model - Uživatel	48
4.5.1	Datový model - Projekt.....	49
4.5.1	Datový model – Úkol	51
4.6	Tvorba grafického designu systému.....	52
4.7	Programování systému.....	53
4.7.1	Systémová struktura	54
4.7.1	Autentizace a autorizace	57
4.7.2	Uživatelská část	58
4.7.1	Administrační část	60
4.8	Testování a nasazení systému	61
5	Závěr.....	63

Seznam obrázků

Obrázek 1: Projektový trojúhelník podle Harolda Kerznera	14
Obrázek 2: Ukázka Ganttova diagramu.....	17
Obrázek 3: Plugin OnTime v Microsoft Visual Studio	23
Obrázek 4: Zpracované funkční požadavky	29
Obrázek 5: Zpracované nefunkční požadavky s adresářovou strukturou požadavků	30
Obrázek 6: Schéma softwarové architektury MVP	36
Obrázek 8: Aktéři systému	39
Obrázek 9: Use-Case diagram Projekty.....	40
Obrázek 10: Use-Case diagram Úkoly	41
Obrázek 11: Use-Case diagram Zprávy	42
Obrázek 12: Use-Case diagram Kalendář	43
Obrázek 13: Balíčkový model návrhových tříd.....	46
Obrázek 14: Datový model – Uživatel	48
Obrázek 15: Datový model – Projekt	50
Obrázek 16: Datový model – Úkol.....	51
Obrázek 17: Design aplikace.....	53
Obrázek 18: Systémová struktura.....	54
Obrázek 19: Posloupnost kroků presenteru	55
Obrázek 20: Úvodní stránka aplikace - Přehled	58
Obrázek 21: Detail doručené textové zprávy	59
Obrázek 22: Ganttův diagram.....	60
Obrázek 23: Sekce zprávy	73
Obrázek 24: Formulář pro vytvoření projektu.....	75
Obrázek 25: Prostředí administrace.....	77

Seznam zkratek

ISO	International Organization for Standardization	mezinárodní organizace pro normalizaci
PMBOK	Project Management Body of Knowledge	standardní terminologie a pokyny pro řízení projektů
SMART	Specific, Measurable, Aligned, Realistic, Timed	mnemotechnická pomůcka používaná v projektovém řízení
CVS	Concurrent Version System	system sloužící ke správě verzí projektu
EVA	Economic Value ADDED	ekonomická přidaná hodnota
PERT	Program Evaluation and Review Technique	technika vyhodnocování a testování programů
DOS	Disc Operating System	diskový operační systém
MS	Microsoft	výrobce software
HTML	HyperText Markup Language	programovací jazyk
UML	Unified Modeling Language	modelovací jazyk
CASE	Computer Aided Software Engineering	počítačem podpořené softwarové inženýrství
MySQL	My Structured Query Language	databázový systém
IDE	Integrated Development Environment	integrované vývojové prostředí
SQL	Structured Query Language	strukturovaný dotazovací jazyk
PHP	Hypertext Preprocessor	programovací jazyk
LAMP	Linux, Apache, MySQL, PHP	sada svobodného softwaru pro implementaci
HTTP	HyperText Transfer Protocol	hypertextový přenosový protokol
ODBC	Open Database Connectivity	databázový systém
SMTP	Simple Mail Transfer Protocol	protokol elektronické pošty

SNMP	Simple Network Management Protocol	protokol správy sítě
FTP	File Transfer Protocol	protokol pro přenos souborů
IMAP	Internet Mail Access Protocol	poštovní protokol Internetu
POP3	Post Office Protocol 3	poštovní protokol Internetu
LDAP	Lightweight Directory Access Protocol	protokol pro přístup k adresářovým službám
AJAX	Asynchronous JavaScript and XML	programovací jazyk
DRY	Don't Repeat Yourself	princip z vývoje softwaru
KISS	Keep It Simple, Stupid	princip návrhu
MVC	Model-view-controller	softwarová architektura
MVP	Model-view-presenter	softwarová architektura
CSRF	Cross-site Request Forgery	metoda útoku na internetové aplikace
XSS	Cross Site Scripting	metoda útoku na internetové aplikace
URL	Uniform Resource Locators	popis umístění zdroje

1 Úvod

Současná doba vyžaduje realizaci mnoha projektů v co nejkratším časovém termínu a to včetně jejich případných změn, s minimalizací jejich nákladů i potřebných zdrojů pro realizaci v rámci sledovaných cílů. Metody na bázi pokusů a omylů jsou v dnešním konkurenčním prostředí nevyhovující, protože nároky na efektivitu projektu nám tuto cestu znemožňují. Vhodným nástrojem pro uspokojení jmenovaných požadavků je využití metod projektového řízení, jejichž primárním cílem je zajistit naplánování a realizaci úspěšného projektu. Tím se rozumí stav, kdy je v naplánovaném čase a se stanovenými náklady dosaženo cílů projektu. Projektové řízení je však velice komplexní problematikou, kdy pouhé zvládnutí jednotlivých nástrojů neznamena záruku úspěšného dosažení cílů. Při řešení projektu je nutno neustále sledovat a řídit celou řadu organizačních a řídicích vazeb, které se často v čase projektu výrazně mění. Rozhodně však platí, že projektové řízení je neoptimálnějším nástrojem pro realizaci projektů, jejichž charakter neumožňuje použít procesní modely řízení.

Informační technologie mají v současné době v oboru řízení projektů již nezastupitelnou roli. Trh je schopen nabídnout množství aplikací podporujících řízení projektů. Nutné však je položit si otázku, jak poznáme, zda se jedná o kvalitní software, který vyhovuje našim nárokům. Právě odpověď na ni je jedním z výstupů této diplomové práce.

Dílčím obsahem této práce je přehled současných trhem nabízených nástrojů pro podporu řízení projektů, který určuje obecné požadavky na aplikace tohoto typu. Následně je navržen a implementován vlastní systém pro podporu řízení projektů na principu webové aplikace pod názvem PROJECT MANAGEMENT EASY. Součástí práce je také stručný popis základní problematiky projektového řízení a popis vybraných technologií pro realizaci řešení. Nedílná část je také věnována problematice praktického nasazení systému.

2 Problematika projektového řízení

Před vlastní analýzou jednotlivých vybraných nástrojů projektového řízení je nutné vydefinovat význam několika základních termínů, které s touto problematikou úzce souvisí.

2.1 Projekt

Základem projektového řízení je především projekt, který je možné popsat jako sled aktivit a úkolů. Ty mají konkrétní cíl a jasně naplánovaný termín začátku a konce. Dalším parametrem jsou zdroje určující prostředky potřebné pro jejich realizaci. Obsahem každého projektu jsou nejčastěji 3 klíčové prvky:

- činnost,
- úkol,
- zdroj.

Činnost představuje konkrétní úkon s naplánovaným zdrojem nebo milník ohraničující určitou etapu procesu, u které lze sledovat návaznosti. Činnost jako taková určuje, co má být v rámci projektu provedeno, v jakém termínu, jakým způsobem a s jakými zdroji. Úkol v projektu zastupuje určitou povinnost, za kterou odpovídá pracovník. Poslední prvek - zdroj - je zodpovědný za definici kapacit a času pro každou činnost.[1]

Projekt je popisován také několika normami a standardy. Níže je pro ukázkou přiložena definice jedné z nich.

Definice normy ISO 10006: *“Projekt je jedinečný proces sestávající z řady koordinovaných a řízených činností s daty zahájení a ukončení, prováděný pro dosažení cíle, který vyhovuje specifickým požadavkům, včetně omezení daných časem, náklady a zdroji.”* [2]

Definice slova projekt existuje celá řada. Ve svém základu je však jejich význam v podstatě stále stejný, i když způsob popisu se liší.

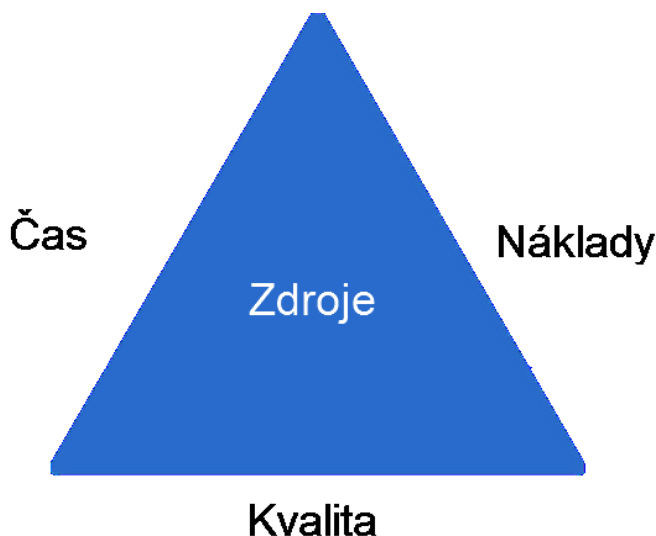
2.2 Charakteristika projektu

Každý projekt je unikátní a je realizován pouze jednou. Obecně však mají všechny projekty několik společných vlastností. Některé z nich vyplývají z výše uvedených definic projektu.

Vlastnosti, které má každý projekt, jsou:

- konkrétní cíl,
- definovaná kvalita,
- definované zdroje,
- určený termín,
- příslušné náklady,
- zvažovaná rizika,
- uvedená omezení.

Vlastnosti projektu lze také popsat pomocí trojúhelníku Harolda Kerznera, který bývá také označován jako *magický trojúhelník*. Ten pracuje se třemi základními parametry, které jsou vzájemně provázány. Konkrétně se jedná o kvalitu, čas a náklady. Principiálně trojúhelník říká, že v případě pokusu o zkrácení termínů projektu musí být automaticky počítáno se zvýšením nákladů nebo snížením kvality. Z pohledu zvýšení kvality je zase nutné počítat s prodloužením termínu nebo nárůstem nákladů. Nelze tedy změnit jeden z těchto parametrů bez promítnutí změny na ostatních parametrech.[3]



Obrázek 1: Projektový trojúhelník podle Harolda Kerznera[3]

Ideálním stavem je samozřejmě mít projekty realizované s minimálními náklady, v co nejkratším čase a s maximální kvalitou. V praxi je však tato idea nedosažitelná. Možné je pouze nalézt určitou proporcionalitu v závislosti na váze požadavků jednotlivých stran projektového trojúhelníku, kterou je potřeba definovat před zahájením projektu. Eliminují se tak případná neadekvátní očekávání od výstupů projektu.

Pro vydefinování požadavků jednotlivých stran projektového trojúhelníku je důležité především stanovit cíle, které mohou být pro každý projekt rozdílné. Úspěšné řízení projektu však vyžaduje držet se při jejich určování několika pravidel, která jsou definována například metodikou SMART.

SMART je účinným nástrojem používaným k určení několika charakteristických rysů každého úkolu. Definuje vlastnosti, které úkol musí mít, aby byl splněn. V případě absence některé z nich se jedná o chybějící informace.

Cíl by měl splňovat následující charakteristiky:

- specifický (**Specific**),
- měřitelný (**Measurable**),
- odsouhlasený (**Aligned**),
- realistický (**Realistic**),
- definovaný v čase (**Timed**). [4]

2.3 Projektové řízení

Pojem projektové řízení zahrnuje aplikaci dovedností, znalostí a doporučení tak, aby bylo možné v požadovaném termínu a s naplánovanými náklady uskutečnit projekt, který dosáhne stanovených cílů.

Předmětem projektového řízení je především samotný projekt, který určuje soubor činností. Ty je nutné naplánovat a provést tak, aby bylo dosaženo požadovaných cílů. V případě, že na projektové řízení bude pohlíženo, jako na systém s více paralelně běžícími projekty, souvisí s projektovým řízením dále také vytvoření organizační struktury projektů a koordinace disponibilních zdrojů.

Podle Rosenaua řízení projektu vyžaduje 5 odlišných manažerských činností:

- *„definování – definování projektových cílů;*
- *plánování – jedná se o tvorbu časového plánu projektu, finančního rozpočtu;*
- *vedení – uplatnění manažerského stylu řízení lidských zdrojů;*
- *sledování (monitorování) – kontrola stavu a postupu projektových prací, aby byly včas zjištěny odchylky od plánu a mohlo se včas přistoupit k jejich korekci;*
- *ukončení – ověření, že hotový úkol odpovídá aktuální definici toho, co se mělo udělat a uzavření všech nedokončených prací“ [5]*

Významným pojmem v oblasti projektového řízení je metoda kritické cesty, která umožňuje stanovit dobu trvání projektu na základě délky této cesty, což je soubor vzájemně závislých činností s nejmenší časovou rezervou. Metoda nám říká, na jaké úkoly by se měl manažer projektu soustředit. Pro kritické úkoly platí, že jejich časová rezerva se rovná nule. To znamená, že jakákoliv zdržení v těchto úkolech mají přímý vliv na dokončení celého projektu.[6]

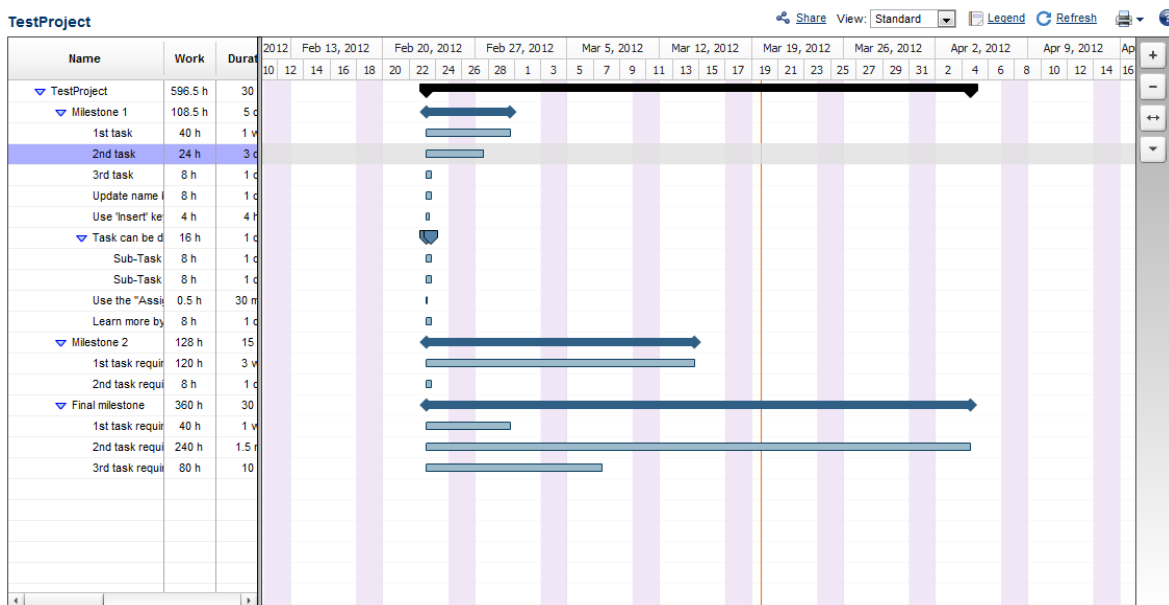
Účelem projektového řízení je řídit projekt jako celek směřující k cíli, nikoliv jako realizaci jednotlivých úkolů projektu dle plánu. Pouze v případě, že projekt je dobře naplánován jako celek, může se projektový manažer zaměřit na úzkou skupinu úkolů a tu řídit podrobně. V případě dodržení byť jen několika základních pravidel se řízení projektu výrazně zkvalitní.

3 Současná řešení

Tato kapitola je zaměřena na možnosti aktuálních trhem nabízených řešení pro podporu projektového řízení. Je zde popsána obecná skladba funkcí s následným popisem několika konkrétních řešení.

3.1 Základní obecné funkcionality

Základní funkcionality, které poskytuje většina současných řešení lze rozdělit do 2 skupin. První skupinou je vlastní projektové řízení, které nám poskytuje možnost založení projektů a podprojektů, vydefinování odpovídajících rolí, alokaci disponibilních zdrojů a delegování projektových úkolů členům týmu. Důležitou funkcí je zde sledování stavu delegovaných úkolů tak, aby projektový manažer získal širší pohled na projekt jako celek. Stejně tak je možné setkat se s kontrolou efektivity plnění jednotlivých úkolů na bázi vyplnění, odeslání a následného schválení pracovního výkazu. Hojně využívaným prvkem v této části je Ganttův diagram, který slouží k zobrazení časové náročnosti a posloupnosti jednotlivých částí projektu.



Obrázek 2: Ukázka Ganttova diagramu
Zdroj: on-line software Clarizen¹ [cit. 2012-03-19]

¹ www.clarizen.com

Při práci na jednotlivých projektech je další podstatnou funkcí zajištění vzájemné spolupráce členů projektového týmu. Do této kategorie patří jednak nástroje pro e-mailovou komunikaci a jednak různé textové komunikační nástroje. V případě změn na projektu, které jsou vzhledem k členům projektového týmu globálního charakteru, je důležitá rozumná forma informovanosti jednotlivých členů. To znamená stav, kdy je informovanost dostatečná tak, aby při větším množství projektů existovaly informace pouze o důležitých změnách, a zároveň nebudou členové týmu informováni o nepodstatných úpravách, které je zahltlí a dojde tak k přehlížení důležitých oznámení. Dalším účinným nástrojem této kategorie, který umožňuje sdílení informací na jednom společném prostoru je tzv. Wiki.

„Wiki je označení webů (nebo obecněji hypertextových dokumentů), které umožňují uživatelům přidávat obsah podobně jako v internetových diskusích, ale navíc jim také umožňují měnit stávající obsah; v přeneseném smyslu se jako wiki označuje software, který takovéto weby vytváří. Wiki umožňuje vytvářet dokumenty kolektivně pomocí jednoduchého značkovacího jazyka za použití webového prohlížeče.“ [7]

Výše uvedené dělení bylo provedeno na základě běžně dostupných aplikací pro podporu projektového řízení, které jsou níže popsány. Řešení mohou mimo obecných funkcí obsahovat i další rozšiřující funkce, která řízení projektu zpřesňují. Mohou být shrnuty takto:

- evidence externích nákladů,
- evidence aktuálních nákladů,
- evidence projektových změn,
- evidence projektových oprav,
- grafická zobrazení aktuálního stavu projektu,
- automatická upozornění na změny a rizika,
- sledování vytížení zdrojů,
- integrace do vývojových a komunikačních nástrojů,
- exporty informací o aktuálním stavu do Microsoft Excelu,
- exporty a importy s Microsoft Project.

3.2 Model implementace

Aplikace pro podporu řízení projektů je možné rozdělit do 2 základních kategorií – počítačové aplikace a webové aplikace. Jejich vlastnosti jsou níže podrobně popsány. Kritériem zvoleného dělení je použitý model implementace.

Každou z výše uvedených kategorií je možné dále rozčlenit do podrobnějších podkategorií dle specializovaných vlastností těchto aplikací. Příkladem takového dělení mohou být následující kategorie.

- Integrované aplikace – Aplikace kombinující řízení projektu s mnoha dalšími aspekty života firem. Každý projekt může být například propojen s informacemi o konkrétním zákazníkovi, pro kterého je projekt vytvářen. Principem je využití informací z existujících systémů společnosti a opačně přenos informací do těchto systémů.
- Jednouživatelské aplikace – Aplikace určené pro práci s jedním uživatelem a nikoliv ke sdílení projektových informací dalším členům projektového týmu. Typicky jsou takové aplikace využívány v malých firmách, kde je projektové řízení doplněno dalšími nástroji mimo tyto aplikace.
- Open-source aplikace – Aplikace určené pro firmy, které nedisponují dostatečnými finančními prostředky pro nákup programového vybavení. Přístup ke zdrojovým kódům bývá většinou neomezený. Na řešení případných systémových chyb se tak může podílet širší skupina lidí. Tento fakt bohužel může pomoci případnému útočníkovi nalézt bezpečnostní díry. Typickými aplikacemi této kategorie jsou počítačové aplikace.

3.2.1 Počítačové aplikace

Projektové řízení je u této kategorie aplikací dostupné prostřednictvím programu, který běží na počítači každého uživatele. Díky tomu se tyto aplikace vyznačují nízkou odezvou a kvalitním grafickým provedením. Projektová data jsou uložena v počítači každého člena projektového týmu, anebo je lze sdílet prostřednictvím centrálních databází a serverů. Ty jsou často přímo ve vlastnictví provozovatele systému. Umožněna je tak kooperace s dalšími aplikacemi ostatních uživatelů.

Výhody a nevýhody počítačových aplikací:

- + vyšší počet nástrojů ve srovnání s webovými aplikacemi,
- + vhodné pro rozsáhlé projekty velkých firem,
- + vyšší bezpečnost dat než u webových aplikací,
- vyšší pořizovací náklady,
- vyžadují instalaci software na každý počítač.

3.2.2 Webové aplikace

Nástroje pro podporu řízení projektů mohou být také implementovány jako webové aplikace, které jsou dostupné pomocí internetového prohlížeče. Datové úložiště i logika aplikace je umístěna na webovém serveru. Pořizovací náklady bývají zpravidla nižší ve srovnání s klientskými, a jsou tak vhodnou volbou pro menší až střední firmy.

Výhody a nevýhody webových aplikací:

- + lze k nim přistupovat z jakéhokoliv počítače bez nutnosti instalace software na počítači uživatele,
- + snadné řízení přístupu,
- + společná verze software a instalace,
- obvykle pomalejší odezva než u klientské aplikace,
- umístění dat na webovém serveru,
- informace o projektu nejsou v případě internetového připojení k dispozici.

3.3 Přehled vybraných komerčních softwarů pro projektové řízení

Software pro podporu řízení projektů existuje v mnoha různých úrovních sofistikovanosti a nákladovosti. Cílem této kapitoly je výběr reprezentativního vzorku několika aplikací, jejichž funkční provedení bude předlohou pro tvorbu vlastní aplikace.

3.3.1 Microsoft Project

Nástroj Microsoft Project spadá do kategorie počítačových aplikací jakožto součást kancelářského balíku Microsoft Office od společnosti Microsoft. Slouží k podpoře projektového řízení, správě úkolů i zdrojů a zjišťování aktuálního stavu projektu. Poskytuje různé výstupy: Ganttův diagram, kalendáře, přehled peněžních toků, analýzy EVA a PERT atd.[8]

Grafické zpracování vychází z běžné produktové řady aplikací MS Office. Nabízeny jsou celkem 4 základní karty (Úkol, Zdroj, Projekt, Zobrazení), které jsou přehledně a tematicky uspořádány. Ve verzi Professional je přidána karta Team s nástroji pro tvorbu projektového týmu. Pod nástrojovými kartami ve standardním zobrazení nalezneme seznam úkolů projektu a jejich grafické zobrazení v Ganttově diagramu.

Aplikace je ve verzi Standard určena pro správu menších projektů bez možnosti koordinace projektového týmu, sdílení dokumentů a verzování. Verze Professional nabízí vše co je k dispozici ve verzi Standard, dále však přidává komponenty pro tvorbu projektů v týmu a lze ji v posledním vydání z roku 2010 napojit na Microsoft SharePoint Server 2010 nebo Microsoft SharePoint Foundation 2010. Díky tomu lze ukládat o projektu informace, ke kterým mohou následně přistupovat další spolupracovníci. V této verzi aplikace plně komunikuje s MS Project Serverem, kam lze kromě vlastních informací o projektu verzovat, sdílet dokumenty pomocí Sharepointu, sdílet zdroje a využívat další komponenty serveru.

Díky vhodně zvolenému rozložení prvků a způsobu ovládání vycházejícího z velmi rozšířených produktových balíčků Microsoft Office, působí aplikace dobrým a přirozeným dojmem. Adaptace na práci se systémem je tak velice rychlá. V současnosti již také produkt existuje v poměrně velkém počtu lokalizací. Jazyková bariéra pro práci s tímto řešením tedy není komplikací.

3.3.2 OnTime

Systémem, který spadá do kategorie počítačových i webových aplikací je OnTime² od společnosti Axosoft. Ta je předním dodavatelem řešení pro řízení projektů se

² www.axosoft.com

specializací na vývoj software. Jejich vlajkový produkt OnTime využívá více než 8 000 vývojových týmů v 55 zemích po celém světě.

Produkt je možné provozovat ve dvou variantách. První z nich je hostované on-line řešení u dodavatele. Řešení je tak dostupné z počítače přes běžný webový prohlížeč. Druhou variantou je instalace aplikace na vlastní počítač a její následné připojení.

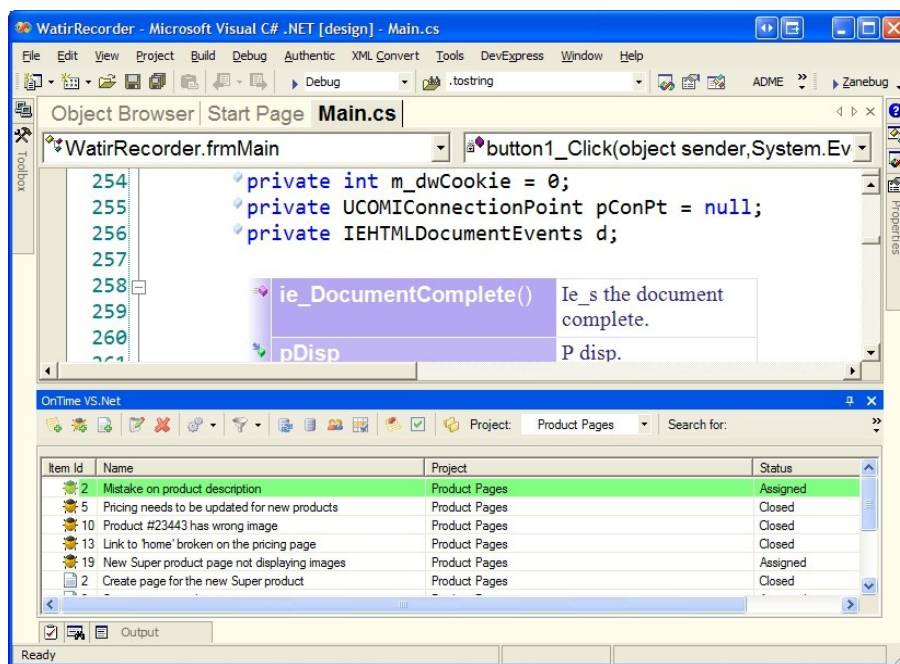
OnTime je velmi propracované řešení, které disponuje funkčním a v případě webového rozhraní i kvalitním grafickým řešením. Je tedy důkazem, že dobrý software může mít kvalitní vzhled, být snadno ovladatelný, a přesto obsahovat všechny funkce, které jsou při projektovém řízení potřeba. Přívětivou vlastností webového prostředí je funkce Drag and drop a možnost použití klávesových zkratk. Pro vyšší mobilitu je k dispozici i aplikace pro „chytré“ telefony a tablety. Velmi zajímavá funkce, která stojí za zmínku je integrace tohoto nástroje pomocí pluginů do produktů Microsoft Visual Studio a Eclipse. Díky nim nemusí programátor pracující na zadaných úkolech přepínat mezi okny programů. Kompletní seznam úkolů a základní nástroje pro práci s nimi má totiž plně k dispozici přímo v pracovních aplikacích.

Základem aplikace je stejně jako u všech aplikací tohoto typu projekt, ke kterému lze vytvářet další podprojekty. Jednotlivým projektům následně přiřazujeme konkrétní úkoly s definovanými parametry (termín zahájení a ukončení, přiřazený zdroj, popis, přílohy atd.). Pro správu verzí integruje řešení primárně se systémem GitHub³. Dále je podporován pomocí pluginu i verzovací systém TortoiseSVN⁴. Prostředí obsahuje další funkce jako je například konfigurovatelný systém oznamování umožňující udržet komunikaci se vzdálenými členy týmu a s dalšími zúčastněnými stranami. Tato komunikace probíhá prostřednictvím e-mailů nebo aktualizací wiki stránky.

Autoři tohoto řešení vsadili především na jeho kvalitní grafické zpracování, které je skutečně na vysoké úrovni. Bohužel z funkčního hlediska lze nalézt nedostatky. Nejvýznamnější je absence evidence odpracovaného času. Kvůli tomuto nedostatku nelze sledovat v systému výkazy za odvedenou práci, a s tím související aktuální nákladovost.

³ www.github.com

⁴ www.tortoisesvn.net



Obrázek 3: Plugin OnTime v Microsoft Visual Studio

Zdroj: www.axosoft.com/ontime [cit. 2012-03-01]

3.3.3 ProjectManager.com

Dalším zajímavým nástrojem na podporu řízení projektů je ProjectManager.com⁵ od stejnojmenné společnosti založené v roce 2008. Celé řešení je nabízeno jako webová aplikace, nicméně pro uživatele, kterým tento způsob nevyhovuje, se výrobce vydal velmi zajímavou cestou. Projekt je možné spravovat z výše popsané aplikace Microsoft project a následně ho se všemi informacemi importovat do aplikace. Díky této funkci uvidí projektový tým všechny informace v systému ProjectManager.com.

Grafické provedení je velmi praktické a má širokou škálu modulárnosti. V případě, že některé moduly tohoto řešení nejsou využívány, lze je jednoduše vypnout. Rozumně je také umožněna volba nastavení upozorňování na změny v projektech, aby si uživatel sám určil, které informace jsou pro něj důležité. Aplikaci si tak přizpůsobíme přesně svým potřebám.

Projekt je zde reprezentován podobně jako u ostatních aplikací souborem úkolů společně s Ganttovým diagramem. Před vytvořením prvního projektu je potřeba nejprve nastavit příslušné zdroje pro projekt, které následně k jednotlivým úkolům přidělujeme.

⁵ www.projectmanager.com

Aplikace obsahuje velké množství nástrojů pro plánování, sledování a grafické vyhodnocování stavu projektů, a to jak z pohledu termínů, tak i nákladů. K jednotlivým projektům je možné přidávat poznámky, diskutovat a přikládat soubory. Dále je do prostředí integrována e-mailová komunikace nebo kalendář s vyznačenými úkoly. ProjectManager.com také disponuje velkou podporou tutoriálů ve formě textu i videa, automatickou aktualizací a tiketovacím systémem pro hlášení chyb.

ProjectManager.com celkově působí velmi dobrým dojmem, a to i navzdory menšímu vývojovému týmu produktu, kterému se podařilo vytvořit opravdu výrazný produkt v daném segmentu.

3.3.1 Clarizen

Poslední ukázkou aplikace je řešení Clarizen⁶. Vyvinula ho stejnojmenná společnost založená v roce 2005, a to s jasným cílem vytvořit aplikaci pro řízení projektů. První verze byla uvolněna v roce 2007. Řešení je kompletně koncipováno jako webová on-line aplikace.

Grafickým rysem řešení je na první pohled až příliš jednoduché rozhraní, které se svou navigací velmi blíží e-mailovým službám. Ačkoliv zprvu nenadchne, při využívání se ukazuje jeho vysoká použitelnost. Při přihlášení do aplikace se zobrazí úvodní obrazovka se seznamem projektů a úkolů, které jsou na daný den naplánovány. Rozhraní umožňuje přidávat, mazat a aktualizovat poznámky, úkoly, různé zdroje a milníky. Díky webovému provedení lze v systému pracovat bez ohledu na využívaný operační systém, pouze pomocí prohlížeče. Podobně jako u výše zmíněného řešení je možné projekt spravovat také z produktu Microsoft Project a informace následně importovat do Clarizen. K dispozici je navíc také plugin pro aplikaci Microsoft Outlook. Lze tak sledovat své úkoly přímo v tomto e-mailovém klientu.

Pro týmovou spolupráci slouží diskuse k jednotlivým projektům a úkolům, Wiki, poznámky a sdílení dokumentů. Clarizen je také schopen publikovat vlastní plány a Ganttův diagram prostřednictvím e-mailu nebo pomocí speciálního widgetu v HTML podobě. Každý člen týmu tedy může dle potřeby tyto plány sdílet kolegům. S vlastním plánem je možné jednoduše manipulovat pomocí funkce Drag and drop. V případě, že

⁶ www.clarizen.com

v systému naleznete úkol, který chcete zahrnout do svého plánu, stačí jej jednoduše do něj myší přetáhnout.

Zdroje Clarizen umožňuje přidávat pro celý projekt nebo konkrétní úkol a lze je dělit na interní a externí. Pro každý zdroj si můžeme zobrazit všechny úkoly, které má naplánovány. Systém také umožňuje nastavení automatického upozornění, je-li některý člen týmu přetížen nebo naopak má podhodnocen počet úkolů. Můžeme tak úkoly rovnoměrně rozdělit mezi zdroje a tím maximalizovat produktivitu projektového týmu.

Clarizen jako software pro řízení projektů disponuje on-line výukovým programem. Ten uživatele pomocí videí provede při každé první návštěvě nové sekce. Srozumitelně a podrobně vysvětlí funkce daného modulu. K dispozici je také e-mailová podpora, on-line chat a fórum.

Celkově se jedná o velmi komplexní nástroj, který se z pohledu počtu obsažených funkcí řadí mezi špičkové produkty. Paradoxně z této výhody plyne i určité riziko pro nové uživatele, kteří pro své produkty nepotřebují tak robustní řešení. Tito uživatelé se totiž mohou při používání systému cítit do jisté míry zahlceni počtem funkcí, které ale pravděpodobně nikdy nevyužijí.

3.4 Shrnutí softwarových produktů

Každý z výše uvedených systémů patří mezi komerční software, za jejichž vývojem stojí týmy specialistů. Obsahují všechny nástroje potřebné k vlastnímu řízení, liší se spíše použitou technologií a modelem implementace. Vzhledem k široké škále potencionálních zákazníků v tomto oboru nelze „favorita“ obecně určit. Každá firma musí dle svých potřeb a možností určit, zda všechny nabízené komponenty daného nástroje využije a jakými prostředky pro provoz systému disponuje. V případě nutnosti pořízení vlastních serverů je jejich nákladnost nezanedbatelnou položkou a právě cena je jedním z důležitých faktorů ovlivňujících výběr konkrétního nástroje. Nespornou výhodou však je v takovém případě vyšší bezpečnost uchovávaných dat z pohledu rizika případných úniků projektových informací do nepovolaných rukou.

Výběr konkrétního nástroje i vlastní zavedení projektového řízení do firmy musí také samo o sobě probíhat jako správně připravený a řízený projekt, aby byla maximálně

snížena možnost jeho neúspěchu. Důležité však je uvědomit si výhody zavedení, které výrazně zvýší konkurenceschopnost, a to zejména v těchto bodech:

- minimalizace rizik v projektovém řízení,
- snížení nákladů na vytvářené projekty,
- zkrácení a jasné definování termínů projektů,
- zvýšení kvality managementu,
- možnost spolupráce na zahraničních projektech (západní firmy v dnešní době často požadují projektové řízení jako standard).

Na základě získaného přehledu o současných softwarových řešeních pro podporu řízení projektů bylo možné stanovit charakteristiku aplikace, jejíž návrh a implementace je cílem praktické části práce. Shrnout se ji lze takto. Aplikace dostupná pomocí webového prohlížeče, jenž umožňuje snadnou realizaci projektové řízení prostřednictvím nástrojů pro práci s projekty, projektovými rolemi, zdroji a úkoly. Podporující sledování stavů delegovaných úkolů a komunikaci mezi členy projektového týmu.

4 Implementace aplikace PROJECT MANAGEMENT EASY

Tato kapitola popisuje postup implementace vlastního řešení pro podporu řízení projektů s názvem PROJECT MANAGEMENT EASY. Celý postup vývoje byl vytyčen několika milníky, které představují jednotlivé dílčí kroky vedoucí k úspěšné realizaci celého řešení. Jejich určení výrazně napomohlo k určení časového harmonogramu vývoje aplikace.

Vytyčené milníky lze je shrnout takto:

- definice požadavků na systém,
- výběr vhodných nástrojů, technologií a postupů,
- návrh Use-Case diagramů,
- návrhový model tříd,
- tvorba datového modelu,
- tvorba grafického designu systému,
- programování systému,
- testování a nasazení systému,
- zaškolení uživatelů.

4.1 Definice požadavků

Konečná podoba každého systému je postavena na množině požadavků. Z tohoto důvodu je jejich definice klíčovým prvkem při vývoji celého systému. Jejich podceněním mohou vzniknout závažné komplikace.

„Inženýrství požadavků (requirements engineering) je termín používaný k popisu aktivit zapojených do zjišťování, dokumentování a údržby množiny požadavků na softwarový systém. Zastupuje odhalování způsobu, jak a k čemu uživatelé daný systém potřebují.“ [19]

Ze získaného přehledu o existujících komerčních nástrojích byly stanoveny funkční požadavky na základní řešení. Kompletně jsou zpracovány v programu Enterprise Architect.

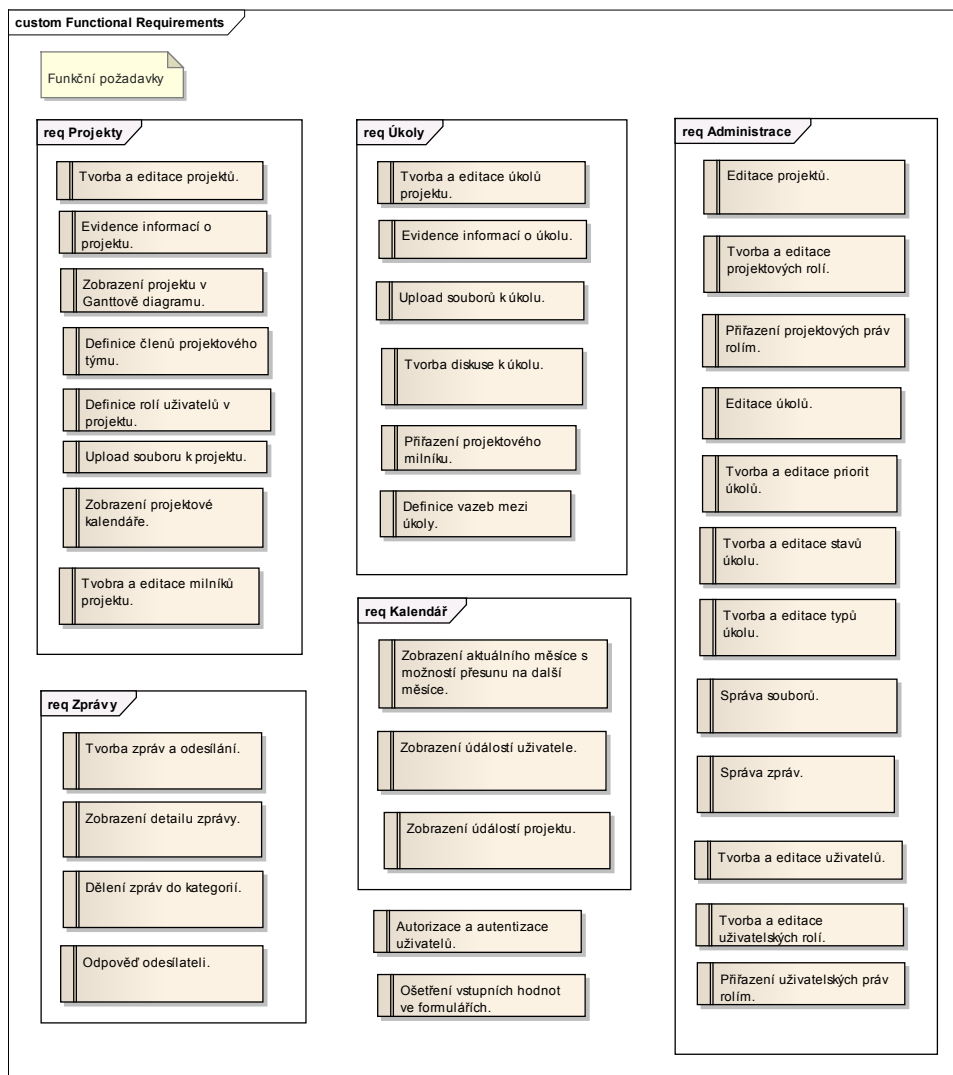
Požadavky je vhodné dělit do několika kategorií. Často používané základní rozdělení je na funkční a nefunkční požadavky. Pro přehlednost je toto dělení dále rozšířeno o několik dalších podkategorií dle struktury aplikace.

4.1.1 Funkční požadavky

Funkčním požadavkem rozumíme popis toho, co by měl systém dělat, tedy popis funkcí, které je nutné naprogramovat.

Seznam vybraných funkčních požadavků:

- ošetření vstupních hodnot ve formulářích,
- tvorba a editace projektů,
- zobrazení projektu v Ganttově diagramu,
- definice členů projektového týmu,
- definice rolí uživatelů v projektu,
- tvorba a editace milníků projektu,
- tvorba diskusí k úkolu,
- upload souborů k projektu a úkolu,
- zobrazení projektového kalendáře,
- zobrazení uživatelského kalendáře,
- tvorba a editace úkolů projektu,
- tvorba zpráv a odesílání,
- dělení zpráv do kategorií,
- tvorba a editace uživatelů,
- autorizace a autentizace uživatelů,
- tvorba a editace uživatelských rolí,
- tvorba a editace projektových rolí,
- přiřazení přístupových uživatelských a projektových práv rolím,
- tvorba a editace priorit, stavů a typů úkolu.



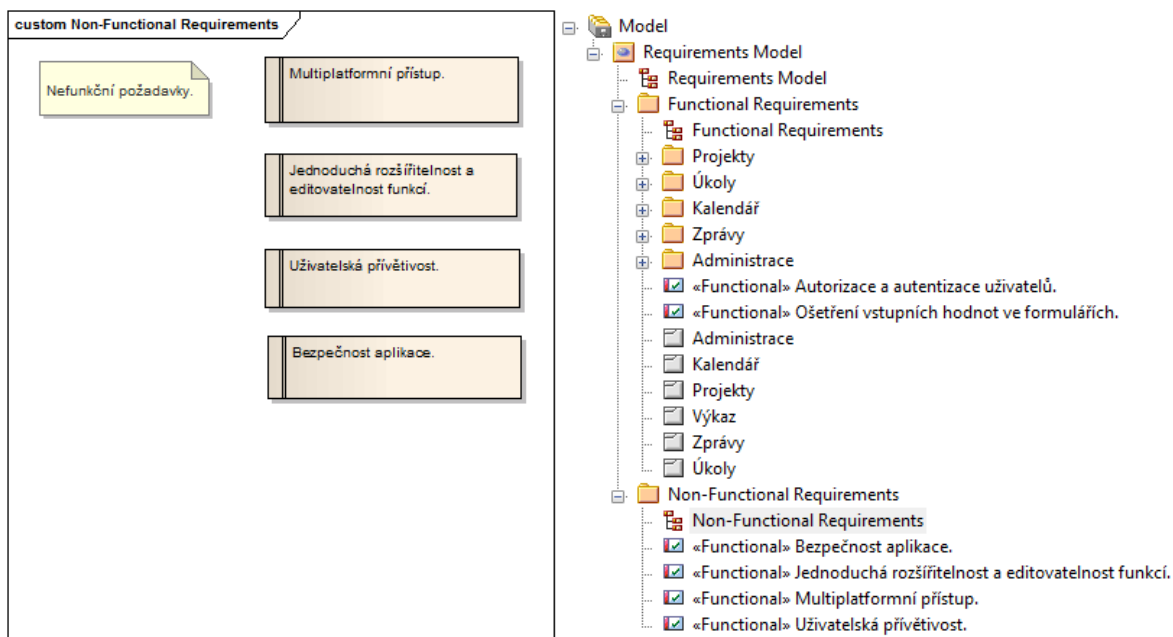
Obrázek 4: Zpracované funkční požadavky

Na obrázku číslo 4 je zobrazeno kompletní zpracování funkčních požadavků v programu Enterprise Architect. V detailu jednotlivých požadavků jsou doplněny i krátké rozšiřující popisy. Obdobným způsobem jsou zpracovány i nefunkční požadavky.

4.1.2 Nefunkční požadavky

Tyto požadavky popisují vlastnosti nebo omezující podmínky daného systému.

- Multiplatformní přístup
- Jednoduchá rozšiřitelnost a editovatelnost funkcí
- Uživatelská přívětivost
- Bezpečnost aplikace



Obrázek 5: Zpracované nefunkční požadavky s adresářovou strukturou požadavků

4.2 Vybrané technologie pro návrh a implementaci systému

Tato kapitola je zaměřena na detailní popis vybraných implementačních nástrojů, technologií a postupů využitých pro implementaci vlastní aplikace, které byly vybrány na základě vlastních zkušeností a znalostí. Přihlédnuto bylo samozřejmě také k míře dostupnosti těchto prostředků

4.2.1 Nástroj pro systémovou analýzu a návrh Enterprise Architect

Enterprise Architect je nástrojem, jenž slouží pro tvorbu systémových modelů. Založen je na syntaxi jazyka UML. Disponuje kvalitním a výkonným prostředím, které umožňuje specifikaci požadavků, modelování, návrh architektury a v neposlední řadě také systémovou analýzu.

Přední vlastností tohoto produktu je podpora vývoje. Ta umožňuje vytvořený návrh architektury aplikace vygenerovat do vybraného programovacího jazyka, jako je například cSharp, Visual Basic, C++ , PHP a další. Disponuje také synchronizací s již existujícím zdrojovým kódem. Vytvořené diagramy je možné uložit do několika formátů - RTF, XML a XMI. [11]

Tento nástroj jsem vybral na základě vlastních znalostí, zkušeností a dostupnosti software, která je díky aktivnímu využívání na Univerzitě Pardubice umožněna i v plné verzi. Nástroj jsem využil k analýze systému v několika fázích popisu v UML. Konkrétně při sběru požadavků, tvorbě Use Case diagramů a modelování návrhových tříd.

4.2.2 Nástroj pro grafické modelování a správu databáze MySQL Workbench

Díky tomuto nástroji lze model databáze vytvořit a následně jej jednoduše synchronizovat se schématem. Postup je možné provést i v obráceném pořadí. Model databáze můžeme nahrát z existujícího schématu. Nechybí ani generátor SQL kódu z vytvořeného modelu.

Aplikace nabízí mnoho funkcí pro práci s databází, jako například:

- tvorbu a editaci tabulek,
- práci s daty tabulek,
- správu oprávnění přístupu k databázi,
- import a export dat,
- tvorba databázových schémat.

Nástroje bylo v rámci tvorby ER Diagramu využito k vytvoření tabulek, atributů a jejich datových typů, definování primárních klíčů, unikátnosti hodnot v tabulkách a integritních omezení.

4.2.3 Vývojové prostředí NetBeans IDE

Tento nástroj je výrazným pomocníkem každého programátora při vývoji aplikací, které lze jeho prostřednictvím možné psát, překládat, ladit a distribuovat. Podporuje velké množství programovacích jazyků, jako například PHP, Java, C/C++ a mnohé další. Disponuje celou řadou rozšiřujících modulů, které zlepšují práci s vybraným programovacím jazykem.[12]

Nástroj zlepšuje orientaci v rozsáhlých zdrojových kódech a pomáhá v odhalování případných chyb. Další velkou výhodou je zjednodušení práce s frameworky. Tato podpora je jedním z hlavních důvodů zvolení právě tohoto nástroje. Aplikace vytvořená v rámci

diplomové práce je implementována s podporou Nette Frameworku. Práce s ním se tak stává efektivnější.

4.2.4 Balík předkonfigurovaných instalací XAMPP Lite

Pro realizaci vlastní aplikace v lokálním prostředí počítače je potřeba zprovoznit vlastní webový server s podporou PHP a databázový server. V dnešní době již není nutné instalovat a nastavovat všechny tyto části samostatně. Existují kvalitní aplikace, které obsahují všechny potřebné komponenty s jednotným prostředím pro jednoduchou administraci.

XAMPP Lite je balíček, pomocí kterého lze snadno a rychle zprovoznit domácí server. Toto ocení zejména ti, kdo potřebují testovat redakční systém nebo své webové prezentace v PHP.

Obsahem XAMPP Lite je:

- MySQL databáze,
- server Apache s PHP.

4.2.5 Databáze MySQL

MySQL je multiplatformní databáze. Pro komunikaci s ní se využívá níže popsany jazyk SQL, který je obdobně jako u ostatních relačních databází doplněn o některá specifická rozšíření.

Tuto databázi je možné nainstalovat na operačním systému Linux, MS Windows, ale i dalších operačních systémech. Jedná se o volně šiřitelný software a řadí se mezi nejpoužívanější databázové systémy. S oblibou je využíván na serverech v kombinaci Linux, MySQL, PHP a Apache. Toto spojení je souhrnně označováno jako technologie LAMP. [13]

Výběr databáze byl primárně podřízen možnostem praktického využití při nasazení. V dnešní době je velmi těžké najít hostingový server, který nepodporuje tuto databázi. Podpora bývá nabízena dokonce jako bezplatná služba. Další motivací pro výběr byla její integrace ve výše popsaném balíčku XAMPP Lite. Díky tomu byl umožněn bezproblémový vývoj na lokálním počítači.

4.2.6 Jazyk SQL

SQL - Structured Query Language (Strukturovaný dotazovací jazyk) je obecný dotazovací jazyk pro manipulaci, správu a organizování dat uložených v databázi.

„SQL patří mezi tzv. deklarativní programovací jazyky, což v praxi znamená, že kód jazyka SQL nepíšeme v žádném samostatném programu (jako by tomu bylo např. u jazyka C nebo Pascal), ale vkládáme jej do jiného programovacího jazyka, který je již procedurální. Se samotným jazykem SQL můžeme pracovat pouze v případě, že se terminálem připojíme na SQL server a na příkazový řádek bychom zadávali přímo příkazy jazyka SQL.“[14]

Příkazy jazyka SQL se dělí do několika skupin.

- DDL (Data definition language)
 - Do této skupiny se řadí příkazy určené pro práci se strukturou databázových objektů, a to nejčastěji tabulek.
- DML (Data manipulation language)
 - Tyto příkazy jsou určené pro manipulaci s daty.
- DCL (Data control language)
 - Příkazy této skupiny slouží k přidání či odebrání oprávnění k databázi a objektů v ní.
- TCL (Transactional control language)
 - Příkazy prezentující operace s transakcemi.[15]

4.2.7 Jazyk PHP

Tento jazyk byl vybrán vzhledem ke koncepci vytvářeného řešení, které má charakter webové aplikace. Ve spojení s níže popsaným webovým frameworkem tvoří velmi efektivní programovací techniku. Dalším faktorem výběru byla jeho znalost nabitá při studiu na Univerzitě Pardubice.

„PHP (rekurzivní zkratka PHP: Hypertext Preprocessor, „PHP: Hypertextový preprocesor“, původně Personal Home Page) je skriptovací programovací jazyk. Je určený především pro programování dynamických internetových stránek a webových aplikací například ve formátu HTML, XHTML či WML. PHP lze použít i k tvorbě

konzolových a desktopových aplikací. Pro desktopové použití existuje kompilovaná forma jazyka.“ [16]

Syntaxe tohoto jazyka vychází z několika programovacích jazyků (Perl, C, Pascal a Java) a je platformě nezávislý. Případné rozdíly v jednotlivých operačních systémech se vztahují pouze na několik systémově závislých funkcí. Přenos mezi nimi je tedy většinou bezproblémový.

Principem jeho využití v dynamických stránkách je provedení požadavku na straně webového serveru. Uživateli je prezentován jeho výstup. PHP nezaostává ani v podpoře nejrozličnějších knihoven pro speciální účely, jako je například zpracování textu, grafika nebo práce se soubory. Především také disponuje knihovnami pro přístup k většině využívaných databázových systémů (MySQL, ODBC, Oracle, PostgreSQL, MSSQL).

PHP je nejrozšířenějším skriptovacím jazykem pro web. Oblíbeným se stal především díky jednoduchosti použití, bohaté zásobě funkcí.

4.2.8 Nástroj pro vytváření webových aplikací Nette Framework

„Nette Framework je open source framework pro tvorbu webových aplikací v PHP 5. Zaměřuje se na eliminaci bezpečnostních rizik, podporuje AJAX, DRY, KISS, MVC a znovupoužitelnost kódu. Využívá událostmi řízené programování a z velké části je založen na použití komponent.“ [17]

Zajímavostí tohoto frameworku je jeho český původ. V mezinárodním měřítku se neřadí mezi nejpoužívanější, ale v rámci České republiky kolem Nette Frameworku vznikla poměrně početná komunita, což ukazuje na jeho velké rozšíření. Autorem je David Grundl, který svůj záměr plně tlumočí uživatelům frameworku a na základě vzniklých diskusí jej dále rozšiřuje. Nejedná se tedy zcela o, jak by se na první pohled mohlo zdát, výtvar pouze jedné osoby. V současné době je již k dispozici 4. verze, která nese označení „2.0“, a na další se usilovně pracuje.

Co do struktury má Nette jednu hlavní třídu s názvem Nette\Object. Ta je společná všem dalším třídám. Nette\Object můžeme chápat jako rozšíření standardního PHP jazyka, které nám například usnadňuje i přístup k sebereflexi třídy pomocí metody getReflection().

Reflexe nám umožňuje zjistit informace typu: jaké metody má daná třída, jaké mají parametry atd.

```
$circle = new Circle;  
echo $circle->getReflection()->hasMethod('getArea'); // existuje metoda 'getArea' ?  
echo $circle->getReflection()->getName(); // vrací název třídy, tj. 'Circle'
```

Další schopností této třídy je vyvolání výjimky při přístupu do nedeklarovaného člena třídy.

```
$circle = new Circle;  
echo $circle->undeclared; // upozorní výjimkou typu Nette\MemberAccessException  
$circle->undeclared = 1; // upozorní výjimkou typu Nette\MemberAccessException  
$circle->unknownMethod(); // také upozorní výjimkou typu Nette\MemberAccessException
```

V drtivé většině případů se jedná především o překlepy, které bývají obtížně odhalitelné. Objekty, které jsou potomky této třídy, při přístupu k nedeklarovanému členu vždy vyvolávají výjimku typu `MemberAccessException`.

Poslední schopnost, kterou bych rád zmínil, se skrývá pod termínem property označujícím speciální členy tříd. S těmito členy lze pracovat jako s proměnnými, ale ve skutečnosti se jedná o metody. U `Nette\Object` jde konkrétně o tzv. getter nebo setter. V případě přiřazení nebo čtení hodnoty se volá příslušná metoda. Díky této vlastnosti můžeme validovat vstupy nebo generovat výstupy až v případě potřeby.

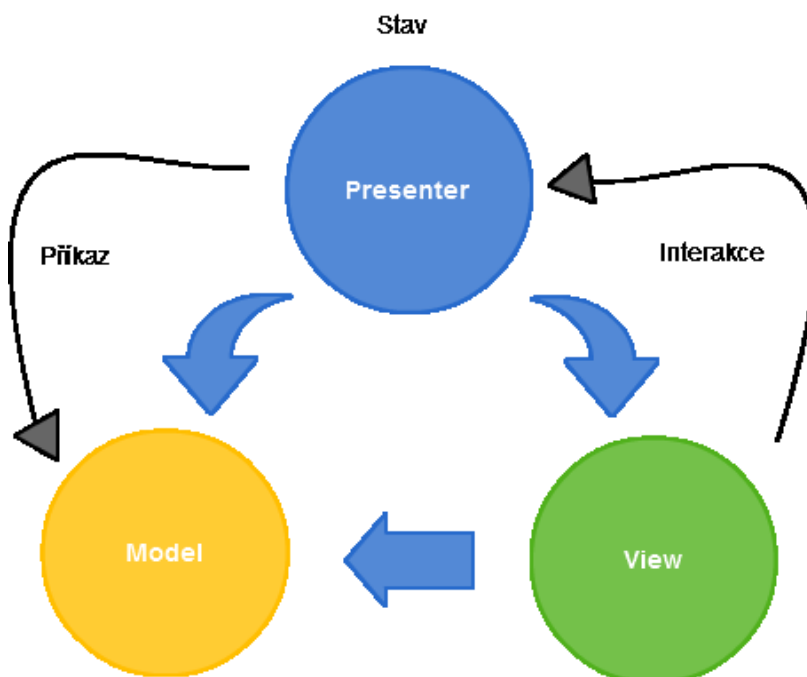
```
// klasický způsob pomocí volání metod  
$circle->setRadius(10); // nastav poloměr kruhu  
echo $circle->getArea(); // vypiš obsah kruhu  
  
// ekvivalentní řešení s využitím properties  
$circle->radius = 10; // ve skutečnosti volá setRadius()  
echo $circle->area; // volá getArea()
```

Každá třída, která dědí z třídy *Nette\Object* má tyto vlastnosti. Vhodné tedy je z této třídy dědit i při tvorbě vlastních tříd. Je důležité však dodržet následující jednoduchou konvenci:

- „*Getter i setter musí být public metoda.*
- *Getter má název ve tvaru getXyz() nebo isXyz(), setter má název ve tvaru setXyz()*
- *Getter i setter je volitelný, mohou tedy existovat read-only nebo write-only property*
- *Názvy property jsou citlivé na velikost písmen (case-sensitive), s výjimkou prvního písmene, které může být velké i malé.*“[18]

Používání různých softwarových architektur je u webových aplikací běžnou vlastností. Nejrozšířenější je MVC model. Nette však využívá trochu odlišný, méně populární přístup. Jedná se o model MPV, který je zkráceninou slov Model View Presenter. Zjednodušeně lze říci, že presenter v Nette je obdobou controlleru. Presenter v Nette hraje čistě roli prostředníka, který jen volá model a výsledky předává view, controller má navíc na starosti i některé události uživatelského rozhraní.

Adresářová struktura je v Nette dle tohoto konceptu již připravená, a vede tak programátory k čistému návrhu aplikace s důrazem na budoucí rozšiřitelnost.



Obrázek 6: Schéma softwarové architektury MVP

Nette myslí také na bezpečnost naprogramovaných aplikací. Obsahuje totiž postupy, kterými snižuje riziko bezpečnostních děr. Zabraňuje napadení aplikace např. přes XSS, CSRF nebo session fixation. Pro obranu před XSS disponuje technologií Context-aware escaping, která vstupy ošetřuje automaticky. Programátor se nemusí tedy zabývat kontrolou všech řetězců. Ošetření navíc probíhá jak pro HTML, tak i JavaScript.

Proti CSRF bojuje Nette generováním a následným ověřováním autorizačních tokenů. U formuláře se tato ochrana aktivuje voláním metody `addProtection()`.

Proti session fixation se lze účinně bránit správnou konfigurací webového serveru a PHP. Nette to je schopné v případě povolení přístupu serverem nakonfigurovat automaticky.

Velkým pomocníkem pro vývoj a debugování, který je také standardní součástí, je tzv. laděnka. Tento nástroj přehledně prezentuje chyby a další ladicí informace přímo v prohlížeči při spuštění aplikace. Na produkčním serveru ale není pro běžného uživatele takové vypisování chyb, spolu s ladicími informacemi, žádoucí. Nette však nabízí možnost přepnutí aplikace do produkčního režimu, v němž nejsou tato chybová hlášení uživateli zobrazena a jsou naopak zaznamenávána pouze do souboru.

Nette jako celek představuje promyšlený a čistý objektový návrh využívající nových vlastností PHP 5, komponent a událostmi řízeného modelování. Podporuje AJAX i SEO prvky, jako jsou „hezké URL“. Velkou výhodou je pak početná komunita, která je velmi aktivní. Obratně a ochotně podává odpovědi na případné komplikace, které se při získávání dovedností mohou vyskytnout.

4.2.9 Jazyk UML

„Jazyk UML (Unified Modeling Language) je univerzálním grafickým jazykem určeným pro vizuální modelování systémů. Je velmi oblíben i díky tomu, že ho lze standardně používat nejen při modelování objektově orientovaných systémů. Jazyk přímo nepatří k žádné konkrétní metodice ani k životnímu cyklu. Struktura vizuálního jazyka se skládá ze tří částí – stavební bloky, společenské mechanismy a architektura.“ [19]

Jazyk UML byl navržen jako univerzální standard pro záznam, konstrukci, vizualizaci a dokumentaci artefaktů systémů s převážně softwarovou charakteristikou.

Cílem bylo spojit nejlepší existující postupy modelovacích technik a softwarového inženýrství. UML nám tak umožňuje navrhovat business procesy, systémové funkce, ale i programové komponenty. Výsledné návrhy je možné díky standardizaci sdílet s ostatními vývojáři.

Stavební bloky tohoto jazyka se skládají ze tří částí.

- *„Z předmětů (things), což jsou samostatné prvky modelu,*
- *vztahů (relationships), jež je pojítkem mezi předměty (relace určují, jak spolu dva nebo více předmětů významově souvisí),*
- *a diagramů (diagrams), což jsou pohledy na modely UML; ukazují kolekce předmětů, které „vyprávějí příběh“ o softwarovém systému a jsou naším způsobem vizualizace, analysis-level diagrams), a toho, jak to bude dělat (návrhové diagramy, design-level diagrams).“ [19]*

Základním předpokladem jazyka UML je umožnění modelování nejrozličnějších systémů jako soubor spolupracujících objektů. Týká se to jednak standardních objektově orientovaných systémů, tak i obchodních systémů. Existují dva pohledy na model UML.

- *„Statická struktura – popisuje typy objektů, jaké jsou pro modelování systému důležité.*
- *Dynamická struktura – popisuje životní cyklus objektů a jejich vzájemnou spolupráci, ze kterých se namodelovaný systém skládá.“ [19]*

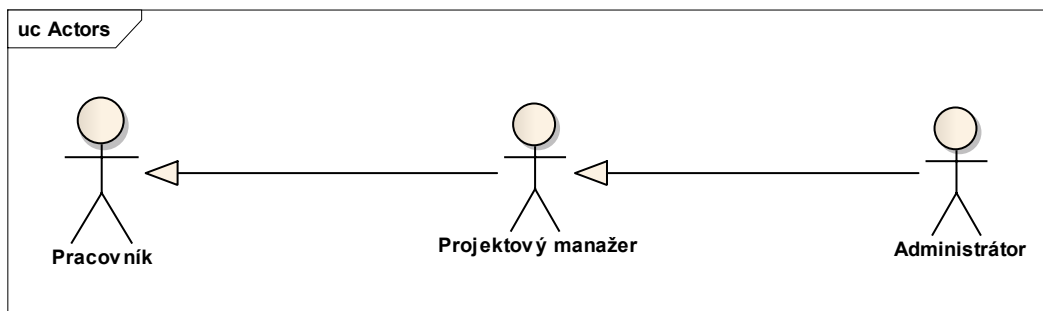
Důležité je uvědomit si, že jazyk UML není žádný druh metodiky modelování, nýbrž poskytuje pouze vizuální syntaxi, kterou využíváme při popisu svých modelů nebo programových komponent.

4.3 Návrh Use-Case diagramů

Dalším krokem návrhu systému je namodelování pomocí Use-Case diagramů se scénáři. V prvním kroku jsou určeni aktéři. V této práci jsem dle uživatelů systému nadefinoval 3 aktéry:

- pracovník,
- projektový manažer,
- administrátor.

Pracovník má k dispozici nástroje pro práci s kalendářem, zprávami a některými funkcemi úkolů pro report jejich stavu. Projektový manažer má stejná práva jako pracovník, dále je schopen pracovat také s projekty a všemi funkcemi úkolů. Administrátor má přístupová práva ke všem funkcím systému.



Obrázek 7: Aktéři systému

Pro přehlednost jsou vytvořeny Use-Case diagramy k jednotlivým funkčním celkům, jejichž dělení je naznačeno výše při definici požadavků. Nad těmito celky je dále vytvořen jeden obecný model, kde jsou naznačena přístupová práva k jednotlivým celkům. Ten lze nalézt v elektronické příloze práce.

4.3.1 Use-Case digram Projekty

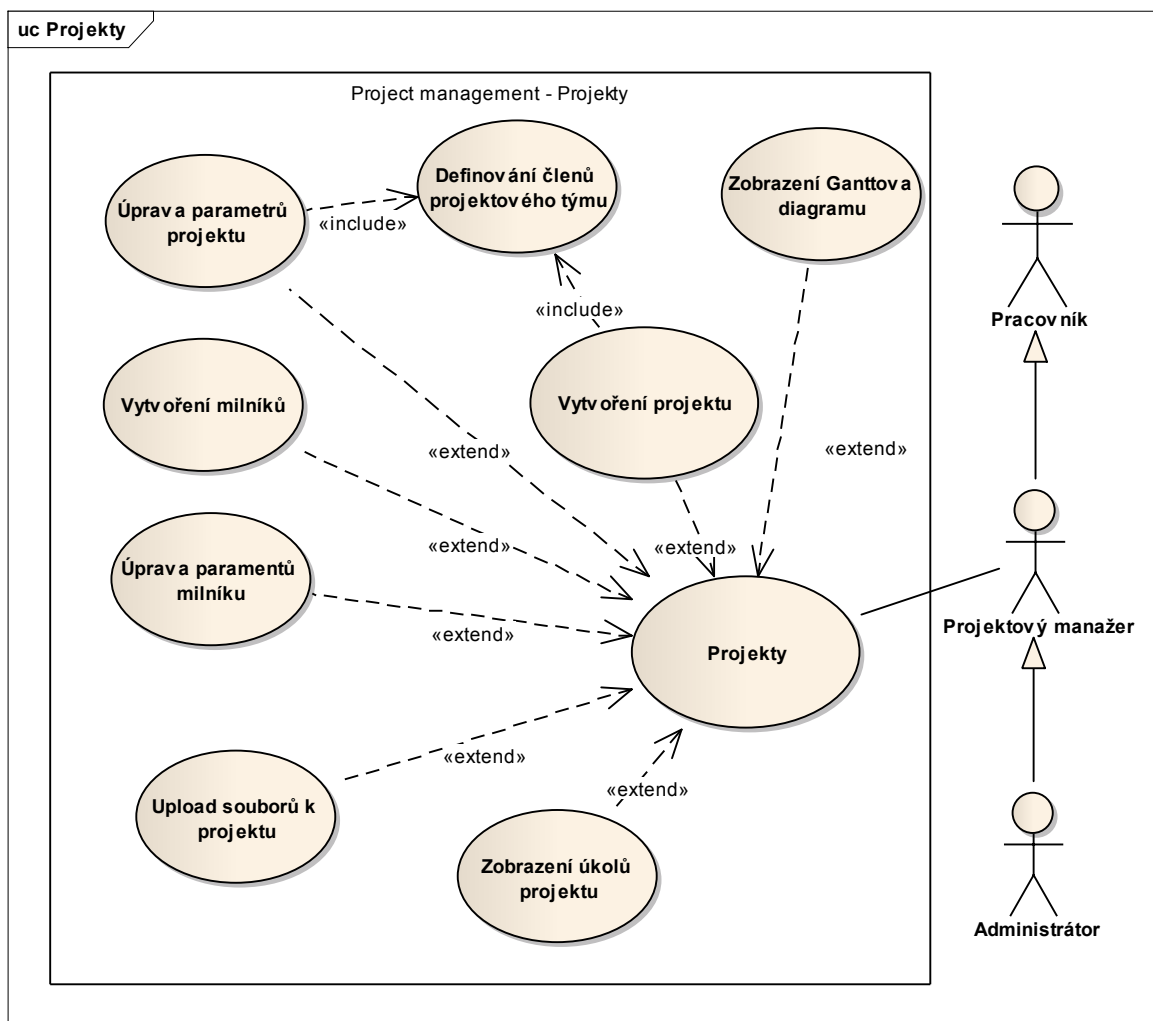
Tento Use-Case diagram zachycuje strukturu celku pracujícího se základním prvkem projektového řízení – projektem. Diagram ukazuje jednotlivé funkcionality, které umožňují manipulaci s projektem a jeho zobrazení. Také prezentuje prvky pro určení členů projektového týmu a komunikaci mezi nimi.

Důležitým vizuálním prvkem je Ganttův diagram, který ukazuje časovou náročnost a návaznost úkolů projektu. Podstatnou měrou tak zvyšuje orientaci v projektu. Projektový manažer snadno zjistí, na jakých úkolech se v daný den má pracovat. Z tohoto digramu lze přímo vstoupit do detailu jednotlivých úkolů. Pro rychlý přehled je také k dispozici tabulkový výpis úkolů.

Úkoly v rámci projektů je vhodné dělit do milníků, které stanovují jednotlivé vývojové fáze projektu. Pro podporu této funkce slouží vytvoření milníků společně s jejich případnou editací.

Vytvoření projektu zastupuje definici základních paramentů, jako je název, termín zahájení i ukončení a krátkým popis. Tato funkcionality dále využívá část pro určení členů projektového týmu. Jedná se o uživatele, kteří mají přístup k informacím o projektu a je jim možné přiřadit úkoly. Takto vytvořený projekt musí disponovat také možností editace.

Pro přenos souborů vztahujících se k projektu je k dispozici upload souborů k projektu.

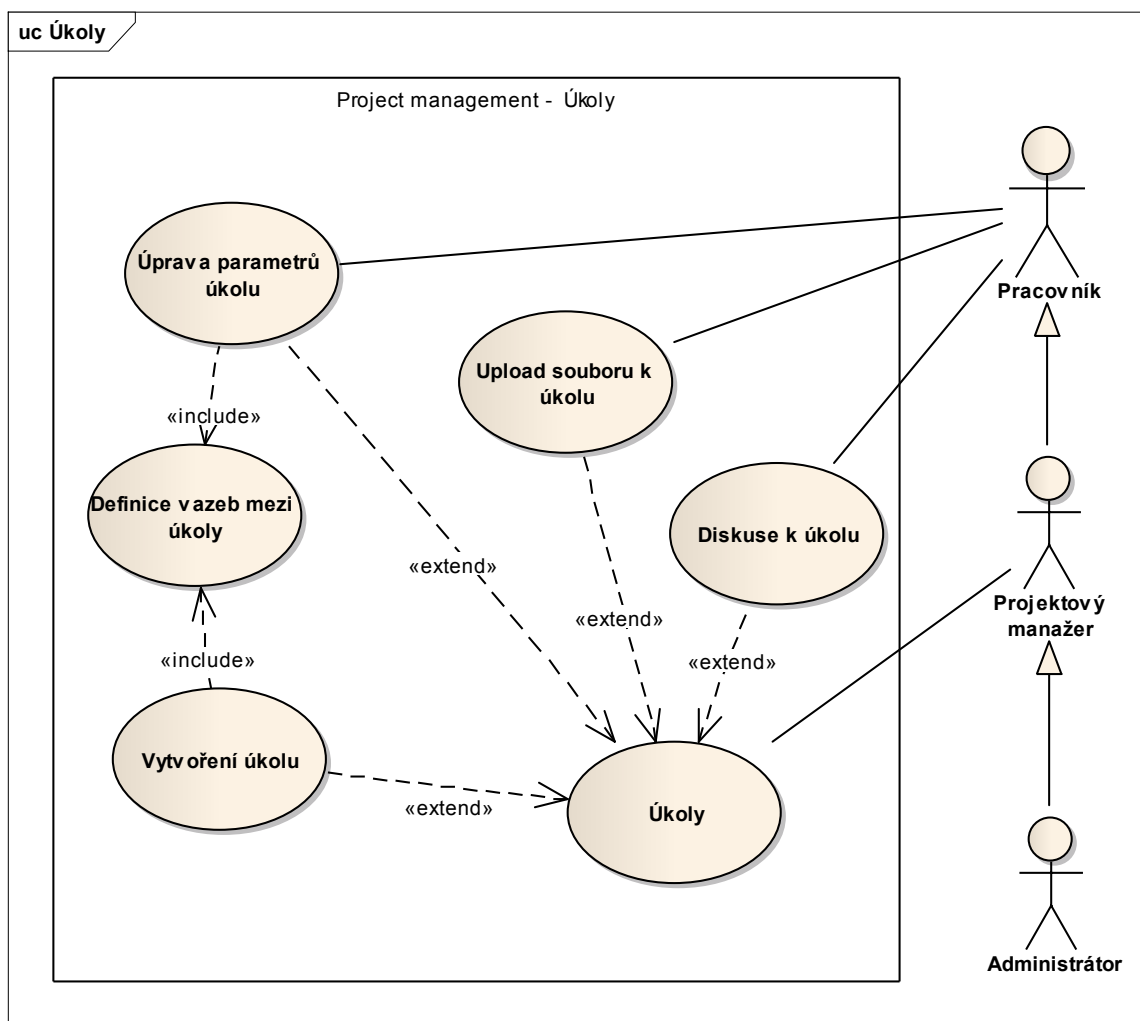


Obrázek 8: Use-Case diagram Projekty

4.3.1 Use-Case digram Úkoly

Use-Case diagram tohoto modulu popisuje práci s úkoly, které chápeme jako soubor činností nutných k realizaci každého projektu. Práva na vytváření úkolů má projektový manažer, který určuje co je obsahem úkolu, jaké jsou jeho parametry a kdo jej

bude vykonávat. Dále může stanovit vazby mezi úkoly projektu ve smyslu jejich zahájení a ukončení. Tím se rozumí určení podmínek, za jakých může být úkol zahájen nebo ukončen v návaznosti na ostatní úkoly projektu. Modul umožňuje projektovému manažerovi vytvořený úkol editovat. Stejně jako je tomu u projektu, je možné i úkol doplnit o upload souboru.

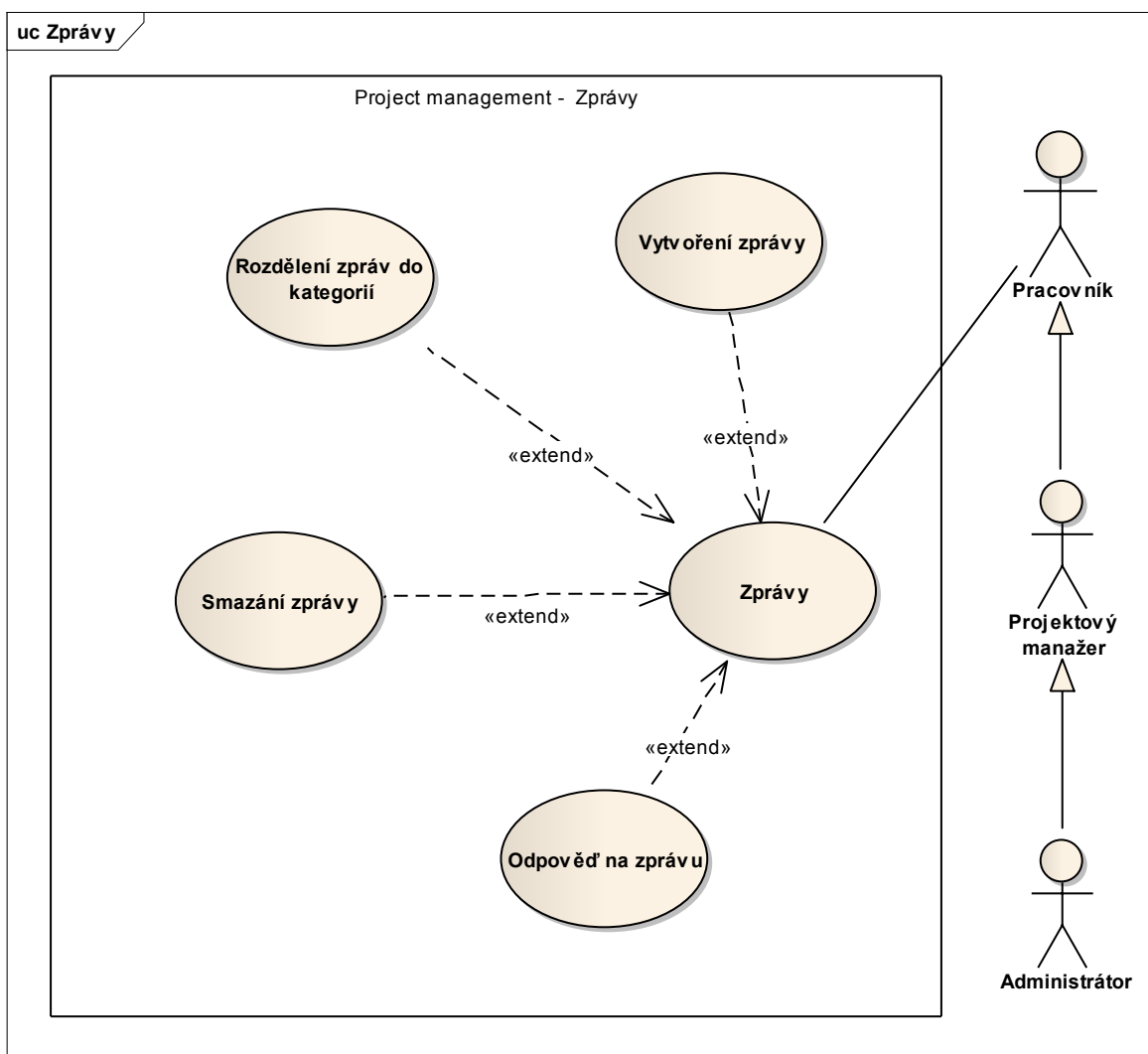


Obrázek 9: Use-Case diagram Úkoly

Pracovník úkoly nevytváří, ale na základě přiřazeného úkolu realizuje kroky vedoucí k jeho úspěšnému dokončení. Ty reportuje úpravou stavu úkolu. Pro případné komplikace, nápady nebo další události, které je nutné konzultovat či zaznamenat, může pracovník využít poznámku nebo diskusi, která je s každým úkolem svázána. Výstup práce je možné odevzdat opět pomocí upload souborů.

4.3.1 Use-Case diagram Zprávy

Zprávy umožňují komunikaci mezi členy projektových týmů, a to na bázi krátkých textových zpráv. Každý uživatel má ve své schránce složku pro doručené zprávy od svých kolegů, pro odchozí zprávy a pro smazané zprávy. Sám má možnost vytvořit zprávu a určit, kterým uživatelům bude odeslána. Na doručené zprávy lze přímo zareagovat odpovědí odesílatelům zprávy.



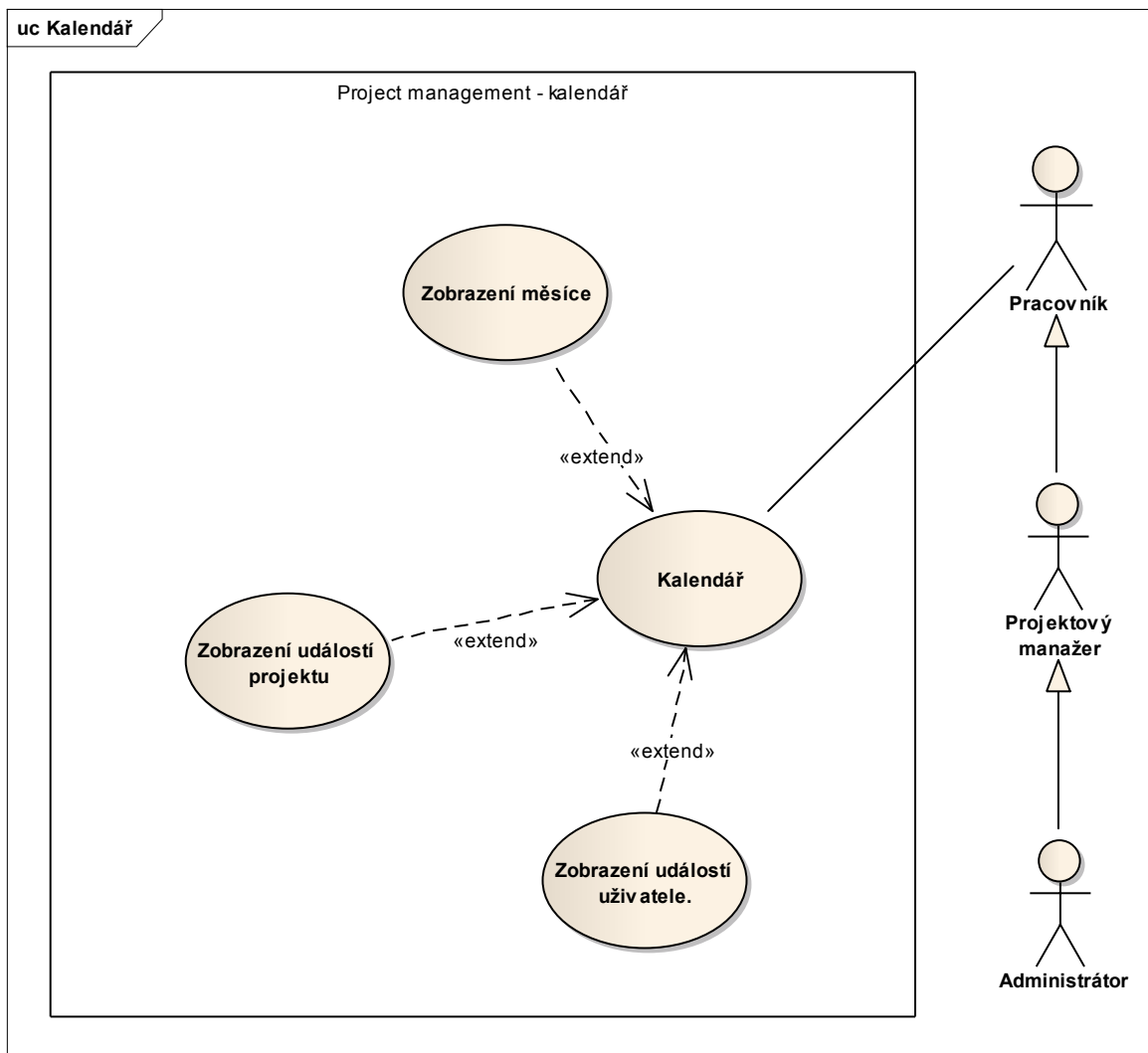
Obrázek 10: Use-Case diagram Zprávy

4.3.1 Use-Case diagram Kalendář

Kalendář představuje nástroj pro časovou orientaci nad projektovými událostmi. Jeho rozhraní disponuje zobrazením měsíčního kalendáře se všemi dny aktuálního nebo vybraného měsíce.

Kalendář uživatele graficky vykresluje všechny události projektu do jednotlivých dní. Těmi jsou myšleny termíny projektů, úkolů a milníků. Podmínkou pro jejich zobrazení je členství uživatele v projektovém týmu. Pro projektový kalendář je tato funkce analogická s tím rozdílem, že v kalendáři figuruji události pouze vybraného projektu.

Přístup ke svému osobnímu kalendáři má každý uživatel. Do projektového pouze člen daného projektu.



Obrázek 11: Use-Case diagram Kalendář

4.3.2 Use-Case diagram Administrace

Tento Use-Case diagram zachycuje prvky, které se starají o administraci všech informací v systému. Jeho plnou podobu naleznete v elektronické příloze této práce. Jako celek jej lze rozdělit do několika částí zaštiťujících administraci vybraných dat systému.

Konkrétně jsou to tyto:

- projekty,
- úkoly,
- milníky,
- soubory,
- zprávy,
- uživatelé,
- oprávnění.

Část projekty umožňuje úpravu všech existujících projektů včetně jejich úplného odstranění ze systému. V této části je dále k dispozici definice projektových rolí v systému spojená s úpravou, odstraněním a přiřazením oprávnění v rámci projektu.

V části úkoly jsou obdobně prezentovány všechny existující úkoly v systému s možností jejich editace. Dalším celkem je pak úprava číselníků využívaných u úkolů. Konkrétně se jedná o priority, stavy a typy úkolů. Tyto číselníky lze přizpůsobit dle potřeby.

Soubory dávají přehled o všech aktuálních souborech u projektů a úkolů. Možné je zobrazit jejich detaily a dle uvážení je ze systému odstranit. Zprávy jsou co do funkčnosti analogické. Z jejich přehledu lze vstoupit do detailů zprávy nebo ji případně odstranit.

V sekci uživatelé se spravují všichni aktivní a neaktivní uživatelé systému. K dispozici je přidání uživatele, k němuž slouží jednoduchý formulář pro vyplnění základních informací o uživateli včetně přiřazení role v systému. Jeden uživatel bude disponovat vždy jen jednou rolí v systému. Pro práci s uživateli jsou k dispozici základní operace – editace, mazání.

Poslední část oprávnění je velmi podobná definici rolí v projektu, ale s tím rozdílem, že v této části určujeme role v systému, nikoliv v projektu.

4.4 Návrhový model tříd

Model tříd je prezentován diagramem tříd, jenž poskytuje statický pohled na třídy systému. Slouží jako předloha k naimplementování aplikace. Každá třída by měla

obsahovat výčet všech atributů s jejich názvy, datovými typy a přístupovými oprávněními. Dále výčet všech metod s jejich parametry a návratovými hodnotami.

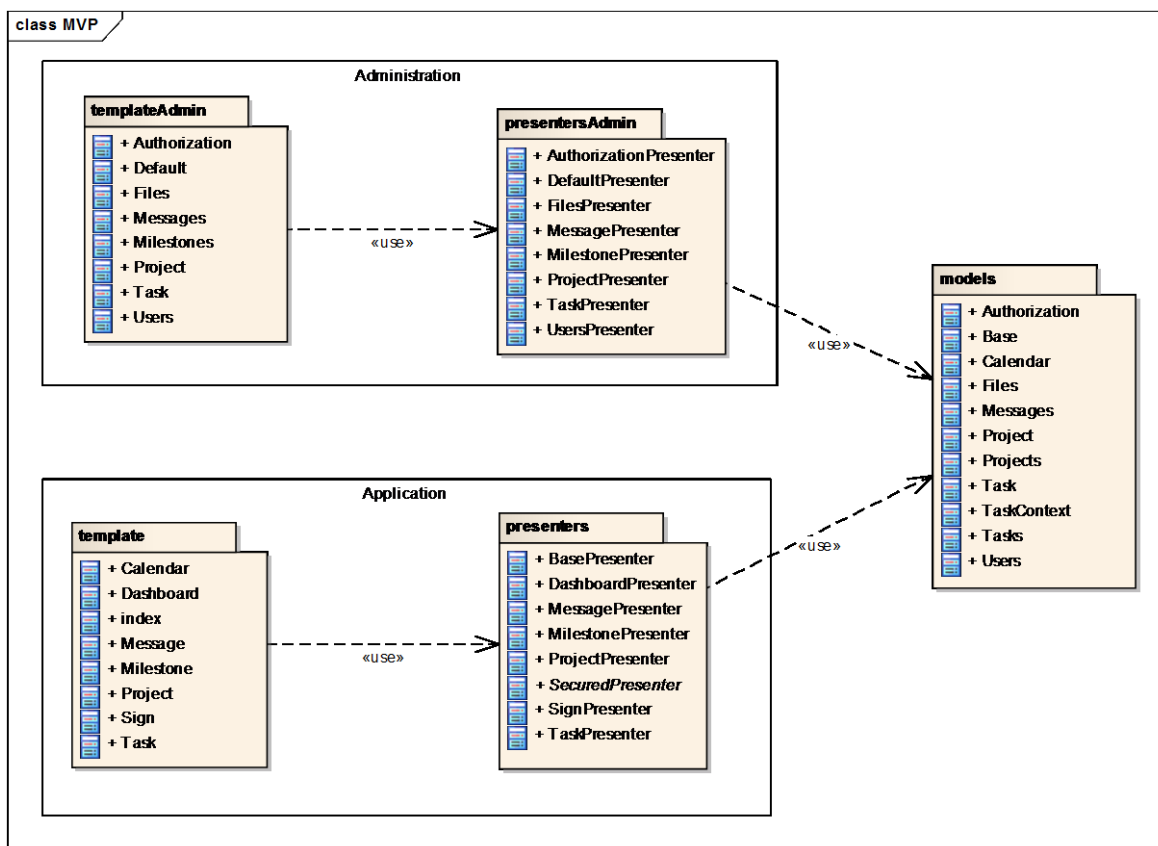
Vzhledem k tomu, že aplikace je od svého počátečního vývoje koncipována jako základní řešení s důrazem na snadnou rozšiřitelnost, jsou třídy navrženy v softwarové architektuře MVP. Tu využívá i vybraná technologie Nette Framework. Jedná se o softwarovou architekturu oddělující aplikační logiku, datové třídy a grafická rozhraní aplikace. Díky tomuto dělení má případná změna jedné vrstvy minimální dopad na ostatní vrstvy, výrazně se zvyšuje přehlednost a je umožněna paralelní spolupráce více vývojářů na jednom projektu.

Krátce lze dělení z pohledu návrhu popsat takto:

- model – nemá vědět o tom, že nějaké view a presentery existují;
- view – o modelu také nemusí vědět, anebo naopak může data získávat přímo z něj (dle zvolené koncepce);
- presenter – propojuje view s modelem (ne naopak) a realizuje uživatelské akce.

Vzhledem k požadavku na administraci existujících dat a číselníků v systému, bylo toto dělení návrhových tříd dále rozšířeno na další dvě části. První část zajišťuje administraci systému a druhá, uživatelská část, se stará o klasický běh aplikace. Každá z těchto částí obsahuje vlastní view a presenter určující specifické funkce každé části. Díky tomuto dělení se ještě více zpřehlednil zdrojový kód celého systému. Na obou částech se může v případě realizace vývojovým týmem pracovat zcela nezávisle.

Vyjmenování a popsání všech tříd společně s popisem jejich logiky je velmi rozsáhlou disciplínou, která by výrazně navýšila rozsah této diplomové práce. Veškeré grafické vizualizace navržených tříd včetně jejich detailů jsou k dispozici v její elektronické příloze. Ta obsahuje celkový pohled na navržený systém v architektuře MVP a také podrobněji rozpracované jednotlivé vrstvy architektury obou částí – administrace a aplikace. Pro představu o celkovém návrhu tříd je na obrázku č. 13 přiložen balíčkový model tříd v jednotlivých vrstvách.



Obrázek 12: Balíčkový model návrhových tříd

4.5 Tvorba datového modelu

Dalším krokem vývoje systému je vytvoření struktury databáze. Jedná se o velmi podstatnou část vývoje, při jejímž nesprávném provedení budou ukládaná data těžko dosažitelná. V horším případě budou znemožněny případné budoucí úpravy datového modelu.

Pro popis databázového modelu byl zvolen nejpoužívanější entitně-relační model, který graficky reprezentuje logické vazby mezi entitami za účelem vytvoření databáze. Založen je na třech základních modelovacích komponentech – entita, vztah a atribut.

Entitu lze popsat jako věc schopnou samostatné existence, která je jednoznačně identifikovatelná. Logický význam entit je ekvivalentní s gramatickými podstatnými jmény, jako jsou například uživatelé, projekty, úkoly nebo zprávy.

Vztah popisuje, jakým způsobem jsou entity mezi sebou provázány. Jejich gramatický ekvivalent je popsán slovesy nebo případně slovními spojeními, jako například

mít nějaký stav, být členem nějaké role, anebo mít nějaké zprávy. Vztah může být dále popsán počtem entit, které do vztahu vstupují. Toto číslo udává stupeň kardinality vazby. Ta může nabývat hodnot 1:1, 1:N nebo M:N.

Atribut je funkce, která vyjadřuje vlastnost nebo charakteristiku dané entity. Příkladem je třeba entita uživatel. Ta může obsahovat atributy: jmeno, příjmení nebo login. Atributů musí entita obsahovat minimálně tolik, aby ji pomocí nich bylo možné jednoznačně identifikovat. Atributy, které přesně vymezují entitu, jsou pak nazývány primárním klíčem.

Struktura databáze je v entitně-relačním modelování souhrnně prezentována jako ER-diagram, kde mohou být entity vyznačeny body, mnohoúhelníky nebo kruhy. Pro každou entitu existuje v databázi ekvivalentní relační tabulka. Platí to i naopak, přesná podoba grafických značek je dána zvolenou anotací.

K tvorbě modelu databáze byl využit nástroj MySQL Workbench, který umožňuje kompletní návrh databáze pomocí výše popsaného entitně-relačního modelu a jeho následné vygenerování do DDL skriptů pro databázi MySQL. Velmi povedenou funkcí, která byla při vývoji aplikace hojně využívána, je synchronizace změn modelu přímo s databázovým schématem. Tato funkce pracuje obousměrně. Změny tedy bylo možné přenášet jak směrem z modelu databáze do jeho funkčního schématu, tak i opačně. Značně tak byly urychleny případné úpravy databáze, které vyvstaly při programování aplikace.

Celkem bylo pro vlastní řešení vytvořeno 21 entit. Kompletní ER-diagram je zachycen v příloze A této diplomové práce. Ten byl navržen tak, aby splňoval 3. normální formu. Tato forma říká, jak vytvořit strukturu databáze, aby byla co nejefektivněji využitelná. Model databáze je ve 3. normální formě v případě, že splňuje 2. a 1. normální formu.

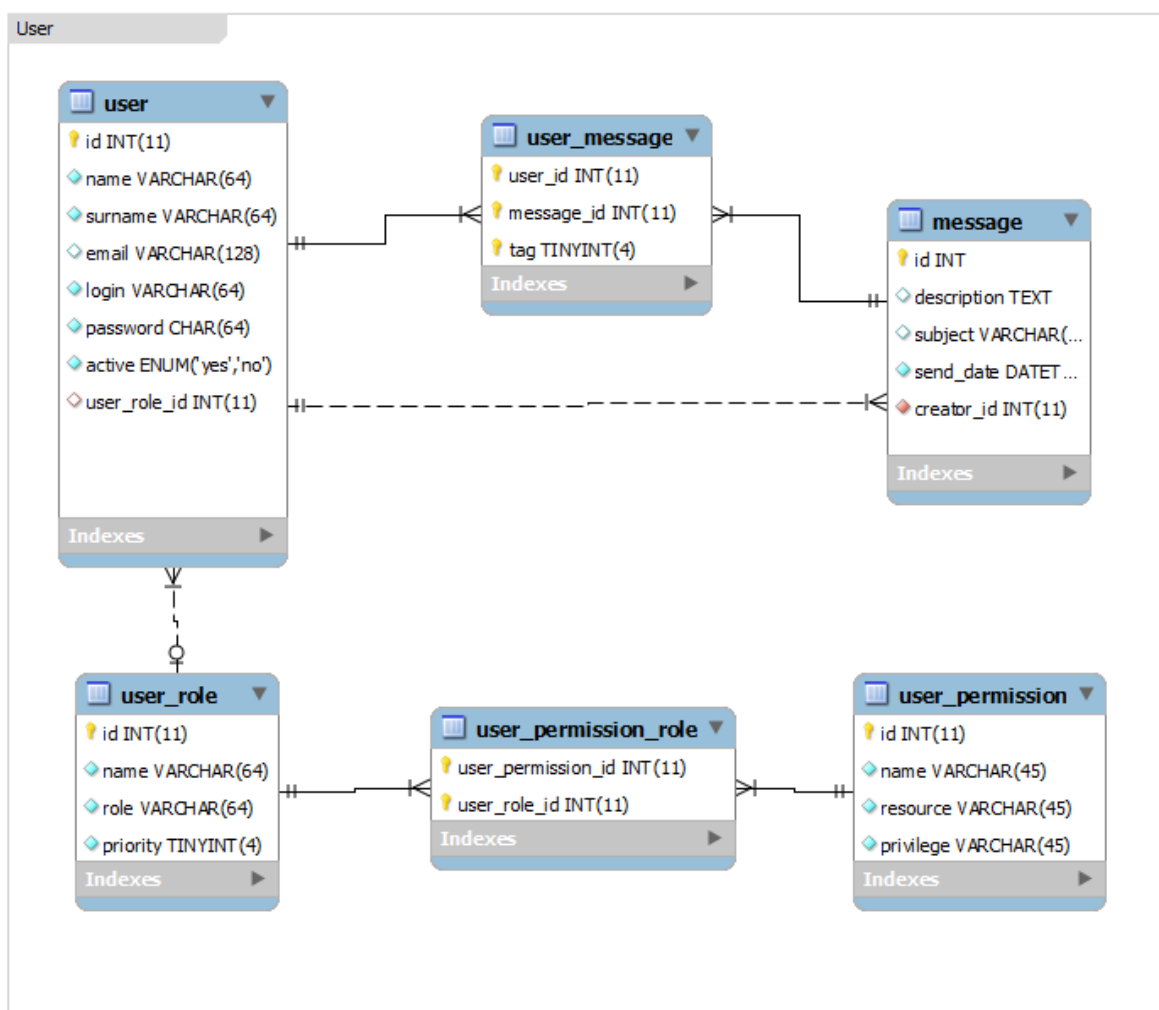
Model v 3. normální formě musí naplňovat tyto podmínky:

- „každý její atribut obsahuje jen atomické hodnoty,
- nekľíčový atribut je plně závislý na primárním klíči, a to na celém klíči a nejen na nějaké jeho podmnožině,
- nekľíčové atributy jsou vzájemně nezávislé.“[20]

Na vytvořeném modelu je možné provést rozpad do 3 částí, které jsou níže podrobně popsány – uživatel, projekt, úkol.

4.5.1 Datový model - Uživatel

Model této části je zachycen na obrázku č. 14. Jejím obsahem jsou tabulky udržující informace o uživatelích systému, rolích a jejich oprávněních. Do této části jsou také zahrnuty tabulky pro podporu komunikace mezi uživateli - zprávy.



Obrázek 13: Datový model – Uživatel

Tabulku *user* lze označit za nejpodstatnější část návrhu datového modelu. Bez této tabulky by do systému neměl přístup žádný uživatel. Pro jeho přihlášení do systému jsou určeny atributy *login* a *password*. Další atributy slouží k identifikaci uživatele.

Tabulka *user* má vazbu na tabulku *user_role*, která slouží pro uchování uživatelských rolí v systému. Atribut *priority* zde slouží k určení pořadí rolí v systému. Podporuje se tak přehlednost a orientace v aplikaci. Tabulka *user_permission* určuje oprávnění rolí pro akce v systému. Atribut *privilege* definuje akci, jako je například upravit, zobrazit atd. Atribut *resource* říká, k jakému zdroji se akce vztahuje (například projekt, úkol atd.). Tabulky *user_role* a *user_permission* jsou propojeny vazbou M:N pomocí tabulky *user_permission_role*.

V tabulce *message* jsou obsaženy atributy pro uchování krátkých textových zpráv mezi uživateli systému. Atribut *creator_id* uchovává informaci o tvůrci zprávy. Atributy *subject*, *description* a *send_date* nesou základní data o zprávě samotné. Konkrétně předmět zprávy, vlastní obsah a čas odeslání. Vzhledem k tomu, že jedna zpráva může mít více adresátů, a stejně tak jeden uživatel může figurovat ve více zprávách, je realizována vazba M:N mezi tabulkou *user* a *message* pomocí tabulky *user_message*. Atribut *tag* je příznakem zařazení zprávy v kategorii uživatele (doručené, odeslaná, smazaná).

4.5.1 Datový model - Projekt

Část datového modelu – Projekt – je další podstatná část systému, která řeší základní kámen projektového řízení. Určuje entity samotného projektu a pomocí vazeb jeho další parametry jako jsou milníky, soubory, typy úkolů, projektový tým a role jednotlivých členů projektového týmu.

Tabulka *project* je speciální svojí vazbou sama na sebe v relaci 1:N, čímž je docíleno dělení projektu do dalších na sobě nezávislých podprojektů.

Dělení úkolů projektu do jednotlivých milníků je podpořeno tabulkou *milestone*. Ta obsahuje informace o zahájení a ukončení milníku. Dále pak pojmenování milníku a identifikaci k jakému projektu náleží.

Projektové zadání je možné doplnit o soubory, jako jsou například technické výkresy nebo schémata, která jsou nahraná přímo k projektu. Informace o takových souborech včetně jejich umístění jsou evidovány v tabulce *project_file*. Fyzicky budou soubory umístěny v ftp úložišti.

Tabulka *project_member* definuje členy projektového týmu. Tabulka *project_role*, *project_permission* a jejich vazba typu M:N realizovaná pomocí tabulky *project_permission_role* podporují projektová oprávnění jednotlivých členů týmu. Tato oprávnění jsou oddělena od oprávnění v celé aplikaci a umožňují určit roli každého uživatele dle potřeby v daném projektu. Uživatel s právy vedoucího v jednom projektu může mít práva obyčejného pracovníka v jiném projektu.

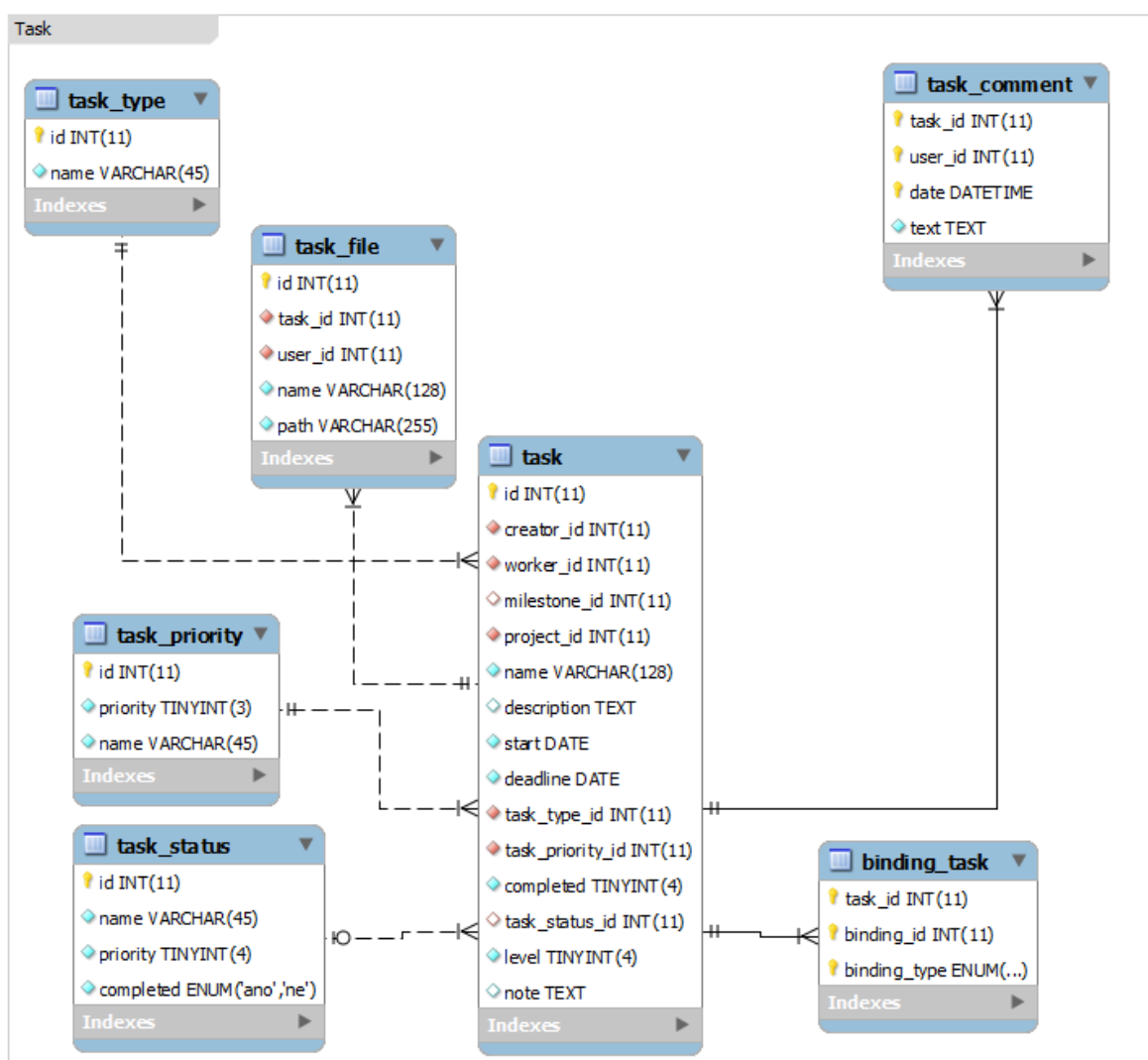


4.5.1 Datový model – Úkol

Poslední část datového modelu je Úkol. Tato část je v porovnání s předchozími modely nejrozsáhlejší.

Nepodstatnější entitou v této části je *task*, který prezentuje základní informace o samotném úkolu. Tím jsou myšleny informace například o tvůrci úkolu; realizátorovi úkolu; milníku, do kterého spadá; případně o projektu, jehož je podprojektem; typu; stavu a prioritě úkolu atd.

Zadání úkolu je možné, obdobně jako u projektu, doplnit o nahrání souboru pomocí entity *task_file*.



Obrázek 15: Datový model – Úkol

Speciální entitou této části je *binding_task*. Ta určuje typy vazeb mezi úkoly jednoho projektu. Je tak podpořena případná návaznost jednotlivých úkolů.

Vazby mezi úkoly mohou nabývat celkem 4 stavy:

- daný úkol může začít, až další určený skončí;
- daný úkol může skončit, až další určený skončí;
- daný úkol může začít, až další určený začne;
- daný úkol může skončit, až další určený začne.

Entity *task_type*, *task_priority* a *task_status* přesněji popisují další vlastnosti úkolu. Typ úkolu je jeho členění do vybraných typů úkolů projektu, které jsou definovány při vytvoření projektu. Priorita je příznak důležitosti pro realizátora úkolu a stav úkolu říká, co se s úkolem aktuálně děje, tedy zda čeká, je konzultován, dokončen atd.

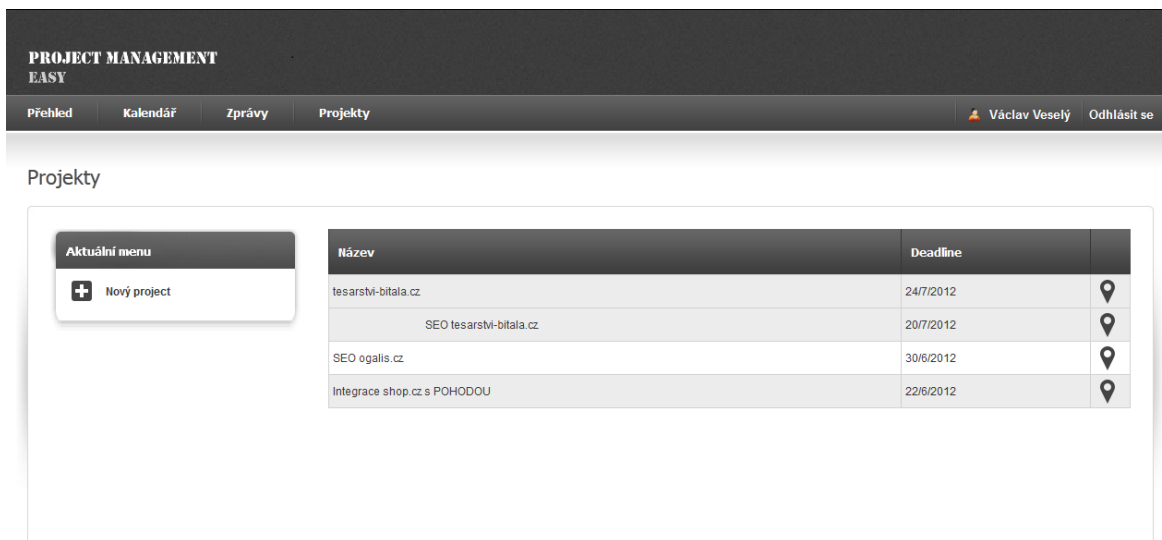
Entita *task_comment* je určena pro uchování komentářů k úkolu. Jedná se o další komunikační prostředek mezi uživateli systému, který se vztahuje ke konkrétnímu úkolu.

4.6 Tvorba grafického designu systému

Po vývoji systému z pohledu návrhových tříd a datového modelu následuje příprava grafického vzhledu aplikace. Tato fáze již probíhala ve výše popsaném nástroji NetBeans, který disponuje kvalitní podporou pro práci v jazyce XHTML a CSS. Grafický výstup zpracovaného návrhu byl následně kontrolován v několika internetových prohlížečích. Konkrétně v prohlížečích Mozilla Firefox 12, Google Chrome 18, Internet Explorer 9.

Vypracování vlastního grafického designu aplikace není cílem této diplomové práce. Na internetu v dnešní době existuje již velké množství připravených a kvalitně zpracovaných návrhů. Ty lze využít i pro vlastní aplikace. Například pod klíčovým slovem *free template* je k nalezení mnoho odkazů na servery nabízející celé sady předpřipravených šablon. Po prozkoumání několika serverů nabízejících tyto služby byl vzhledem k povaze aplikace vybrán návrh ze serveru www.netdreams.co.uk[21]. Přičemž hlavním kritériem výběru byla grafická čistota, respektive grafická jednoduchost. U aplikace pro podporu řízení projektů je především důležitá přehlednost a funkčnost. Vybraný návrh je zajímavý tím, že i přes zmíněné kritérium působí velmi propracovaně a přitom neruší uživatele nadměrným množstvím grafických prvků.

I přes velmi zdařilou šablonu však bylo nutné provést její korekci pro konkrétní vytvářenou aplikaci. Jednalo se především o úpravy pomocí CSS. Zřídka došlo k čistě grafickým úpravám s výjimkou ikon.



Obrázek 16: Design aplikace

Úpravy šablony byly provedeny především v těchto bodech:

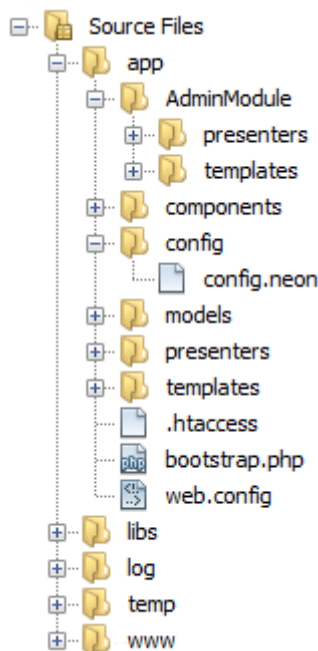
- úprava menu,
- úprava formulářů,
- připojení vertikálního menu,
- změna ikon,
- vytvoření kalendáře,
- vytvoření Ganttova diagramu.

4.7 Programování systému

S připraveným designem začíná hlavní část tvorby aplikace – implementace. Zde je v úvodu popsána struktura systému s bližším popisem MVP modelu. Dále následuje popis jednotlivých částí aplikace, který byl nastíněn v návrhu.

4.7.1 Systémová struktura

V návrhu byla aplikace rozdělena na dvě části dle svého určení. Tomuto členění je také uzpůsobena její adresářová struktura. Stěžejním bodem z pohledu bezpečnosti je dát uživateli přístup pouze do nezbytných adresářů. Struktura systému je tedy rozdělena na adresáře, do kterých lze uživatelem přímo vstoupit a na adresáře, které využívá aplikace pro svůj chod.



Obrázek 17: Systémová struktura

Vybraná technologie pro vývoj aplikace – Nette Framework – je z tohoto pohledu velmi dobře připravena. Pevně daná adresářová struktura je modifikovatelná. To je umožněno pomocí tzv. *loaderu*, který automaticky načítá třídy aplikace. S těmi pak pracujeme nezávisle na jejich umístění. Ve vytvořené struktuře aplikace na obrázku č. 18 je uživateli přístupný pouze adresář *www*. Ten obsahuje CSS styly, JavaScripty, nahrané soubory a soubor *index.php*. V tomto souboru je pak definována cesta k adresáři aplikace *app*, knihovnám *libs* a dočasným souborům *temp*. Právě ve složce *libs* je umístěna i knihovna Nette Frameworku. Ve finále je v *index.php* obsaženo volání souboru *bootstrap.php*.

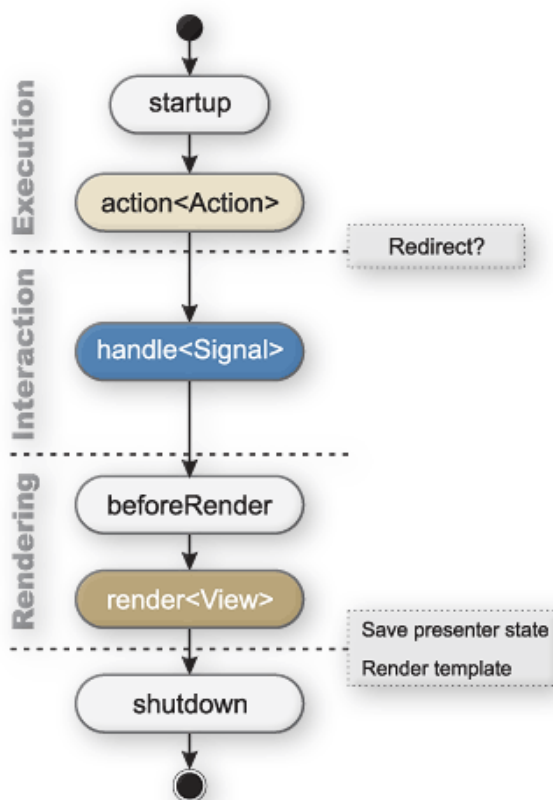
bootstrap.php nastavuje chování aplikace, jako například debugování, routování nebo načtení výše uvedeného *loaderu*. Konfigurační soubor aplikace se nazývá

config.neon, ten *bootstrap.php* také načítá a nastavuje aplikaci dle definovaných parametrů. Obsahuje například parametry pro připojení k databázi nebo pro přihlášení uživatele.

Místem, kde se nachází konkrétní vytvořená aplikace je složka *app*. Ta dle návrhového vzoru MVP obsahuje složky pro presentery, pohledy a modely. Složka *AdminModule* má vlastní presentery a pohledy, které určují specifické chování této části aplikace. Složka *components* slouží pro předem připravené komponenty, které následně využíváme v celé aplikaci.

Presenter

Presenter je prvek realizující změny v modelu nebo presenteru na základě reakcí od uživatele. Je členěn do několika částí dle volání metod, které mohou být typu *action<Action>*, *handle<Signal>* a *render<View>*. Metody určené pro reakci na akci uživatele jsou *action<Action>* a *handle<Signal>*. Jednotlivé části jsou přehledně vidět na obrázku č. 19.



Obrázek 18: Posloupnost kroků presenteru

Zdroj: <http://doc.nette.org/cs/quickstart/presenter>[cit. 2012-05-03]

Pro zpracování signálu je určená metoda *handle*. Tím se rozumí akce, při které se pohled nemění, ale upravuje se pouze ten stávající. Změna se tedy projeví na aktuální stránce. Slouží také ke zpracování AJAXových akcí. Například se může jednat o smazání položky z tabulky úkolů.

```
public function handleDeleteTask($id) {  
    // odebrání úkolu  
}
```

Metoda *action* naopak může obsahovat operace, po kterých následuje přesměrování na jiný pohled. Metoda *render* obstarává vykreslení pohledu. Do šablony je tak předáno několik užitečných informací, jako například *\$basePath*, která určuje absolutní URL cestu ke kořenovému adresáři nebo *\$user*, což je objekt reprezentující uživatele a mnohé další.

Pohled

Pohled je v Nette Frameworku tvořen šablonami. Ty jsou standardně spjaty v konkrétním presenterem, což je umožněno díky dodržení jednoduché konvence pojmenování. Například pro presenter *ProjectPresenter* je výchozím pohledem *default.latte* ve složce *Project*, která se nalézá v adresáři šablon *template*. Další šablony v této složce pak mají stejné pojmenování jako akce presenteru.

Nette Framework využívá v pohledech šablonovací systém Latte, který usnadňuje jejich zápis. Ten obsahuje dva typy speciálních značek:

- makra ve složených závorkách, například {foreach ...};
- n:makra, například n:href="..."

Využití těchto značek umožňuje pracovat s informacemi získanými z presenteru a dynamicky tak vypisovat jejich hodnoty. Příklad jejich použití je možné vidět na následující ukázce zdrojového kódu vypisujícího 7 posledních komentářů.

```
{foreach $lastComments->limit(7) as $comment}  
    <tr n:class="$iterator->isOdd() ? alternate-row : even">  
        <td>{$comment[date]}date:'j.n. H:m'</td>  
        <td>{$comment->task[name]}</td>  
        <td>{$comment->user[name]} {$comment->user[surname]}</td>  
        <td><a n:href="Tasks:comment, $comment[task_id]" title="Detail" class="icon-  
            1 info-tooltip"></a></td>  
    </tr>  
{/foreach}
```

Model

Model využívá v našem případě knihovnu Nette\Database, která je součástí Nette Frameworku. Ta podporuje efektivní práci s databází. Díky ní lze snadno skládat dotazy a získávat tak data. Příklad použití je vidět na ukázce získání členů projektového týmu.

```
public function getProjectTeam() {  
    return $this->database->table('project_member')->where('project_id', $this->id)  
        ->order('project_role.priority, user.surname');  
}
```

4.7.1 Autentizace a autorizace

Celá aplikace je přístupná pouze pro přihlášené uživatele, jejichž přihlášení je realizováno po zadání uživatelského jména a hesla do přihlašovacího formuláře. Uživatelé systému jsou určeni databázovou tabulkou *user*. Aby bylo heslo v databázi bezpečné, je před uložením zahashována zvolená přísada a samotné heslo pomocí metody *sha512*. Na tuto kombinaci je následně použita metoda *sha1()*. Tento finální výstup je pak uložen do databáze.

```
public static function calculateHash($password) {  
    return sha1(hash("sha512", self::$salt . $password));  
}
```

Přístupová oprávnění jsou definována v souboru *libs/Authorizator.php*. Třída tohoto souboru zajišťuje implementaci *ACL*, což je zkratkou slov *Access Control List*. Tato metoda pracuje s definicí 3 základních parametrů – role, zdroj, oprávnění. Množiny těchto parametrů jsou získány z databáze. Jednotlivá oprávnění jsou pak nadefinovanou kombinací uvedených parametrů určena.

```
$this->allow('user_administrator', 'administration', 'show');
```

Říkají tedy jaká práva má pro konkrétní zdroj daná role. Kombinace parametrů je opět získána z databáze – tabulky *user_permission* a *project_permission*. V aplikaci se pak na oprávnění dotazujeme.

```
{if $user->isAllowed('administration', 'show')}
```

Ve vytvářené aplikaci jsou oprávnění rozdělena do 2 typů. Konkrétně na oprávnění aplikační a projektová. Aplikační oprávnění slouží pro určení přístupových práv k funkčním celkům aplikace, jako je například kalendář nebo zprávy. Projektová oprávnění

určují dle role člena projektu jeho oprávnění pro operaci s projektem. Tím se rozumí například právo na vytvoření milníku nebo přidání komentáře. Uživatel může mít v každém projektu jinou roli.

4.7.2 Uživatelská část

Celá aplikace je rozdělena na 2 části – uživatelskou a administrační. Uživatelskou se rozumí ta část aplikace, která slouží pro vlastní projektové řízení. Adresáře této programové části rozdělené dle modelu MPV se nacházejí přímo v kořenovém adresáři aplikace *app*.

Uživatelská část umožňuje autorizovaným uživatelům práci s úkoly, projekty a komunikaci s dalšími uživateli aplikace. Před vstupem do uživatelské části je uživatel nucen přihlásit se přes přihlašovací formulář. Po úspěšném přihlášení je přesměrován na přehled, což je úvodní strana aplikace se seznamem přiřazených úkolů a seznamem posledních komentářů v jemu přiřazených úkolech nebo úkolech, kde komentoval. Z těchto přehledových seznamů je možné pomocí grafických ikon přejít do detailů jednotlivých položek.

PROJECT MANAGEMENT
EASY

Přehled

Kalendář

Zprávy

Projekty

Václav Veselý

Odhlásit se

Přehled

Moje úkoly

Priorita	Projekt	Úkol	Deadline	Dokončeno	
Normální	tesarstvi-bitala.cz	Modul novinky	20/6/2012	0%	
Normální	tesarstvi-bitala.cz	Modul fotogalerie	22/6/2012	0%	
Normální	SEO ogalis.cz	Optimalizace zdrojového kódu	30/5/2012	0%	
Normální	Integrace shop.cz s POHODOU	Realizace integrace	12/6/2012	0%	
Normální	SEO tesarstvi-bitala.cz	Optimalizace zdrojového kódu	22/6/2012	0%	
Nejvyšší	tesarstvi-bitala.cz	Modul texty	15/6/2012	70%	
Vysoká	Integrace shop.cz s POHODOU	Vypracování návrhu integrace	4/6/2012	30%	

Poslední komentáře

Datum	Úkol	Uživatel	
16/5/2012 14:15	Kodovani ostatní stránek	Karel Němec	
16/5/2012 14:14	Kodovani ostatní stránek	Václav Veselý	
16/5/2012 14:12	Modul texty	Václav Veselý	
16/5/2012 14:11	Modul texty	David Černý	

Obrázek 19: Úvodní stránka aplikace - Přehled

Dalším přístupným celkem je kalendář s celoměsíčním přehledem dní. V kalendáři jsou vyznačeny události, které se vztahují ke konkrétnímu uživateli. Tím se rozumí projektové události z projektů, kterých je uživatel členem. Každá tato projektová událost je

vyznačena příslušným grafickým prvkem a názvem. Události slouží také jako odkaz do jejich detailů. Prezентuje také určený termín jejich ukončení a je členěna celkem do tří typů:

- konec úkolu,
- konec milníku,
- konec projektu.

Sekce Zprávy slouží pro podporu komunikace mezi uživateli systému, a to pomocí krátkých textových zpráv. Každou zprávu je možné zaslat několika příjemcům a každý z nich na ni může přímo reagovat. Podpořeno je také členění zpráv do kategorií.

PROJECT MANAGEMENT EASY

Přehled Kalendář Zprávy Projekty Václav Veselý Odhlásit se

Detail zprávy

Aktuální menu

- + Reagovat
- + Nová zpráva
- ✉ Doručené
- ➦ Odeslané
- ☒ Smazané

Od: David Čemý

Pro: Jan Svoboda, Kateřina Dvořáková, Daniela Procházková, Václav Veselý, Lucie Pospíšilová

Čas: 16/5/2012 16:43

Předmět: Testovací zpráva

Text: Testovací zpráva

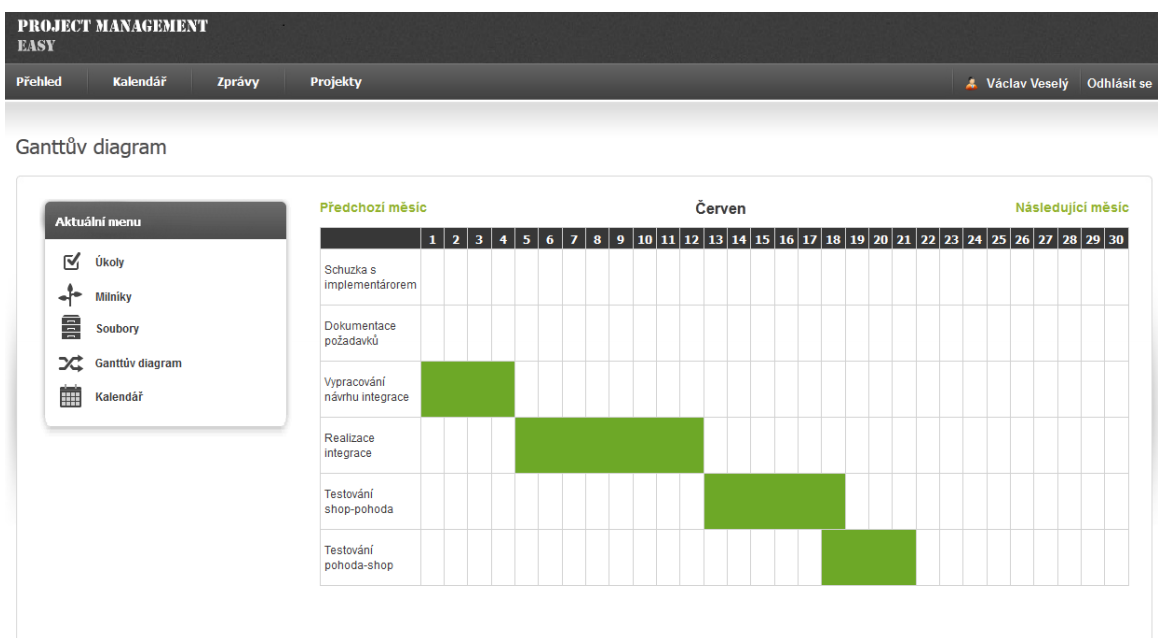
Obrázek 20: Detail doručené textové zprávy

Hlavní uživatelskou částí aplikace je sekce Projekty, která obsahuje seznam projektů příslušného uživatele společně s nástroji pro řízení konkrétního projektu. Mezi tyto nástroje se řadí například možnost nahrání souboru nebo úprava jeho parametrů.

Nejdůležitější částí každého projektu jsou úkoly. Ty prezentují seznamy všech úkolů daného projektu. Každý úkol je možné detailně prohlédnout, s příslušnými právy také upravit, přiložit k němu soubor nebo přidat komentář do diskuse u konkrétního úkolu. Právě na tyto komentáře je uživatel ve svém přehledu upozorňován.

Rozdělení úkolů v rámci projektu je podpořeno sekci Milníky, která umožňuje jejich rychlé vytvoření. Ke každému milníku jsou v jejich přehledu zobrazeny všechny úkoly náležící do příslušného milníku.

U každého projektu je také k dispozici Ganttův diagram ukazující časovou náročnost jednotlivých úkolů. Každý úkol má graficky zvýrazněný příslušný časový úsek, kdy má být vykonáván, a to s ohledem na aktuálně zobrazený měsíc a dny v něm. Detaily jednotlivých úkolů jsou dostupné odkazem z jejich názvů.



Obrázek 21: Ganttův diagram

Události každého projektu jsou také prezentovány v měsíčním kalendáři, který je analogický s kalendářem uživatele. V tomto kalendáři jsou obsaženy události pouze jednoho konkrétního projektu

4.7.1 Administrační část

Přístup do administrační části je realizován přidáním klíčového řetězce */admin* za doménu aplikace. Uživatel je tak přesměrován do prostředí určeného pouze pro administraci. To platí však pouze v případě, že disponuje příslušnými právy. Odstraněny jsou prvky uživatelského prostředí, které jsou určené pro projektové řízení. Aplikační soubory této části nalezneme v adresáři *AdminModule*. Zde se nachází presentery a pohledy. Model je pro celou aplikaci společný.

Po vstupu do administračního rozhraní je uživateli zobrazena úvodní obrazovka, informující o vstupu do této části, společně s vertikálním menu, které slouží jako

rozcestník do jednotlivých částí administrace. Každá část administruje určitou specifikou část systému.

Část Projekty prezentuje seznam všech projektů v systému s možností jejich úprav. Administrátor má stejně jako v ostatních částech právo jednotlivé položky definitivně z databáze odstranit. Důležitou sekcí této části je *Role v projektech*. Zde se definují jednotlivá oprávnění projektových rolí. Výčet rolí není konečný, je možné přidat i další roli a té následně určit specifická práva.

V milnících se nalézá seznam milníků všech projektů. Každý milník lze editovat nebo případně smazat. V případě, že mazaný milník obsahuje úkoly, je uživatel upozorněn patřičnou textovou informací.

Část Úkoly disponuje obdobnými funkcemi jako sekce Projekty, ale zde slouží k práci s úkoly. Rozšiřující funkcí je zde definice několika číselníků využívaných v úkolech – priority, stavy a typy. Ty umožňují měnit jejich názvy a pořadí v systému.

Sekce Soubory a Zprávy obsahují seznam všech existujících položek těchto entit. Každou položku je opět možné odstranit nebo případně prohlédnout její detail.

Nejdůležitější část Uživatelé, jak název napovídá, definuje uživatele systému. Umožňuje přidat nového uživatele s definicí jeho základní parametrů, především jeho uživatelský login, heslo a roli v aplikaci. Role v aplikaci a její oprávnění se určuje obdobně jako v sekci projekty.

4.8 Testování a nasazení systému

Nasazení celé aplikace na produkčním serveru je podmíněno jejím otestováním. V průběhu vývoje byla aplikace testována pomocí XAMPP balíčku se serverem Apache ve verzi 2.2.21 a PHP ve verzi 5.3.8. Databázovou podporu v tomto balíčku poskytovala databáze MySQL ve verzi 5.5.15. Výstup aplikace byl kontrolován v několika nejpoužívanějších webových prohlížečích. Konkrétně v prohlížečích Mozilla Firefox 12, Google Chrome 18 a Internet Explorer 9.

Pro funkčnost aplikace na produkčním serveru je potřeba ověřit několik základních podmínek, které jsou k chodu aplikace nutné. První podmínkou je podpora Nette Frameworku s PHP na daném produkčním serveru. Pro ověření, zda daný produkční server

splňuje všechny parametry funkčnosti z pohledu této technologie, existuje skript s názvem *checker.php*. Ten je umístěn v adresáři *www/checker*. Zadáním příslušné URL adresy k tomuto souboru ve webovém prohlížeči skript ověří nastavení serveru a prezentuje výstup této kontroly s komentáři k případným varováním a chybám.

Další podmínkou je ověření přístupových práv do jednotlivých adresářů aplikace. *Document root* aplikačního serveru má směřovat do adresáře *www*. Zbylé adresáře by z důvodu bezpečnosti neměly být prostřednictvím http protokolu přístupné.

Nastavení databázového připojení na produkční databázi se realizuje v souboru *app/config/config.neon*. Při úpravě tohoto souboru je nutné brát v potaz jeho strukturu. Ta využívá tabulátory, a je tedy důležité s touto vlastností počítat. Pro vytvoření základní databázové struktury slouží přiložený soubor *install_db_script.sql*. Ten strukturu vytvoří pomocí DDL skriptů a naplní ji základními informacemi. Tím se rozumí základní data, jako je účet administrátora (uživatel: admin, heslo: admin), přednastavená oprávnění rolí a obsah číselníků systému. Pro případnou ukázkou aplikace s testovacími daty je vytvořen další soubor *example_data_script.sql*, jehož data rozšiřují předchozí skript o předpřipravenou ukázkou několika rozpracovaných projektů.

Administrace aplikace je přístupná po přihlášení administrátora z URL adresy *www.[doména-aplikace].cz/admin*.

5 Závěr

Jedním z cílů této diplomové práce bylo získání přehledu o současných komerčních nástrojích pro projektové řízení. Takto získané informace posloužily jako základ pro praktickou část této práce. Ta obsahuje podrobný popis návrhu a tvorby vlastní aplikace pro projektové řízení fungující na principu webové aplikace.

I samotná tvorba aplikace pro podporu řízení projektů probíhala jako projekt, a to v několika fázích, které jsou rozděleny do milníků. Ty byly podrobně zdokumentovány. Za časově nejnáročnější lze považovat analytickou fázi práce. Čas investovaný v této fázi se však zhodnotil při implementaci, která stavěla především na vhodně navrženém databázovém úložišti.

Cílem práce nebylo vytvoření „té nejlepší“ aplikace pro projektové řízení. Její vývoj byl směřován především k vytvoření základní modulární aplikace s důrazem na rozšiřitelnost, použitelnost a bezpečnost. To vše za podpory provedené analýzy principů projektového řízení, komerčních nástrojů a s využitím moderních technologií, jako je návrhový vzor MVP nebo Nette\Database knihovna pro práci s databází. Na těchto pilířích vznikla aplikace PROJECT MANAGEMENT EASY, která plní svůj účel. Uživatelům systému nabízí bezpečnou a jednoduchou aplikaci umožňující projektové řízení. Provozovateli na druhou stranu jednoduchou administraci a možnost dalšího relativně snadného rozšíření systému.

Z časových důvodů obsahuje vytvořená aplikace především základní služby, které umožňují spravovat jednotlivé projekty a úkoly. Implementace všech prvků projektového řízení je nad rámec této diplomové práce. Další služby však plánuji do budoucna přidat. Aplikace s rostoucím počtem nabízených služeb bude více schopná konkurovat existujícím komerčním nástrojům. Plánováno je například integrování aplikace pro podporu verzí projektů, sledování nákladovosti projektu, sledování vytíženosti zdrojů, vytvoření Wiki stránek nebo exportní rozhraní do MS Project a mnohé další. Plánována jsou však také vylepšení stávajících služeb, jako je například rozesílání zpráv projektové skupině, grafické zobrazení návaznosti úkolů v Ganttově diagramu nebo filtrování seznamových položek.

Na aplikaci vytvořenou v rámci této diplomové práce nakonec platí, stejně jako pro všechny ostatní aplikace, obecně známé pravidlo. Vždy se najde něco, co lze zlepšit a zdokonalit s cílem zvýšení kompetence aplikace a komfortu uživatelů. Podstatné však je, že díky použitým standardům a známým postupům se případná rozšíření a úpravy budou relativně snadno realizovat.

Zdroje

- [1] KOVAČIK, J. Hlad po projektovém řízení roste. *IT SYSTEMS* [on-line]. 2010(č. 10) [cit. 2012-02-08]. Dostupné z: <http://www.systemonline.cz/řízení-projektu/hlad-po-projektovem-řízení-roste.htm>
- [2] ISO 10006. *Systémy managementu jakosti - Směrnice pro management jakosti projektů*. Český normalizační institut, 2004.
- [3] KERZNER, H. *Project Management Metrics, KPIs, and Dashboards: A Guide to Measuring and Monitoring Project Performance*. Ventyx: Cover Art, 2011. ISBN 1118026527.
- [4] DOLEŽAL, J. LACKO, B. MÁCHAL, P. *Projektový management podle IPMA*. Praha: Grada, 2009. ISBN 8024728486.
- [5] MILTON, D. Rosenau. *Řízení projektů*. 2. vyd. Brno: Computer press, 2003. ISBN 80-7226-218-1.
- [6] Metoda CPM (Critical Path Method). *ManagementMania.com* [on-line]. 2011 [cit. 2012-05-05]. Dostupné z: <http://managementmania.com/metoda-cpm>
- [7] Wiki. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001-2012 [cit. 2012-02-18]. Dostupné z: <http://cs.wikipedia.org/wiki/Wiki>
- [8] Microsoft Project. In: *Wikipedia: the free encyclopedia* [on-line]. San Francisco (CA): Wikimedia Foundation, 2001-2012, 25. 7. 2011 [cit. 2012-02-20]. Dostupné z: http://cs.wikipedia.org/wiki/Microsoft_Project
- [9] KATHY, S. a Krejčí, H. *Řízení projektů v IT*. Brno: Computer Press, 2011. ISBN 978-80-251-2882-4.
- [10] KUBÁLEK, T. a KUBÁLKOVÁ, T. *Řízení projektů v Microsoft Project 2010 : učebnice*. Brno: Computer Press, 2010. ISBN 978-80-251-3266-1.
- [11] Základní popis aplikace Enterprise Architect. *Enterprise Architect CZ* [on-line]. 2010 [cit. 2012-02-28]. Dostupné z: <http://www.enterprise-architect.cz/>
- [12] *NetBeans* [on-line]. 2012 [cit. 2012-02-28]. Dostupné z: <http://netbeans.org/>
- [13] MySQL. In: *Wikipedia: the free encyclopedia* [on-line]. San Francisco (CA): Wikimedia Foundation, 2001-2012, 10. 2. 2012 [cit. 2012-02-29]. Dostupné z: <http://cs.wikipedia.org/wiki/MySQL>

- [14] Databáze a jazyk SQL. *Interval.cz* [on-line]. 04. 08. 2000 [cit. 2012-02-29]. Dostupné z: <http://interval.cz/clanky/databaze-a-jazyk-sql/>
- [15] Jazyk SQL. Ing. Vojtěch HORDEJČUK [on-line]. [cit. 2012-02-29]. Dostupné z: <http://voho.cz/wiki/informatika/jazyk/sql/>
- [16] PHP. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001-2012 [cit. 2012-03-01]. Dostupné z: <http://cs.wikipedia.org/wiki/PHP>
- [17] Nette Framework. In: *Wikipedia: the free encyclopedia* [on-line]. San Francisco (CA): Wikimedia Foundation, 2001-2012 [cit. 2012-03-01]. Dostupné z: http://cs.wikipedia.org/wiki/Nette_Framework
- [18] *Dokumentace Nette* [on-line]. 2012 [cit. 2012-03-01]. Dostupné z: <http://doc.nette.org/>
- [19] NEUSTADT, I. a ARLOW, J. *UML 2 a unifikovaný proces vývoje aplikací. 2., aktualiz. a dopl. vyd.* Brno: Computer Press, 2007. ISBN 978-80-251-1503-9.
- [20] Modelování databází. *Root.cz* [on-line]. 3. 4. 2002 [cit. 2012-04-30]. Dostupné z: <http://www.root.cz/clanky/modelovani-databazi/>
- [21] Free admin skin available. *Internet Dreams* [on-line]. 2012 [cit. 2012-05-03]. Dostupné z: <http://www.netdreams.co.uk/index.php/blog/2010/02/18/free-admin-skin-available-for-download/>

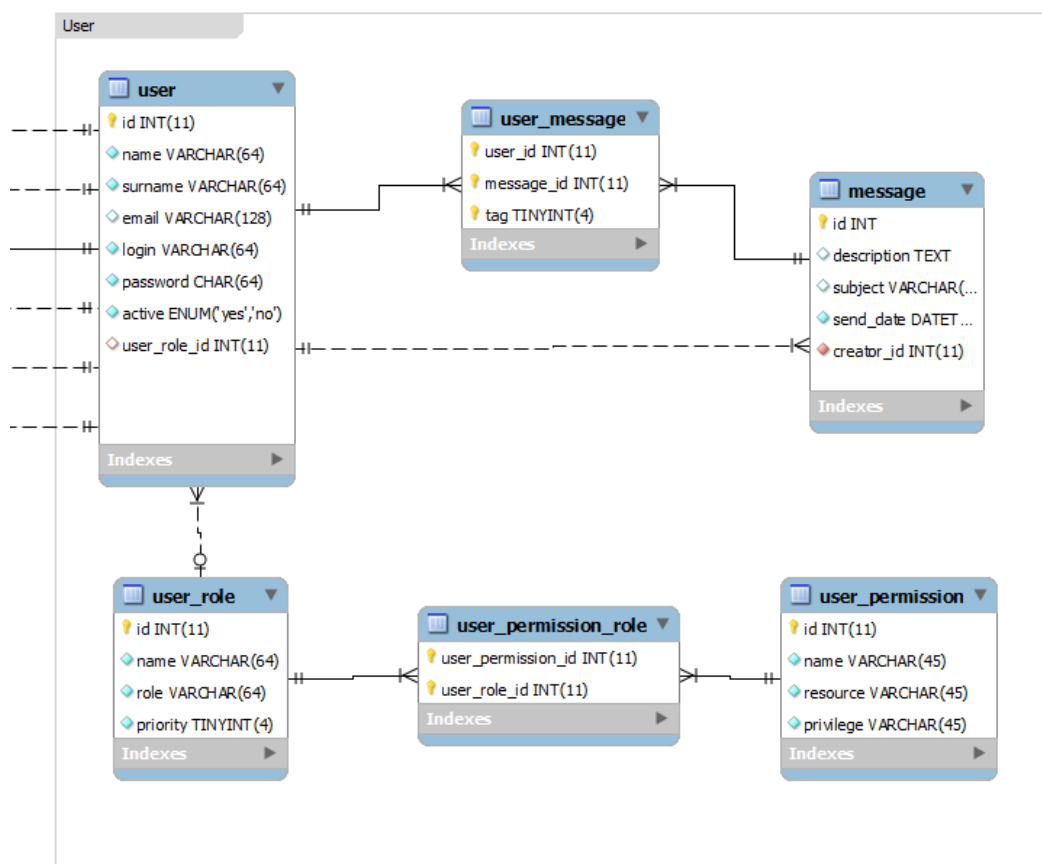
Seznam příloh

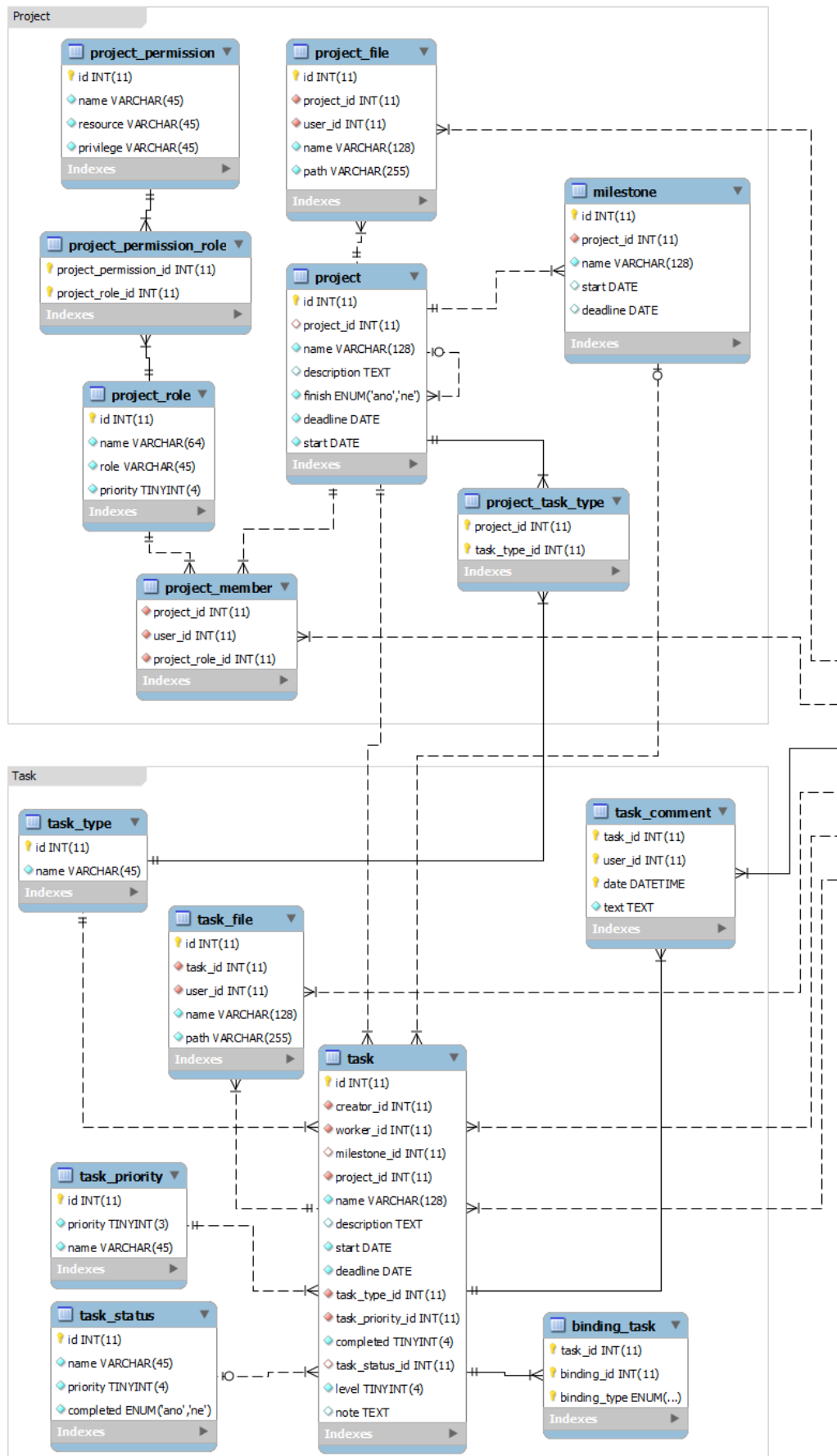
Příloha A – Adresářová struktura přiloženého CD s popisem obsahu	68
Příloha B – Kompletní datový model databáze	69
Příloha B – Uživatelský manuál	71

Příloha A – Adresářová struktura přiloženého CD s popisem obsahu

- *PROJECT MANAGEMENT EASY* - Adresář obsahující data vytvořená při návrhu a implementaci aplikace PROJECT MANAGEMENT EASY.
 - *analiza.eap* - Soubor obsahující zpracované požadavky, Use-Case diagramy a návrhové třídy.
 - *aplikace* - Adresář obsahující zdrojové kódy aplikace.
 - *databaze* - Adresář obsahující diagramy a skripty databáze.
 - *er_diagram_pdf.pdf* - Schéma databázového modelu ve formátu pdf.
 - *er_diagram_workbench.mwb* - Schéma databázového modelu ve formátu aplikace MySQL Workbench.
 - *example_data_script.sql* - SQL skript pro vygenerování ukázkových dat na databázi MySQL.
 - *install_db_script.sql* - SQL skript pro vygenerování základní databázové struktury na databázi MySQL.
- *BitalaR_NavrhImplementace_JH_2012.pdf* - Text diplomové práce ve formátu pdf.
- *readme.txt* – Soubor popisující obsah CD, postup instalace aplikace a nahrání ukázkových dat.

Příloha B – Kompletní datový model databáze





Uživatelský manuál

PROJECT MANAGEMENT EASY

1 Obecné informace

Celá aplikace je přístupná pomocí webového prohlížeče, a to po zadání příslušné adresy aplikace. Systém je optimalizovaný pro internetové prohlížeče Mozilla Firefox 12, Google Chrome 18 a Internet Explorer 9. Aplikace po přihlášení obsahuje hlavní horizontální menu, které může dle uživatelských oprávnění obsahovat následující položky:

- Přehled,
- Kalendář,
- Zprávy,
- Projekty.

Tyto položky jsou v aplikaci v případě potřeby doplněny vertikálním menu na levé straně, které rozšiřuje funkce aktuální sekce. Všechny tyto části jsou níže podrobně popsány. Obsahem každé z výše uvedených položek jsou seznamy informací dle jejich zaměření. Ty jsou vhodně doplněny akčními ikonami pro operace s jednotlivými položkami.

V případě vytváření nových položek pomocí formulářů jsou zadaná data kontrolována. Tím se rozumí kontrola povinně vyplněných položek a správnosti vyplněných informací. V případě rozporu je uživatel o chybě textově informován.

Práce s aplikací je umožněna pouze přihlášeným uživatelům. První stránkou po zadání adresy aplikace je tedy přihlašovací okno. V tomto kroku je nutné, aby uživatel systému vyplnil příslušné přihlašovací údaje – „Uživatel“ a „Heslo“. Tyto údaje jsou přiděleny administrátorem systému, který uživatele zakládá. V případě jejich správného zadání a odeslání tlačítkem „Přihlásit“ je uživatel přesměrován na úvodní stranu aplikace, kterou je „Přehled“. V opačném případě je uživatel v přihlašovacím okně informován o nesprávně vyplněných údajích.

Popis níže uvedeného obsahu aplikace je globální. Její funkce mohou být limitovány uživatelskými oprávněními.

2 Přehled

Jak již bylo výše naznačeno, po přihlášení se zobrazí sekce Přehled. Ta obsahuje dva oddělené seznamy přehledně informující uživatele o aktivitách v systému, které se k němu přímo vztahují. Konkrétně se jedná o seznam úkolů, které jsou uživateli přiřazeny a seznam posledních komentářů přidáných k těmto úkolům.

Položky v seznamu přiřazených projektů obsahují dvě akční ikony. První ikona slouží k prohlídce detailních informací o úkolu. Ty obsahují v tříslopcovém formátu parametry úkolu, popis s poznámkou a vazby úkolu k dalším úkolům projektu. Ty budou níže popsány v části Úkoly, která spadá pod sekci Projekty. Uživatel tak získá zadání úkolu a může jej realizovat. Druhá akční ikona vede k úpravě úkolu. Zde uživatel průběžně upravuje stavy úkolů, které jsou po uložení promítány do systému.

Seznam posledních komentářů zobrazuje 5 posledních komentářů, které byly vloženy k úkolům přiřazeným uživateli. Říkají uživateli o případných nových informacích, které se vztahují k jeho úkolům. Akční ikona položky vede přímo do sekce komentářů u daného úkolu.

3 Kalendář

Tato sekce slouží k orientaci uživatele v časovém harmonogramu projektových událostí. Po vstupu je zobrazen kalendář aktuálního měsíce s výpisem všech jeho dní. Přesun na jiný kalendářní měsíc je možné realizovat pomocí odkazů Předchozí měsíc, respektive Následující měsíc.

V kalendáři jsou příslušným grafickým prvkem znázorněny 3 typy projektových událostí:

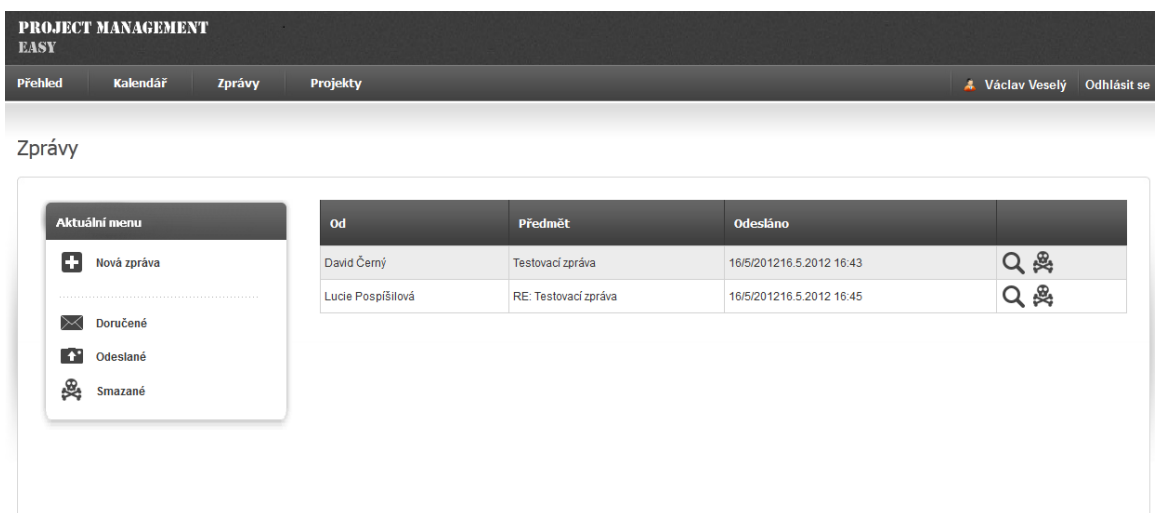
- ukončení projektu,
- ukončení milníku,
- ukončení úkolu.

Do kalendáře jsou promítnuty veškeré projektové události, kterých je uživatel členem. Každá z nich v kalendáři obsahuje také svůj název. Ten slouží k přechodu do detailu konkrétní projektové události.

4 Zprávy

Zprávy slouží pro komunikaci mezi uživateli systému pomocí krátkých textových zpráv. Ty je možné zaslat jednomu konkrétnímu uživateli nebo vybrané množině uživatelů.

Po vstupu do této sekce se zobrazí stránka se seznamem doručených zpráv. Každá položka seznamu značí jednu doručenou zprávu a je doplněna dvěma akčními ikonami. Na levé straně se nalézá doplňující menu, které je rozcestníkem do dalších sekcí, jako jsou Odeslané zprávy, ty obsahují seznam zpráv, které uživatel odeslal ostatním uživatelům systému a Smazané zprávy obsahující položky, které uživatel z doručených a odeslaných zpráv odstranil v pořadí druhou akční ikonou. V menu se také nalézá volba pro vytvoření nové zprávy.



Obrázek 22: Sekce zprávy

Po kliknutí na volbu Vytvoření zprávy je zobrazen jednoduchý formulář s položkou pro předmět a vlastní obsah zprávy. Uživatelé, kterým je zpráva určena se definují na pravé straně okna, a to je jednoduchým zaškrtnutím konkrétních uživatelů. K odeslání zprávy dojde po vyplnění informací kliknutím na tlačítko Odeslat.

První akční ikona u každé položky v této sekci slouží k nahlédnutí do detailu zprávy. Kliknutím jsou uživateli prezentovány informace o odesílateli, adresátech, čase odeslání, předmětu a vlastním obsahu zprávy. V této části je vertikální menu doplněno položkou pro reakci na zprávu. Jejím využitím je uživatel přesměrován na formulář pro tvorbu nové zprávy, který je vyplněn informacemi z dané zprávy včetně adresátů. Textové

informace jsou doplněny o znaky značící, že se jedná o reakci na zprávu. Uživatel pak doplněním dalších informací a odesláním reaguje na konkrétní zprávu.

5 Projekty

Tuto sekci lze považovat za stěžejní v celé aplikaci. Vstupní zobrazení obsahuje seznam projektů, v nichž je uživatel členem. Na levé straně se opět nalézají vertikální menu s položkami pro aktuální sekci. To je v detailech jednotlivých projektů rozšířeno o akční ikony projektu.

Vytvoření nového projektu se realizuje přes položku Nový projekt. Dojde k přesměrování na formulář pro jeho vytvoření, kde se definují základní parametry včetně členů projektového týmu. Ti jsou určeni zaškrtnutím na pravé straně v části Pracovníci. Po zaškrtnutí je zobrazena u konkrétního uživatele volba jeho role ve vytvářeném projektu. Konečné vytvoření projektu je provedeno kliknutím na tlačítko Uložit.

Po vstupu do projektu přes akční ikonu v její položce je zobrazena v levém menu sada odkazů, které jsou rozcestníkem do obsažených nástrojů pro řízení projektů. V pořadí se jedná o Úkoly, Milníky, Soubory, Ganttův diagram a Kalendář. Tyto nástroje jsou níže podrobně popsány. Poslední ikona (Úpravy) slouží k úpravě parametrů projektu.

5.1 Úkoly

Zde se nachází přehledný seznam základních pilířů celého projektu. Novou položku do tohoto seznamu lze přidat přes položku Přidat úkol, a to z aktuálního menu. To vede na formulář k vytvoření nového úkolu se všemi jeho parametry, jako je například Realizátor, Deadline nebo vazba na další úkoly. Ty lze definovat v pravé části v okně Vazby, a to jednoduchým zaškrtnutím a výběrem příslušné vazby k dalším úkolům projektu.

Vazby jsou rozděleny do 4 skupin:

- úkol může začít, až vybraný skončí;
- úkol může skončit, až vybraný skončí;
- úkol může začít, až vybraný začne;
- úkol může skončit, až vybraný začne.

Definitivní vytvoření úkolu je realizováno prostřednictvím tlačítka Uložit.

PROJECT MANAGEMENT EASY

Přehled Kalendář Zprávy Projekty David Černý Odhlásit se

Nový projekt

Název:

Začátek:

Deadline:

Dokončeno:

Podprojektem:

Popis:

Typy úkolů:
☒ Návrh
☒ Analýza
☒ Programování

Pracovníci

- ☐ Admin Admin (0)
- ☐ David Černý (0)
- ☐ Kateřina Dvořáková (0)
- ☐ Karel Němec (0)
- ☐ Jiří Novák (0)
- ☐ Stanislav Novotný (0)
- ☐ Lucie Pospíšilová (2)
- ☐ Daniela Procházková (4)
- ☐ Jan Svoboda (3)
- ☐ Václav Veselý (7)

Uložit

Obrázek 23: Formulář pro vytvoření projektu

Jednotlivé úkoly je možné podrobně prohlížet pomocí první akční ikony u každé položky. Druhá pak slouží k úpravě parametrů konkrétního úkolu.

Třetí ikona vede ke komentářům sloužícím pro komunikaci mezi uživateli v rámci jednotlivých úkolů. V horní části je umístěno okno pro vytvoření nového komentáře. Ten je přidán kliknutím na tlačítko Odeslat. Pod tímto oknem jsou umístěny komentáře, které jsou seřazeny od nejnovějších po nejstarší.

Poslední akční ikona každé položky podporuje nahrávání případných dokumentů k úkolu. V první řadě je zde zobrazen seznam nahraných souborů s možností jejich stažení. Pod tímto seznamem se nachází jednoduchý formulář pro nahrání nových souborů, které jsou vybrány přímo z počítače.

5.2 Milníky

Díky milníkům je umožněno dělení úkolů projektu do celků ohraničujících určitý cíl. Jejich vytváření se provádí pomocí další položky z aktuálního menu – Přidat milník, kde se kromě jeho názvu určuje jeho začátek a deadline. Seznam pak přehledně zobrazuje

jejich množinu v rámci projektu. Každý milník prezentuje také seznam úkolů, které jsou v každém z nich obsaženy.

5.3 Dokumenty

Zde se jedná o analogický nástroj jako dokumenty u úkolu. Umožňuje doplnit projekt o dokumentaci prostřednictvím souborů nahraných přímo z počítače.

5.4 Ganttův diagram

Tento speciální diagram poskytuje přehled o časové náročnosti a návaznosti všech úkolů projektu. Po vstupu je zobrazen aktuální měsíc s výčtem všech jeho dní. Přejít na další měsíce je umožněn pomocí odkazu Předchozí měsíc, respektive Následující měsíc nad tímto diagramem.

V levém sloupci diagramu jsou seřazeny jednotlivé úkoly. Ty slouží také pro přechod do svých detailů. Graficky jsou pak zvýrazněny na stejném řádku časové úseky dle určeného termínu zahájení a ukončení konkrétního úkolu.

5.5 Kalendář

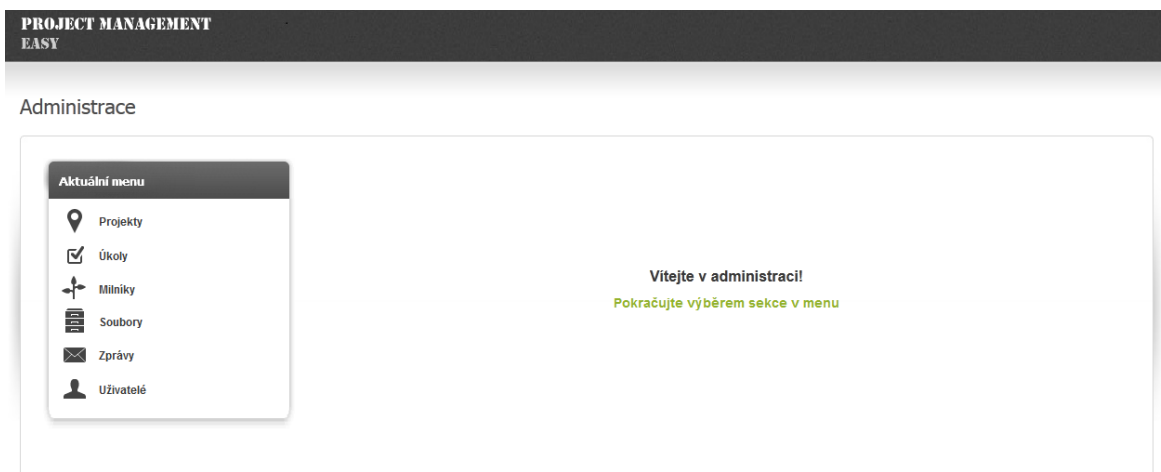
Tento nástroj v prostředí úkolu funguje analogicky jako kalendář uživatele popsaný výše. Rozdíl je v tom, že v tomto kalendáři jsou prezentovány pouze události jednoho konkrétního projektu.

6 Administrace

Administrace je speciální částí aplikace oddělené od nástrojů projektového řízení. Ta správci systému umožňuje mít přehled nad všemi informacemi, které se v systému nachází. Práva vstupu do této části má pouze uživatel s administrátorským oprávněním.

Vlastní přechod do administrace se provádí přidáním klíčového *admin* za doménu aplikace – [www.\[doména-aplikace\].cz/admin](http://www.[doména-aplikace].cz/admin). Po zadání je uživatel přesměrován na stránku s informacemi o vstupu do této sekce a s vertikálním menu, které je rozcestníkem do jednotlivých administrovatelných částí.

Práce s položkami v jednotlivých sekcích je analogická s prací v aplikaci. Rozdíl je pouze v tom, že případná odstranění položek jsou zde chápána jako jejich definitivní odstranění z databáze. V případě, že položku není možné odstranit z důvodu integritních omezení, je uživatel informován chybovou hláškou s příslušnými informacemi.



Obrázek 24: Prostředí administrace

Důležitými částmi celé administrace jsou sekce Role v projektech (v části Projekt) a Role v aplikaci (v části Uživatelé). Cestu k nim nalezneme v obou případech v aktuálním menu příslušných částí. Zde jsou určena oprávnění jednotlivých rolí, a to buď v rámci projektu nebo aplikace. Pod seznamem rolí, který je možné rozšířit o další role využitím příslušných odkazů z aktuálního menu, jsou vypsána jednotlivá oprávnění. Ta lze zaškrtnutím nebo odškrtnutím upravit. Úpravy jsou pak stvrzeny kliknutím na tlačítko Uložit.

Správa číselníků využívaných v úkolech – priorit, stav, typ – se nalézá v části Úkoly. Každý z těchto číselníků lze dle konkrétních užití editovat nebo rozšiřovat.