VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

VÝBĚR A NÁVRH SYSTÉMU PRO TÝMOVOU SPOLUPRÁCI A SPRÁVU ÚKOLŮ

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE FRANTIŠEK VÍDEŇSKÝ
AUTHOR

BRNO 2015

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

VÝBĚR A NÁVRH SYSTÉMU PRO TÝMOVOU SPOLUPRÁCI A SPRÁVU ÚKOLŮ

SELECTION AND PROPOSAL OF THE SYSTEM FOR TEAM COLLABORATION AND ACTIVITIES CONTROL

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE FRANTIŠEK VÍDEŇSKÝ
AUTHOR

VEDOUCÍ PRÁCE Ing. ŠÁRKA KVĚTOŇOVÁ, Ph.D.
SUPERVISOR

BRNO 2015

Abstrakt

Tato bakalářská práce se zabývá problematikou projektového řízení a návrhem webové aplikace pro jeho podporu. Nejprve je vysvětlena teorie projektového řízení, největší důraz je kladen na řízení času. Následuje analýza stávajících řešení a návrh vlastní aplikace. Aplikace byla implementována v jazyce C# s použitím platformy ASP.NET MVC.

Abstract

This bachelor´s thesis deals with the issues of the project management and desing web application for its support. At first the theory of the project management is explained, the greatest emphasis is given to the Time Management. The next is analysis of existing solutions and of desing my own appliacation. The applicaton has been implemented in the programming language C# with using of ASP.NET MVC platform.

Klíčová slova

projekt, projektové řízení, projektový management, týmová spolupráce, CPM, PERT, ASP.NET, MVC

Keywords

project, project management, teamwork, CPM, PERT, ASP.NET, MVC

Citace

Vídeňský František: Výběr a návrh systému pro týmovou spolupráci a správu úkolů, bakalářská práce, Brno, FIT VUT v Brně, 2015

Výběr a návrh systému pro týmovou spolupráci a správu úkolů

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením paní Ing. Šárky Květoňové, Ph.D. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

……
František Vídeňský
18. května 2015

Poděkování

Rád bych poděkoval paní Ing. Šárce Květoňové, Ph.D. za odborné vedení, cenné rady a vstřícnost při konzultacích a vypracování této práce.

© František Vídeňský, 2015
Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1 Úvod ... 3
2 Projektové řízení ... 4
2.1 Projekt a trojí omezení projektu ... 4
2.1.1 Charakteristické rysy projektu ... 4
2.1.2 Trojí omezení projektu ... 4
2.2 Životní cyklus řízení projektu ... 5
2.2.1 Předprojektová fáze ... 6
2.2.2 Projektová fáze ... 6
2.2.3 Poprojektová fáze ... 7
2.3 Procesy projektového řízení ... 7
2.4 Časové plánování ... 8
2.4.1 Seřazení aktivit ... 9
2.4.2 Síťové diagramy ... 9
2.4.3 Ganttův diagram ... 10
2.4.4 Metoda CPM ... 10
2.4.5 Metoda PERT ... 11
2.5 Řízení lidských zdrojů a komunikace ... 12
2.5.1 Nábor pracovníků ... 13
2.5.2 Plánování komunikace ... 13
2.5.3 Životní fáze týmu ... 14
3 Analýza stávajících řešení ... 15
3.1.1 Nástroje pro menší podniky ... 15
3.1.2 Nástroje pro velké podniky ... 16
3.1.3 Zhodnocení analýzy ... 17
4 Návrh Aplikace ... 18
4.1 Definice kritérií ... 18
4.2 Možnosti aplikace ... 18
4.3 Návrh datového modelu ... 21
5 Implementace ... 24
5.1 Použité nástroje ... 24
5.1.1 Entity framework ... 24
5.1.2 ASP.NET ... 24
5.1.3 MVC architektura ... 25
5.1.4 Repozitáře ... 26
5.1.5 Dependency injection ... 26

5.1.6 Knihovna JQuery ... 27
5.2 Implementace vybraných částí ... 27
5.2.1 Ganttův diagram ... 27
5.2.2 Síťový diagram ... 28
5.2.3 Přidávání lidí do projektu a jejich přiřazení k aktivitám ... 28
5.2.4 Práce na aktivitě ... 29
6 Popis jednotlivých stránek ... 30
6.1.1 Společné stránky ... 30
6.1.2 Stránky pro pracovníky ... 32
6.1.3 Stránky pro projektového manažera ... 34
7 Ověření funkčnosti ... 37
8 Možnosti dalšího rozšíření ... 38
9 Závěr ... 39

1 Úvod

V dnešní době, kdy je nutné rychle a dynamicky měnit plány, se podniky všech velikostí potýkají s hledáním co nejúčinnějších nástrojů a prostředků pro efektivní realizaci svých podnikatelských aktivit.

Z tohoto důvodu používají projekty k plánování své činnosti a řízení. Projekty existují už dlouhou dobu a prošly různými změnami při svém vývoji. Dnes známe několik různých metod a postupů, jak je řídit. Projekty mohou být velmi jednoduché a krátké, ale také velmi náročné. Může být do nich zařazeno nespočet různých lidí, či dokonce několik různých firem. Ovšem vždy projekt můžeme popsat jako „organizované úsilí k dosažení cíle“ [1].

Ke správné organizaci projektu a dosažení co nejlepšího výsledku je nutné dodržet správné projektové řízení, o kterém pojednává druhá kapitola. Velmi výhodné je mít správnou softwarovou podporu, která nám poslouží jako průvodce jednotlivými kroky, provede za nás různé výpočty a přehledně nás bude informovat o stavu nebo různých úskalích našeho projektu. Můžeme jeho pomocí vytvářet různé dílčí úkony, přiřadit je jednotlivým lidem a sledovat jejich průběžné plnění. Projekty nám mohou také pomoci s rozdělením firemních zdrojů mezi více projektů, které běží současně.

Cílem mé práce je vytvořit vlastní nástroj pro podporu projektového řízení, který bude však více zaměřen na týmovou spolupráci. Třetí kapitola obsahuje analýzu různých známých softwarů pro podporu projektového řízení, kde jsou rozděleny do kategorií podle velikostí podniků, pro které jsou určeny. Návrh aplikace je podrobně popsán ve čtvrté kapitole. Následná implementace je popsána v kapitole páté.

2 Projektové řízení

Asi každému z nás byl již někdy v minulosti, dán na bedra úkol vést třeba i malý projekt. V této situaci k dokončení nestačí už jen naše odborné dovednosti, ale musíme se začít zabývat časovým plánem, organizací lidských zdrojů, sestavením a dodržením rozpočtu. K úspěšnému naplánování a dokončení je velmi vhodné znát zásady a metody projektového řízení. Proto jsou v této kapitole popsány jednotlivé části řízení, kde největší důraz je kladen na řízení času.

2.1 Projekt a trojí omezení projektu

Už v první kapitole jsme si uvedli základní definici projektu. Existuje jich samozřejmě mnohem více a každá společnosti si může pod tímto pojmem představit něco jiného.

Projekt je dočasný úkol s přesně stanoveným cílem a jeho splnění vyžaduje organizované využití odpovídajících zdrojů. Formulace projektu je klíčovým bodem. Když projekt není správně připraven, nemůže být ani správně řízen [2].

2.1.1 Charakteristické rysy projektu

Existují čtyři charakteristické rysy projektu, které, pokud se vyskytují souběžně, odlišují řízení projektu od jiných manažerských činností. Projekty mají cíl, který má tři rozměry, jsou jednoznačné, zahrnují zdroje a jsou realizovány v rámci organizace [1]. Čtyři základní rysy jsou dle [1] popsány následovně:

- *Cíl.* Každý projekt má svůj trojrozměrný cíl, který nám říká, jakou práci, za jaký čas a za kolik peněz jsme schopní vykonat. Tento trojrozměrný cíl se označuje také jako trojí omezení projektu nebo trojimperativ a je mu věnována příští podkapitola 2.1.2.
- *Jedinečnost.* Projekt je vždy jedinečný, protože je vždy vykonáván jen jednou a jen dočasně na něm pracuje skupina lidí. Mnohé projekty jsou si velmi podobné, ale nikdy nejsou totožné. Projekty začínají a končí v jiném časovém období, nábor zaměstnanců je organizován podle různých pravidel a i lidé a firmy se projekt od projektu liší.
- *Zdroje.* Projekty potřebují vždy nějaké zdroje, ať už se jedná o materiální nebo lidské zdroje. Správný projektový manažer musí správně naplánovat práci pro lidské zdroje tak, aby efektivně využily materiální zdroje.
- *Organizace.* Řízení projektů je z velké části zvládání mezilidských vztahů, které jsou neodmyslitelnou součástí složitých situací uvnitř organizace.

Pro úspěch projektu jako celku je velmi důležitá osoba *projektového manažera* („Projektový manažer je odpovědný za spolupráci se zadavatelem projektu, projektovým týmem a dalšími lidmi zapojenými do projektu a jeho úkolem je naplnit stanovené cíle projektu.“ [3]).

2.1.2 Trojí omezení projektu

V souvislosti s projekty a projektovými cíli se setkáváme v podstatě vždy se třemi základními pojmy – výsledkem, časem a zdroji – tzv. Trojím omezením (trojimperativem) projektového řízení, kdy je účelem optimálně vyvážit tyto tři základní složky [5].

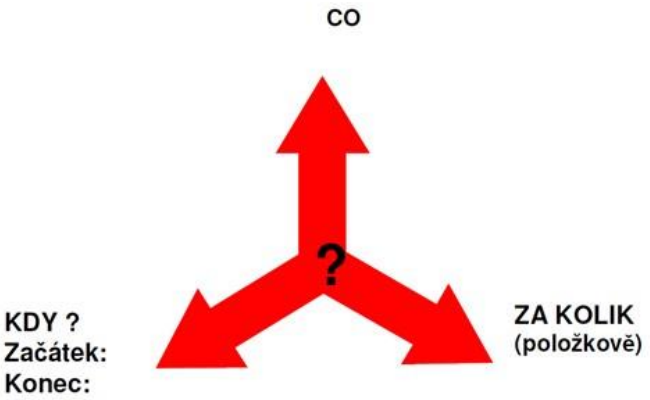
Pro úspěšné dokončení projektu musí jeho projektový manažer správně zvážit všechny tři položky. Mezi těmito často si konkurujícími cíli najít vhodnou rovnováhu. Musíme tedy dle [3] zvážit následující body:

- *Rozsah projektu*. Jaké práce se budou v rámci projektu vykonávat? Jaký jedinečný výrobek očekává náš zákazník po jeho dokončení?
- *Čas*. Kolik času potřebujeme k dokončení projektu a jaký je jeho časový plán?
- *Náklady*. Kolik nás bude ve výsledku stát dokončení projektu? Jaký je jeho rozpočet?

Trojí omezení projektu je znázorněno na obrázku 2.1. Každá z uvedených veličin (rozsah, čas a náklady) mají na začátku projektu stanovený svůj cíl. Ty může projektový manažer určit například podle dřívějších zkušeností nebo se nechá inspirovat obdobnými projekty, které už dříve implementovaly společnosti stejné velikosti.

Při řízení trojího omezení musíme přijímat různé kompromisy mezi cíli rozsahu jednotlivých veličin. Jako příklad můžeme uvést, že odložením konečného času projektu musíme také navýšit náklady, protože musíme zaměstnancům zaplatit odpracované hodiny navíc. Zkušenější projektoví manažeři se zaměřují podle důležitosti na jednu z veličin a dle ní upravují ostatní. Pokud je pro projekt důležitý čas, musí se mu přizpůsobit použité zdroje a pracovní plán našeho produktu. Pokud je ovšem důležitý rozsah, musíme dle toho přizpůsobit čas a rozpočet.

Trojí omezení sice popisuje vzájemné vazby mezi jednotlivými veličinami, ale významnou roli tu hrají i jiné elementy. Klíčovým faktorem může být také uspokojení zákazníka kvalitou našeho produktu. Z toho důvodu se mluví někdy o čtveru omezení [3], kdy se mezi veličiny přidá ještě *kvalita produktu*. Pokud se bude veškerá práce věnovat jenom rozsahu, času a ceně, nemusí dojít k uspokojení zákazníka a veškerá práce může přijít vniveč. Z tohoto důvodu musí projektový manažer během celého průběhu projektu komunikovat se zákazníkem a dbát na to, aby byl s konečným produktem spokojený [3].



Obrázek 2.1: Trojí omezení projektu [4]

Při pracích na projektu je také důležité postupovat podle fází projektu, které jsou podrobněji rozebrány v následující kapitole.

2.2 Životní cyklus řízení projektu

Projekt, jako celek, můžeme z časového hlediska a dle charakteru prováděných činností rozdělit na několik fází řízení projektu, které dohromady tvoří životní cyklus řízení projektu (manažerský pohled) [5]. Před samotným zahájením musí existovat konkrétní zadání. Zadání není přímou součástí

fáze životního cyklu projektu, stejně jako po ukončení jeho vyhodnocení, přesto jsou tyto části pro jeho realizaci velmi důležité. Jednotlivé životní fáze můžeme obecně dle [5] rozdělit na:

2.2.1 Předprojektová fáze

První a nejdůležitější fáze projektu má za účel prozkoumat příležitost pro projekt a posoudit jeho proveditelnost. Jednodušeji řečeno, „předprojektové fáze mají za účel prozkoumat příležitost pro projekt a posoudit proveditelnost daného záměru“[5]. V této fázi se zpracovávají různé analýzy a studie, typické hlavní dva typy výstupu jsou dle [5]:

- *Studie příležitosti*. Tento výstup má zodpovědět otázku, zda je správná doba navrhnout a realizovat zamýšlený projekt. Musí vzít v úvahu situaci v organizaci, situaci na trhu, předpokládaný vývoj trhu i firmy atd. Ve studii můžeme také definovat cíl. Jednou z pomůcek pro správné definování cíle je metoda SMART (Specific, Measurable, Agreed, Realistic, Timed), podle které by měl cíl být specifický, měřitelný, akceptovatelný, realistický a termínovaný. Můžeme zde využít několik metod analýzy. Například SWOT (Strenghts, Weaknesses, Opportunities, Threats) analýzu, která nám identifikuje silné stránky, slabé stránky, příležitosti a hrozby našeho projektu.
- *Studie proveditelnosti*. Pokud se organizace na základě předchozího dokumentu rozhodne projekt uskutečnit, měla by tato studie ukázat nejvhodnější cestu k jeho realizaci a upřesnit obsah projektu. Cílem je vytvořit seznam všech možných cest, které vedou k dokončení projektu. Cesty jsou různě ohodnoceny z hlediska celkových nákladů, potřebného času a tak dále.

Na konci předprojektové fáze bychom měli znát odpovědi na strategické otázky ohledně projektu (k jakému cíli chceme dojít a jakou cestou). Nejdůležitější odpovědí je však, zda se do projektu vůbec pustit. [5]

2.2.2 Projektová fáze

Do projektové fáze patří všechny aktivity odehrávající se od zahájení projektu až po jeho ukončení. Při zahájení projektu je třeba ověřit, případně upřesnit cíl projektu, požadované výstupy a podobně. Toto může pokrýt zakládací listina projektu, která je poté základním projektovým dokumentem.

Při následné přípravě projektu je jmenován tým, který podrobně definuje rozsah projektu, vytvoří plán řízení a jeho harmonogram. Po zahájení realizace je vhodné uskutečnit tzv. *kick-off meeting* („Jedná se o zvláštní typ setkání důležitých zainteresovaných stran, kde je (dle konkrétního pojetí) např. zrekapitulován plán řízení projektu a harmonogram projektu, jsou navzájem seznámeni zástupci zúčastněných stran a především je všem oznámeno, že fyzická realizace začíná.“ [5]). V průběhu realizace je potřeba projekt sledovat a porovnávat jeho průběh s plánem. Pokud dojde k odchylce, je třeba provádět korekční opatření, přeplánovávat a v případě potřeby vytvořit nový plán projektu.

Ve fázi ukončení projektu dochází k fyzickému i protokolárnímu předání výstupů. Projektový tým v této fázi řízení projektu zpracovává závěrečnou zprávu o projektu, ve které je souhrn zkušeností z realizace projektu a případná doporučení do dalších projektů. Rozpustí se projektový tým a ukončí se veškeré procesy projektu [5].

2.2.3 Poprojektová fáze

Poprojektová fáze následuje po ukončení projektu. Nové poznatky a zkušenosti, které jsme získali během realizace projektu, lze využít v dalších projektech. Je třeba vyhodnotit průběh projektu. Cílem vyhodnocení projektu je zjistit, co jsme dělali správně a kde jsme dělali chyby.

Vyhodnocení projektu by mělo probíhat objektivně, protože účelem našeho hodnocení je získání poznatků pro další zlepšování úrovně řízení budoucích projektů a vyvarování se stále stejných chyb. V projektu vyhnojujeme dle [5] následující kritéria:

- Časové skluzy.
- Překročení plánovaných nákladů.
- Odchylky od návaznosti činnosti.
- Důvody provádění různých změn.
- Účinnost a vhodnost použitých metod a různých pomůcek.
- Práci projektového týmu jako celku i jednotlivých členů a vedoucích.
- Efektivitu a funkčnost programu pro podporu projektového řízení.
- Působení směrnic a jiných firemních metodických pokynů.
- Konflikty, krize a mimořádné události.
- Úroveň dokumentace.
- Schopnost a činnosti subdodavatelů.

Po ukončení projektu je doporučeno ustanovit skupinu pro vyhodnocení, připravit časový plán vyhodnocení, shromáždit všechny materiály, které se mohou vztahovat k hodnocenému projektu. Zpracovanou analytickou zprávu popisující závěry projektu projednat a předat zprávu k dalšímu zpracování.

Produkty a výsledky projektu nyní přejdou do provozní fáze, což znamená, že naše organizace k nim může mít nějaké závazky. Na tyto závazky musíme myslet již při návrhu projektu na začátku životního cyklu. [5]

Řízení jednotlivých fází probíhá pomocí procesů, které jsou detailně uvedeny dále.

2.3 Procesy projektového řízení

Řízení projektu je uplatnění veškerých poznatků, dovedností, nástrojů a technik na aktivity projektu takovým způsobem, aby byly splněny požadavky na projekt. Projektový manažer se musí tedy snažit nejen o splnění konkrétních cílů projektu ohledně požadovaného rozsahu, času, nákladů a kvality, ale současně musí v celém procesu usnadnit splnění potřeb a očekávání osob, již jsou do projektu zapojeny nebo jsou jeho aktivitami ovlivněny [3].

Základním objektem řízení je popsaný, definovaný, strukturovaný, zdrojově a vstupy zabezpečený *proces* („soubor vzájemně souvisejících nebo vzájemně působících činností, které dávají přidanou hodnotu vstupům – při využití zdrojů – a přeměňují je na výstup, které mají svého zákazníka. Přičemž vstup (vstupy) představuje vždy definovanou vstupní veličinu (zadání) a výstup pracovní výsledek činnosti“ [10]), jenž je uskutečňován pro konkrétního zákazníka a má jednoznačně stanoveného vlastníka.

Důvod existence každé organizace je vždy charakterizován hlavními procesy [10].

Procesy projektového řízení můžeme rozdělit dle PMI [2] do devíti doporučených oblastí:

- *Řízení integrace.* Popisuje procesy požadované pro zajištění koordinace projektu. Zahrnuje sestavení plánu projektu, jeho realizaci a kompletní koordinaci změn.
- *Řízení rozsahu prací.* Popisuje procesy zajišťující, aby projekt obsahoval všechny požadované práce a aby mohl být úspěšně dokončen. Obsahuje zahájení, plánování rozsahu projektu, řízení změn a ověřování rozsahu projektu.
- *Řízení času*. Popisuje procesy požadované pro včasné dokončení projektu. Skládá se z definování činností a jejich řazení, odhadování dob trvání, sestavení časového rozvrhu a jeho kontroly.
- *Řízení nákladu*. Popisuje procesy pro dokončení projektu v rámci schváleného rozpočtu. Skládá se z plánování zdrojů pro projekt, odhadování a řízení nákladů.
- *Řízení kvality*. Zde jsou popsány procesy pro uspokojení potřeb, kvůli kterým je projekt tvořen. Skládá se z plánování kvality, zabezpečení kvality a jejího řízení.
- *Řízení lidských zdrojů*. Popisuje procesy pro nejefektivnější využití osob v projektu. Skládá se z organizačního uspořádání, náboru pracovníku a rozvoje týmu.
- *Řízení komunikace*. Popisuje procesy pro zajištění včasného a řádného šíření, uchování informací o projektu. Skládá se z plánování komunikací a šíření informací během projektu.
- *Řízení rizik v rámci projektu*. Zde jsou popsány procesy, které rozpoznávají a analyzují rizika a reagují na ně během projektu. Skládá se z identifikace rizik, jejich ohodnocení, tvorby opatření a jejich řízení.
- *Řízení obstarávání v rámci projektu*. Popisuje procesy pro zajištění dodávek a služeb. Obsahuje plánování obstarávání, plánování poptávek, výběr zdrojů atd.

S ohledem na cíle této práce je důležité řízení času, jež je dále podrobně popsáno.

2.4 Časové plánování

Časové plánování (nebo také řízení času v projektu), označuje jednoduše veškeré procesy nezbytné ke včasnému dokončení celého projektu. Dosáhnout včasného dokončení není ale v žádném případě nijak jednoduché. Do řízení času je zapojeno dle [3] šest hlavních procesů:

- *Definice aktivit*. Jedná se o identifikaci konkrétních aktivit, které musí členové projektového týmu a účastníci vykonat, aby byly všechny části projektu dokončeny. Aktivita neboli úkol je pak úsek práce, jenž je obvykle definován ve sktruktuře rozpisu prací *WBS* („WBS (Work Breakdown Structure), jedná se o jednoduchou analytickou techniku, jejímž cílem je rozložit projekt na jednotlivé činnosti až do takové úrovně podrobnosti, aby k nim bylo možné přiřadit odpovědnosti, pracnost a časový horizont“ [6]), a ke kterému je přiřazena určitá očekávaná doba trvání, náklady a požadované zdroje. Výstup tohoto procesu je seznam aktivit a jejich atributů.
- *Řazení aktivit*. Obsahuje identifikaci a dokumentování vztahů mezi aktivitami v projektu. Hlavními výstupy tohoto procesu je síťový diagram časového plánu projektu, požadované změny a aktualizované verze seznamu aktivit a jejich atributů.
- *Odhad zdrojů na aktivit*y. Představuje odhadování objemu prostředků neboli zdrojů (lidí, zařízení, materiálu). Výstupem tohoto procesu jsou požadavky aktivit na zdroje, rozdělení zdrojů, požadované změny a aktualizované verze atributů aktivit a kalendářů zdrojů.
- *Odhad trvání aktivit*. Znamená stanovit odhad doby práce potřebné k dokončení jednotlivých aktivit. Hlavními výstupy tohoto procesu jsou odhady dob trvání aktivit a aktualizované verze atributů aktivit.

- *Vývoj časového plánu*. Jedná se o analýzu posloupnosti aktivit, odhadů dob trvání aktivit a požadavků na zdroje, jejímž cílem je vytvoření časového pánu projektu. Výstupem je tedy časový plán, aktualizované verze požadavků na zdroje, atributy aktivit, kalendáře zdrojů a plán řízení projektu.
- *Řízení časového plánu*. Představuje kontrolu a řízení změn v časovém plánu projektu. Jeho výstupem je měření pracovního výkonu, požadované změny a tak dále.

Celé řízení času v projektu můžeme zlepšit jednak pomocí výše uvedených procesů, jednak pomocí jistých základních nástrojů a technik pro řízení projektů, jako jsou například síťové diagramy, Ganttovy diagramy a analýza kritické cesty [3]. Tyto základní nástroje a techniky jsou podrobně popsány dále.

2.4.1 Seřazení aktivit

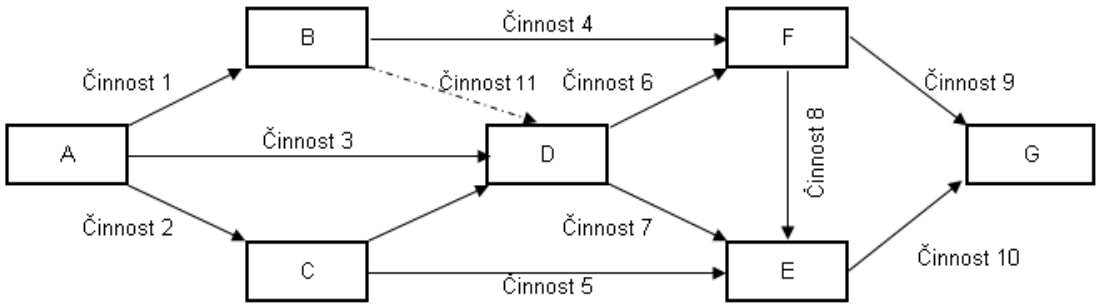
Seřazení aktivit je nalezení logických vazeb mezi aktivitami. Aktivity musí být realizovány v určitém pořadí a musí určitým způsobem navazovat, aby bylo možno sestavit časový diagram.
Vazby jsou dány technologickým postupem nebo mohou být ovlivněny například termíny dodávek. Mezi typické zobrazení dle [5] patří:

- *Konec-začátek*. Předcházející aktivita musí skončit, aby následující mohly začít.
- *Konec-konec*. Předcházející aktivita musí skončit, aby následující mohly skončit.
- *Začátek-začátek*. Předcházející aktivita musí začít, aby následující mohly začít.
- *Začátek-konec*. Předcházející aktivity musí začít, aby následující mohly skončit.

Výsledkem procesu řazení aktivit je dnes obvykle nějaká forma grafického znázornění [5], které je probráno dále.

2.4.2 Síťové diagramy

Síťové diagramy jsou nejvhodnější technikou zobrazování posloupnosti aktivit. Jedná se o znázornění logických vztahů nebo seřazení aktivit v projektu. Sítově diagramy se mohou vyskytovat ve více formách, nejčastější jsou *uzlově definované síťové grafy* (činnosti jsou znázorněny pomocí ohodnocených uzlů, kdy orientované hrany představují závislosti mezi činnostmi) a *hranově definované síťové grafy* (pro znázornění činností jsou použity ohodnocené orientované hrany, uzly představují okamžik začátku a konec činností) [5].



Obrázek 2.2: Hranově definovaný síťový diagram [12]

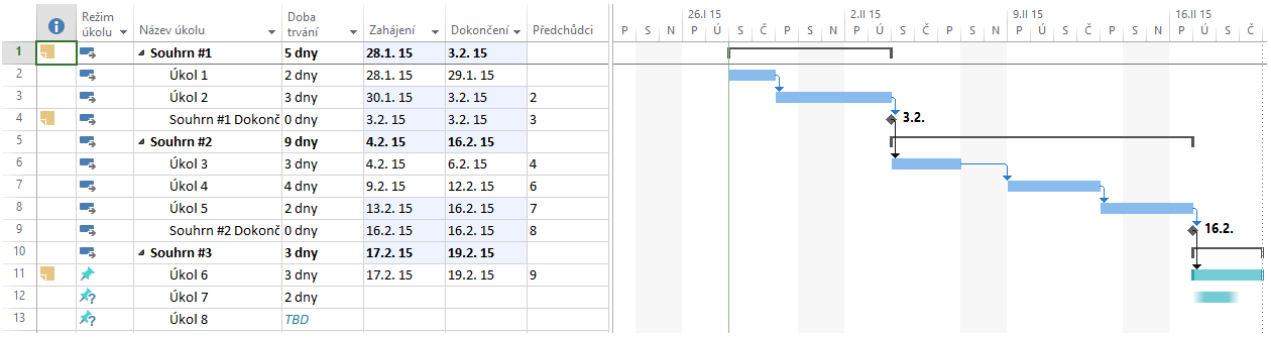
Na obrázku 2.2 si můžeme všimnout nejdůležitějších částí síťového diagramu. Máme zde zobrazeny aktivity (činnosti) očíslované od 1 do 10 a jsou zde také vyznačeny závislosti mezi nimi. Tyto aktivity jsou převzaty ze struktury rozpisu prací (WBS). Šipky pak vyjadřují seřazení aktivit neboli vztahy mezi úkoly, například že činnost 1 je nutné vykonat před činností 4 a tak dále.

Do síťového diagramu není nutné zařadit úplně všechny aktivity ze struktury rozpisu prací, zejména u rozsáhlejších projektů. Někdy stačí zakreslit do síťového diagramu jen souhrnné úkoly [3].

V dnešní době se však s čistě síťovými grafy už příliš nepracuje. Oblíbené je zobrazení, které kombinuje síťový graf s Ganttovým diagramem, kdy nejsou jen vyznačovány vazby mezi aktivitami, ale doplňují je i další informace (např. počet a typ požadovaných zdrojů) [5].

2.4.3 Ganttův diagram

Jedná se o typ úsečkového grafu a je to standardní formát pro grafické znázornění informací o časovém plánu projektu. Jsou v něm uvedeny jednotlivé aktivity projektu a jim odpovídající datum zahájení a ukončení. Aktivity v Ganttově diagramu by měly být v souladu s aktivitami uvedenými ve struktuře rozpisu prací (WBS) [3].



	Režim úkolu	Název úkolu	Doba trvání	Zahájení	Dokončení	Předchůdci
1		Souhrn #1	5 dny	28.1. 15	3.2. 15	
2		Úkol 1	2 dny	28.1. 15	29.1. 15	
3		Úkol 2	3 dny	30.1. 15	3.2. 15	2
4		Souhrn #1 Dokonč	0 dny	3.2. 15	3.2. 15	3
5		Souhrn #2	9 dny	4.2. 15	16.2. 15	
6		Úkol 3	3 dny	4.2. 15	6.2. 15	4
7		Úkol 4	4 dny	9.2. 15	12.2. 15	6
8		Úkol 5	2 dny	13.2. 15	16.2. 15	7
9		Souhrn #2 Dokonč	0 dny	16.2. 15	16.2. 15	8
10		Souhrn #3	3 dny	17.2. 15	19.2. 15	
11		Úkol 6	3 dny	17.2. 15	19.2. 15	9
12		Úkol 7	2 dny			
13		Úkol 8	TBD			

Obrázek 2.3: Ganttův diagram

2.4.4 Metoda CPM

Metoda CPM (Critical Path Method) je používána pro stanovení *kritické cesty* („Kritická cesta v projektu je posloupnost aktivit, která určuje nejdříve možný okamžik dokončení projektu. Je to nejdelší cesta v síťovém diagramu (síťovém grafu), na které nejsou časové rezervy (nebo na níž jsou nejmenší rezervy), a na které se tedy jakékoli zpoždění promítá do zpoždění projektu jako celku.“ [3]). Časová rezerva je „ *množství času, o který může být daná aktivita zpožděná, aniž by ohrozila včasný začátek jakékoli z bezprostředně následujících aktivit*“ [3]. Při práci na projektu za normálních okolností probíhá několik aktivit současně a v síťovém diagramu většiny projektů vede z počátečního do koncového uzlu vícero cest; kritická cesta tedy udává datum dokončení projektu [3].

V projektu musíme nejprve vytvořit správný síťový diagram, který zachycuje vazby mezi aktivitami dle seznamu aktivit v WBS a doplněné odhady doby trvání jednotlivých aktivit nebo Ganttův diagram s vazbami, který také zahrnuje odhady dob trvání [5].

Odhad doby trvání představuje stanovení času potřebné pro vykonání jednotlivých aktivit. Je nezbytné uvažovat o množství zdrojů, které jsou pro danou činnost potřebné, o jejich produktivitě a dostupnost. Odhad trvání na základě jejich pracnosti by měly určovat osoby, které znají technologii příslušné činnosti. Bývají využity techniky odborného úsudku, odhadu na základě podobností, odhadu na základě simulací. Nejčastěji jsou používány postupy [5]:

- Odhad na základě osobní zkušenosti.

- Expertní odhad.
- Odhad na základě dokumentace předchozích projektů.
- Odhad na základě norem.
- Parametrické odhadování (např. dle metrů, počtu kusů a tak dále.).

Pro zjištění kritické cesty se musí sečíst doby trvání všech aktivit na všech cestách v síťovém diagramu a nejdelší z nich je onou kritickou cestou. V rozsáhlém projektu může kritických cest existovat i více.

V každém okamžiku řešení projektu je důležité, aby projektový manažer znal kritickou cestu a díky ní mohl provádět různé kompromisy a rozhodovat se mezi různými variantami. Existuje několik technik, jež pomáhají projektovým manažerům při dosahování kompromisů v časovém plánu, jako je stanovení časové rezervy a *celkové rezervy* („množství času, o které může být daná aktivita opožděna oproti svému zahájení, aniž by ohrozila konečné datum ukončení celého projektu“ [3]) u každé z aktivit v projektu. Pro výpočet časové rezervy aktivit a celkovou časovou rezervu projektu ze síťového diagramu projektu se používají dle [3] dvě metody:

- *Dopředný průchod*. Metoda stanoví nejbližší datum zahájení a dokončení všech aktivit. Okamžik zahájení projektu je roven okamžiku zahájení první aktivity v síťovém diagramu. Nejbližší okamžik dokončení první aktivity je tedy roven okamžiku zahájení plus době trvání této aktivity. Okamžik dokončení je nejbližší možný okamžik, kdy může být daná aktivita dle síťové logiky projektu dokončena. Okamžik dokončení aktivity určí zároveň nejbližší okamžik zahájení všech aktivit, které jsou v síťovém grafu projektu jeho následovníky.
- *Zpětný průchod*. Tato metoda stanoví nejzazší datum zahájení a ukončení dané aktivity. Nejzazší okamžik zahájení je nejpozdnější možný okamžik zahájení dané aktivity, který neohrozí datum dokončení našeho projektu. Nejzazší okamžik dokončení je nejpozdnější okamžik dokončení aktivity, přičemž ještě nebude opožděno dokončení celého projektu.

Kritická cesta, i když se jedná o nejdelší cestu v síťovém grafu, znázorňuje nejkratší dobu, za kterou je možné stihnout dokončit celý projekt. Úkolům na kritické cestě se totiž nemůžeme vyhnout.

Při řízení projektu je důležité stanovit kritickou cestu na začátku projektu, ale také celý plán průběžně aktualizovat a doplňovat. Různé změny vedou ke změně kritické cesty, a tím pádem také k novému odhadu ukončení projektu [3].

2.4.5 Metoda PERT

Metoda PERT (Program Evaluation and Review Technique) jedná se o metodu časové analýzy projektu zahrnující stochastické časové ohodnocení činností projektu. Metoda původně vznikla v roce 1958 pro řízení vojenských programů v USA [11].

Tato metoda síťové analýzy se používá pro odhad doby trvání aktivity v případech, kdy jsou odhady dob trvání zatíženy velkou mírou nejistoty. Metoda PERT tedy v podstatě používá metodu CPM na vážený průměr z pravděpodobných odhadů dob trvání. Podobně jako metoda CPM i metoda PERT vychází ze síťového grafu. V této metodě se dle [3] pracuje s takzvanými pravděpodobnými odhady doby trvání aktivit:

- Optimistická (nejkratší) doba trvání *O*.
- Nejpravděpodobnější doba trvání *N*.
- Pesimistická (nejdelší) doba trvání *P*.

Očekávanou hodnotu a směrodatnou odchylku lze dle [11] určit použitím vztahů:
Očekávaná hodnota:

$$E_{PERT} = (O + 4 * N + P) / 6 \tag{2.1}$$

Směrodatná odchylka:

$$\sigma_{PERT} = (P - O) / 6 \tag{2.2}$$

Očekávaná doba projektu je spočítána jako součet očekávaných hodnot dob trvání na kritické cestě a směrodatná odchylka projektu jako odmocnina ze součtu kvadrátů směrodatných odchylek na kritické cestě. Podle získaných hodnot je možné zakreslit distribuční rozdělení a pravděpodobnost pro celkovou dobu trvání projektu. Výsledné rozdělení má tvar normálního rozdělení pravděpodobnosti. Z tohoto rozdělení je možné získat údaje, s jakou pravděpodobností nepřekročí doba trvání projektu stanovanou mezní dobu trvání projektu a jaká doba trvání nebude překročena se stanovenou pravděpodobnosti [11].

Po pečlivém naplánování projektu se můžeme pustit do jeho realizace, k tomu však potřebujeme dobrý projektový tým, který musí spolupracovat. Tomuto tématu je věnována další podkapitola.

2.5 Řízení lidských zdrojů a komunikace

Projektové řízení není možné vykonávat bez projektového *týmu* („Tým je skupina lidí, kteří mají vzájemně se doplňující schopnosti a kteří si vzájemně zodpovídají za společné úsilí při dosahování společného cíle.“ [9]). Každý projektový tým se skládá z alespoň jednoho manažera a skupiny pracovníků. V projektovém týmu je důležitý každý jedinec, jeho individuální cíle, hodnoty, kritéria a přesvědčení. To nejen umožňuje, ale hlavně významně podporuje vzájemné doplňování svých schopností a sdíleně se rozvíjet ve prospěch celého týmu i jeho jednotlivých členů [9].

Projekty provádí týmy lidí, kteří obvykle byli shromážděni specificky za účelem projektu. Element kompetence „Týmové práce“ v sobě zahrnuje řízení a vedení při vytváření týmu, fungování v týmech a skupinovou dynamiku [5].

Jak již bylo zmíněno, součástí projektového týmu je také projektový manažer. Manažer v organizaci je člověk zodpovědný za svěřenou organizační jednotku nebo jinak vymezenou oblast. Úkolem manažera je řídit. Tedy plánovat, vést, organizovat, kontrolovat jak lidi, tak procesy a další zdroje ve svěřené odpovědnosti v rámci organizace [7]. Musí se ale také hlavně správně rozhodovat.

Rozhodování bezesporu patří mezi jedny z nevýznamnějších činností, které manažeři uskutečňují v rámci managementu. Rozhodování je možné chápat jako jádro řízení a mnohdy je také jako synonymum řízení chápáno. Podle některých pojetí řízení lze dle [8] rozdělit manažerské funkce do dvou skupin:

- *Sekvenční manažerské funkce*. Realizující se v určitém časovém sledu a zahrnující plánování, organizování, výběr a rozmístění pracovníků, vedení lidí a kontrolu.
- *Funkce, které se provádějí průběžně*. V podstatě prostupují sekvenční manažerské funkce. Mezi ty, kromě analýzy činností a komunikace, patří právě rozhodování. Nejvýrazněji se uplatňuje při plánování, neboť jádro plánovacích procesů tvoří právě rozhodovací procesy.

Podstatou rozhodovacího procesu je volba alespoň mezi dvěma možnostmi, dvěma variantami rozhodování. Při manažerském rozhodování se prolínají vědecké přístupy s uměním rozhodovat, tj. s menším či větším podílem intuice [8].

Aby vůbec projektový tým mohl vzniknout, musíme nabrat jeho jednotlivé členy.

2.5.1 Nábor pracovníků

Nábor pracovníků je získávání potřebných lidských zdrojů a jejich přiřazování k práci na projektu. Nábor pracovníků může mít dle [2] následující výstupy:

- *Jmenování (přiřazení) pracovníků projektu.* Projekt je personálně zabezpečen, jestliže byli spolehlivě přiřazeni vhodní pracovníci k práci na projektu. V závislosti na potřebách projektu mohou být pracovníci jmenováni na plný, částečný nebo proměnlivý úvazek.
- *Adresář pracovníků projektu.* Adresář pracovníků projektu obsahuje seznam všech členů projektového týmu a dalších klíčových zainteresovaných stran.

Protože nemusí být k dispozici „nejlepší“ zdroje, musíme projektovému týmu zajistit takové zdroje, které budou splňovat požadavky na projekt [2].

2.5.2 Plánování komunikace

Komunikace v týmu je velmi důležitá pro všechny. Musí vědět, jakým způsobem si předávat úkoly a další informace o stavu projektu. Plánování komunikace se především týká informačních a komunikačních potřeb zainteresovaných stran, a tedy kdo a jaké informace potřebuje, kdy je potřebuje a jak mu budou sdělovány. Pro dosažení úspěchu v projektu je důležité stanovení informačních potřeb zúčastněných a stanovení prostředků pro komunikaci.

Požadavky na komunikaci jsou definovány typem a formátem požadovaných informací společně s analýzou hodnot těchto informací. Mohou zahrnovat například rozdělení odpovědnosti mezi organizací projektu a zainteresovanými stranami, obory, organizační jednotky zapojené do procesu nebo logistiku osob, které budou zapojeny do projektu, a na jakých pozicích budou pracovat. K plánování komunikace jsou potřeba určité komunikační technologie pro přenos informací mezi prvky projektu. Mohou to být stručné rozhovory, ale také projektové porady. Může se jednat o jednoduchý písemný dokument nebo až o rozsáhlý časový rozvrh a databázi informací.

Výstupem plánování komunikace by měl být plán zabezpečení komunikace, jenž bude dle [2] stanovovat:

- Systém sběru informací a jejich uchování.
- Systém šíření informací, který upřesňuje, komu jsou informace určeny a jaké metody budou používány k šíření různých informací.
- Popis informací, které mají být šířeny, včetně formátu, obsahu, stupně podrobnosti a používaných definic.
- Časové rozvrhy šíření informací popisují, kdy budou jednotlivé informace šířeny.
- Metody zpřístupňování informací.
- Metody aktualizace a úprav plánu zabezpečení informací během postupu projektu.

Komunikace v projektovém týmu může být omezena různými faktory. Když se projekt provádí na základě smlouvy, často existují specifická ustanovení smlouvy, která ovlivňují plánování komunikace [2].

2.5.3 Životní fáze týmu

Projektové týmy mají svůj životní cyklus. V každé z životních fází je důležité něco jiného, z čehož vyplývá požadavek jiného stylu managementu. Dynamika týmu většinou vychází z teorie psychologa Bruce Tuckmana [5], která je známá termíny forming – storming – performing:

- *Forming* (formování). Členům týmu není úplně zřejmé zaměření, smysl a účel týmu. Členové týmu „testují“ hranice úkolu, ale také ostatní členy týmu a zejména svého manažera. Z toho plyne, že manažer musí být připraven hodně věcí definovat a vysvětlovat. Jelikož nejsou stanoveny procesy, musí vedoucí říkat, co a kdy se bude dělat. Fáze formování je tedy charakterizována testováním, které slouží k naplnění potřeby orientovat se v prostředí, úkolu a v závislosti na svém manažerovi.
- *Storming* (fáze konfliktů a polarizací). Fáze je typická konflikty a polarizací postojů. Ne všem členům týmu vyhovuje jejich úkol nebo také nemusí být spokojeni s ostatními členy týmu. Fáze je velmi kritická a u spousty projektů je také poslední fází. Je možné zmírnit dopad konfliktů a polarizace a zmírnit jejich intenzitu dobrou a častou komunikací ze strany projektového manažera.
- *Norming* (normální). Členové týmu si ozřejmují společné i individuální hodnoty, kritéria, schopnosti, chování, pracovní nástroje, termíny a podobně, začínají důvěřovat jeden druhému. Z toho vyplývající činnosti se stávají přirozené, hladké a nekonfliktní. Vzrůstá motivace, protože lidé nyní vědí s kým, proč, jak, kde a kdy pracují. Vytváří se také standardy a normy. Členové týmu by se měli podílet na odpovědnosti při přijímání rozhodnutí i na způsobu dosahování jednotlivých dílčích cílů.
- *Performing* (fáze výkonu). V této fázi jsou členové týmu schopni pracovat jako vysoce efektivní a výkonná jednotka i bez svého manažera. Umí takto pracovat a mají k tomu motivaci.

Posléze byla doplněna ještě jedna fáze, a to rozpuštění týmu poté, co bylo dosaženo jeho účelu, nebo jeho pověření novým úkolem [5].

3 Analýza stávajících řešení

V této kapitole porovnám stávající řešení softwarové podpory, vyhledám jejich přednosti a nedostatky. Výsledky analýzy použiji pro návrh výsledné aplikace. Stávající řešení jsem se rozhodl rozdělit na programy pro menší firmy a pro velké podniky. Jednotlivé řešení budu posuzovat dle nabízené škály funkcí, uživatelského rozhraní a možnosti práce více lidí na projektu zároveň.

3.1.1 Nástroje pro menší podniky

Do této kategorie spadají programy, které nejsou určeny pro velké korporace a často se jedná o open source aplikace. Do této kategorie řadím podniky, které za rok realizují maximálně deset projektů.

ProjectLibre

Aplikace ProjectLibre je open source řešení pro projektové manažery, které vzniklo jako nástupce aplikace OpenProj a stejně jako její předchůdce je nejlepší bezplatnou náhradou Projectu od firmy Microsoft. Aplikace je přeložena do velké spousty světových jazyků a nechybí mezi nimi ani čeština. Pro analýzu jsem využil verzi 1.5.9.

Aplikace má jednoduché a přehledné rozhraní, které nabízí projektovým manažerům vše, co potřebují pro plánování projektů. Základní zobrazení časového plánu je realizováno pomocí Ganttova diagramu. Aplikace obsahuje také síťové diagramy a možnost mezi nimi libovolně přepínat. Úkoly jsou hierarchicky uspořádané pomocí WBS a lze také využít PERT metodu. Program dále obsahuje přehledné organizování zdrojů a jejich přiřazování k jednotlivým aktivitám. Velkou výhodou je možnost obousměrného přenášení časového plánu projektu s aplikací Microsoft Project.

Jedinou větší nevýhodou aplikace je, že se s ní dá pracovat jen offline a pokud se chceme s plánem podělit s kolegy, musíme jim náš uložený soubor poslat. Koordinace změn v projektu více uživateli je proto velmi komplikovaná.

Všechny důležité informace, novinky či odkazy ke stažení nalezneme na [13].

Trello

Trello je cloudová služba, jež je vhodná pro správu projektu v projektovém týmu. Pro skupiny a jednotlivce je služba zcela zdarma, pro komerční použití se platí roční poplatek, za který se také zpřístupní pokročilé funkce.

Grafické rozhraní je řešené pomocí přehledných karet, které jsou řazeny do sloupců, nebo jim může být dána barva podle kategorie činnosti. Kartě, která reprezentuje aktivitu, se dají přiřadit pracovníci a úkol rozdělit na více částí. Pokud je nějakému uživateli zadán úkol, je o něm informován a může na něm pracovat, informovat projektového manažera o průběžném pokroku. Dále může přidávat v kartě přidávat komentáře, přílohy pro své spolupracovníky na úkolu.

Na jednotlivých kartách také probíhají diskuze mezi řešiteli a případně projektovým manažerem. Každý úkol má tedy oddělenou vlastní diskuzi, neruší ostatní uživatele a nemusí být složitě řešena přes velký počet emailů.

Trello je dobré pro komunikaci mezi projektovým manažerem a zbytkem týmu, kdy mohou jednoduše a přehledně zadávat práci jednotlivým členům projektového týmu nebo je informovat o různých změnách v plánu, tak pro uživatele, již mají přehledně před sebou jen informace, které potřebují pro svou práci, a jednoduché rozhraní pro komunikaci se svými spolupracovníky a nadřízenými. To je však vše, co může projektový manažer v Trellu dělat, služba neobsahuje žádné

další nástroje pro podporu plánování a hodí se tak pouze pro velmi malé projektové týmy pracujících na jednoduchém projektu.

Základní informace, návody jak s nástrojem pracovat, i nástroj samotný nalezneme zde [14].

3.1.2 Nástroje pro velké podniky

Do kategorie patří programy s velkou škálou funkcí, které jsou převážně placené. V této kategorii se nachází podniky realizující více než deset projektů za rok.

Microsoft Project

Aplikace byla vyvinuta firmou Microsoft a první verze byla vydána v roce 1984. Od té doby prošla mnohými změnami a bylo vydáno mnoho jejich verzí. K analýze jsem použil zatím poslední vydanou verzi 2013. Program není k dispozici zdarma, licence stojí kolem 20 000 Kč, proto jej využívají větší podniky, které na to mají rozpočet a využijí většinu z jeho rozsáhlé škály funkcí [15].

Microsoft Project je na velmi profesionální úrovni s širokým panelem nástrojů. Program podporuje zobrazování plánu projektu v různých formách od jednoduchého seznamu přes síťový diagram po Ganttův diagram, dovoluje mezi nimi libovolně přepínat a změny v jednom zobrazení se přenášení do ostatních. Aplikace podporuje také metody CPM a PERT. Je zde možnost spravovat celé projektové týmy, zadávat úkoly dle termínů, zdrojů a tak dále. Microsoft Project se nabízí v různých formách podle toho, komu má sloužit.

Program nabízí velkou škálu funkcí a pro úplné začátečníky může být jeho uživatelské rozhraní velmi nepřehledné. Ovšem jakmile se v něm člověk zorientuje, poskytuje aplikace podporu pro projektové řízení na opravdu profesionální úrovni. Jeho ohromnou výhodou je týmové plánování a synchronizace plánu projektu. Na projektu tedy může pracovat snadno více lidí naráz. Nové verze umožňují také plánování online a připojení k serveru, což velmi zjednodušuje komunikace a sdílení dat v projektovém týmu.

Další informace a novinky nalezneme zde [15].

Oracle Primavera P6

Program Primavera od společnosti Oracle je velkou konkurencí na trhu pro Microsoft Project.

Primavera je léty prověřený software pro podporu projektového řízení a byl používán pro spoustu velkých projektů. V České republice ho používá například společnost ČEZ.

Program je placený a stejně jako u Microsoft Projectu jeho cena za jednu licenci zasahuje do řádu několika desítek tisíc korun. Licence pro podnik stojí více než 60 000 Kč. Bohužel se mi nepodařilo dostat se k žádné free verzi programu, proto jsem musel program analyzovat podle webových stránek a několika výukových videí.

Stejně jako Microsoft Project také Primavera obsahuje velmi širokou paletu nástrojů a obsahuje všechny potřebné funkce pro řízení projektů. Paleta mi připadá více nepřehledná než u Microsoft Projectu a na jeho grafickém zpracování také není nic, co by uživatele zaujalo. Plán práce si můžeme v Primaveře zobrazit jak v Ganttově diagramu, tak také v síťovém diagramu. Jako velká výhoda Primavery se uvádí možnost nastavit různé role pro uživatele. Můžeme rozdělit lidem v týmů různá práva a funkce, které odpovídají jejich potřebám a funkcím, tyto možnosti má však také Microsoft Project.

Aplikace se připojuje k databázi, takže na projektu může pracovat více lidí naráz a nevyskytují se problémy s přenosem dat.

Základní informace, novinky a výuková videa nalezneme zde [16].

Easy Project

Easy Projekt je webová aplikace jak pro řízení projektu, tak celé firmy. Služba je placená, ovšem oproti Microsoft Project je rozdíl v tom, že uživatel nemusí platit za celý produkt jako celek, ale pouze za jednotlivé služby, které opravdu využívá. Základní řešení bez rozšíření (která se také pohybují v řádech až desítek tisíc korun) stojí podnik do 10 uživatelů necelých 25 000 Kč. Podnik, který potřebuje produkt až pro 50 zaměstnanců, zaplatí necelých 100 000 Kč. Aplikace poskytuje velké spektrum služeb, plánování a řízení více projektů najednou, zadávání, kontrola úkolů, zobrazení vytížení zdrojů, majitelům podniku poskytuje zisky a aktivity jednotlivých oddělení a projektů. Pro zobrazení plánu projektu je využíván Ganttův diagram, síťové diagramy bohužel chybí.

Pracovníci mají jasně zadané úkoly, aplikace nabízí jednoduchou podporu pro vykazování práce, plánování dovolené a další funkce, které podniky potřebují. Jelikož se jedná o webovou aplikaci, je zde snadné sdílet nejrůznější dokumenty napříč týmem nebo celou společností.

Uživatelské rozhraní je hezky navržené, přehledné a intuitivní. Jediné co mu lze vytknout, je absence rychlé palety oblíbených tlačítek. Hlavní paleta je přehledná, ale ke všemu se musí postupně doklikat.

U této aplikace se nemusí řešit přenositelnost plánů, ty jsou uloženy na serveru a je k nim přístup kdykoliv a odkudkoliv. Služba nabízí také vlastní mobilní aplikaci.

Informace k produktu, samotný program i možnost vyzkoušet si ho na 14 dní zdarma nalezneme zde [17].

3.1.3 Zhodnocení analýzy

Všechny programy dostupné na trhu v některém kritériu vynikají, ovšem v dalších bývají až podprůměrné. Pokud bych je měl osobně hodnotit dle škály funkcí, jasným vítězem by se stal Microsoft Project. Jeho cena však nepatří k nejnižším a pro menší podniky může být z toho důvodu nedostupný. Nejjednodušší a nejintuitivnější uživatelské rozhraní má ProjectLibre, ten ovšem, jak již bylo zmíněno, velmi pokulhává ve sdílení dat mezi více lidmi v projektovém týmu. Nejjednodušší práci a komunikaci v týmu nabízí Trello, bohužel to je jeho jediná silná stránka, k plánování rozsáhlejších projektů nemá prakticky žádnou podporu. Primavera vyniká, oproti dříve zmíněným programům, možností přiřadit různé role uživatelům, jinak nabízí stejný standard jako Microsoft Project. V obecném měřítku je favoritem Easy Project, obsahuje celkem širokou škálu funkcí, pracovníci mají přehledně rozdělené úkoly a nemusí se zvlášť řešit přenositelnost dat. Jeho uživatelské rozhraní je také přehledně zpracováno. Nevýhodou je absence síťových diagramů a i když je jeho cena nižší než u Microsoft Projectu, pro menší firmy to může být překážka.

Z těchto důvodů jsem se rozhodl vytvořit vlastní aplikaci, která by měla jednoduché uživatelské rozhraní, uživatelé by se nemuseli starat o přenos dat, podporovala týmovou spolupráci a zároveň nabídla projektovému manažerovi vše, co potřebuje pro plánování a řízení projektu.

4 Návrh Aplikace

V této kapitole podrobně rozeberu návrh aplikace. Při návrhu aplikace jsem využil poznatky získané z analýzy, kterou jsem provedl v předchozí kapitole. Dále je zde popsán návrh aplikace s případy užití a návrh datového modelu aplikace.

4.1 Definice kritérií

Na základě výsledků analýzy stávajících řešení z předešlé kapitoly jsem si určil následující hlavní kritéria:

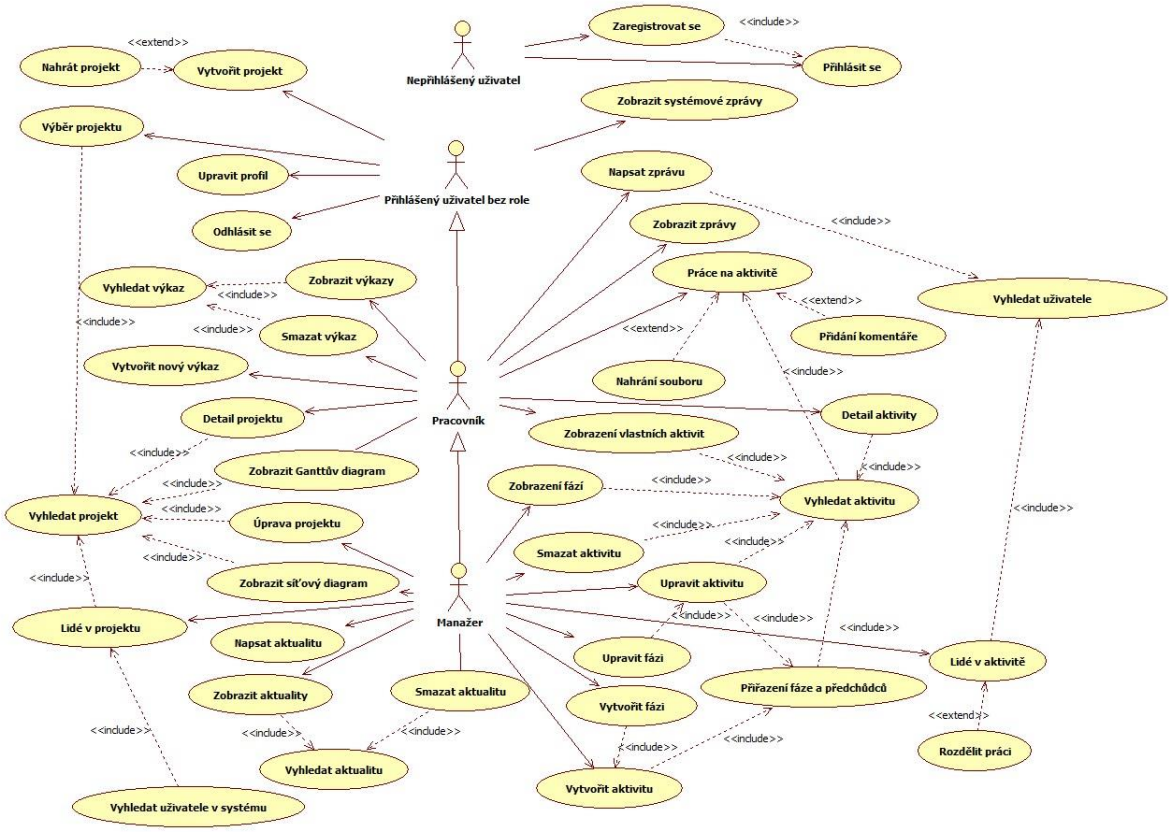
- *Jednoduchost uživatelského rozhraní.* Uživatelské rozhraní musí být jednoduché a intuitivní, uživatel by se v něm neměl ztratit a nalezení požadovaných informací by mu mělo zabrat co nejméně času. V případě zadávání informací by uživatel měl okamžitě pochopit, co se po něm požaduje zadat.
- *Rozdělení do rolí.* V projektovém týmu existují různé role. Minimálně se každý tým rozdělí na pracovníky a na manažera. Aplikace by také měla umožnit tohle dělení. Manažer by měl mít k dispozicí širší škálu nástrojů a možnost vytvářet nebo upravovat všechny úkoly. Pracovníci by měli mít možnost se podívat, jak pokračuje práce na projektu, ale měli by mít možnost zasahovat jen do úkolu, ke kterému jsou přiřazeni.
- *Dostupnost základních nástrojů.* Manažer musí mít všechny základní prostředky pro dobré plánování a řízení projektu. Pro ostatní uživatele je důležitá komunikace v týmů. Aplikace by měla tedy umožňovat zasílání zpráv mezi uživateli a možnost sdílení poznámek a souborů k jednotlivým aktivitám.
- *Přenositelnost.* Jelikož se jedná o aplikaci pro řízení spolupráce, je velmi důležité, aby se uživatel mohl kdykoli a odkudkoli dostat ke svým úkolům nebo je zadat či pozměnit. Z toho důvodu je aplikace navržena jako webová, kdy uživateli postačí jen internetové připojení a jakýkoliv prohlížeč a není vázán na to, že si musí s sebou neustále nosit nějaké vlastní zařízení nebo USB disk.
- *Veřejná aplikace.* Všechna porovnávaná existující řešení byla zaměřená na organizace. Žádné řešení však nebere v potaz, že podporu při plánování a řízení projektů by mohli také využít například studenti při práci na týmových školních projektech. Studenti nemají v každém projektu vždy roli vedoucího, mohou pracovat na dvou různých projektech a v obou mít různou roli. Proto výsledná aplikace bude umožňovat komukoliv založit projekt, čímž získá automaticky roli manažera. Uživatel si však bude kdykoliv moci zvolit projekt, na kterém chce aktuálně pracovat, a podle toho bude mít také nastavenou roli.

Těmto kritériím je podřízen celý další návrh aplikace.

4.2 Možnosti aplikace

V následujících odstavcích popíši, jaké by měla mít aplikace možnosti, aby splňovala výše zmíněná kritéria. Možnosti jsem rozdělil do tří části, nejprve jsem popsal možnosti pro všechny uživatele

a dále pro jednotlivé role, tedy na možnosti manažera a pracovníka. Jednotlivé možnosti zobrazuje diagram užití na obrázku 4.1.



Obrázek 4.1: Diagram užití

Společné možnosti

Každý uživatel by měl mít následující možnosti:

- Registrovat se do systému.
- Přihlásit se do systému.
- Mít možnost vybrat si projekt, na kterém chce nyní pracovat.
- Komunikovat s ostatními uživateli.

Po přístupu do naší aplikace, by měl uživatel mít nejprve možnost registrace, nebo přihlášení do systému, pokud je již registrován. Při registraci budou požadovány základní informace o uživateli, jeho email jako uživatelské jméno a heslo, pro ověření, že se přihlašuje opravdu majitel účtu. Po úspěšné registraci bude uživatel automaticky přihlášen. Po přihlášení by měl mít uživatel možnost upravit svůj profil a odhlásit se ze systému.

Po přihlášení bude uživateli nabídnut seznam projektů k výběru. Projekty budou rozděleny do dvou kategorií. První bude seznam projektů, ve kterých je uživatel manažer, a druhá kategorie bude s projekty, na kterých uživatel pracuje.

Jelikož je aplikace zaměřená na spolupráci v projektovém týmu, je důležité, aby uživatelé měli nástroj pro základní komunikaci. Tuto funkci obstarají zprávy, jež si mohou uživatelé mezi sebou posílat. Uživatelé však nebudou mít možnost posílat zprávu, komukoliv budou chtít, ale jen lidem pracujícím aktuálně na stejných projektech.

Možnosti pro pracovníky

Pro pracovníka je důležité, aby přehledně viděl, kdy a co má udělat, a mohl pohodlně komunikovat s kolegy při řešení určité aktivity. Proto by měl mít následující možnosti:

- Vidět aktuální informace o projektu.
- Mít přehled o svých úkolech.
- Vědět jaké další aktivity řeší jeho kolegové.
- Komunikovat s kolegy při práci.
- Mít možnost vykazovat odpracované hodiny.

Uživatel po výběru projektu by měl mít k dispozici nejnovější novinky o projektu a možnost podívat se na informace uvedené k projektu.

Nejdůležitější je však pro pracovníka sledování aktivit. Měl by mít k dispozici přehledný seznam aktivit, ke kterým je přiřazen, a možnost upravovat jejich stav. Při úpravě stavu by měl mít možnost komunikovat s ostatními kolegy, kteří na aktivitě pracují. Bude tedy možné k aktivitám přidávat komentáře a nahrávat soubory, aby k nim měli přístup všichni, kdo na aktivitě pracují, a nemusí komunikovat výhradně pomocí zpráv.

Pracovník bude mít také možnost vidět i ostatní aktivity a jejich závislosti, ty však nebude moci upravovat, ale jen si zobrazí jejich zadání a stav.

V neposlední řadě by měl mít pracovník možnost vykazovat odpracované hodiny. K tomu budou sloužit výkazy, které se budou vázat k nějaké aktivitě, a uživatel zadá, co a jak dlouho dělal.

Možnosti pro projektového manažera

Projektový manažer je důležitý člen projektového týmu, který musí dobře plánovat a řídit, měl by proto mít k dispozici možnosti, které mu to umožní:

- Přidávat lidi do projektového týmu a odebírat je z něj.
- Upravit a zobrazit informace o projektu.
- Vytvářet a upravovat aktivity.
- Přiřadit lidi k aktivitám a odebírat je z nich.
- Mít přehled o všech aktivitách v projektu.
- Mít možnost informovat všechny členy projektového týmu.

Pokud by byl projektový manažer v projektovém týmu sám, moc by toho asi nezvládl, proto je důležité, aby mohl do projektového týmu přidávat další lidi, kteří jsou registrováni v systému. Samozřejmě by měl mít možnost je také z projektového týmu odstranit. U obou těchto událostí by měl být pracovník informován zprávou.

V průběhu projektu se mohou některé základní věci o projektu pozměnit, projektový manažer by měl mít proto možnost je upravit. Měl by mít také možnost si je zobrazit a mít přehled o všem, co se děje. Měl by tedy vidět všechny aktivity, nejlépe ve vhodné formě síťového diagramu.

Nejdůležitější je však, aby projektový manažer měl možnost vytvářet aktivity, napsat jejich zadání a dobu trvání. Pokud nebude přesně znát dobu trvání, měl by mít k dispozici metodu PERT pro výpočet odhadu. Aktivitu by měl mít možnost kdykoliv upravit.

K jednotlivým aktivitám je potřeba přiřadit pracovníky. Proto by měl mít projektový manažer k dispozici jednoduchý nástroj, kdy bude přidávat/odebírat lidi u jednotlivých aktivit. Všichni

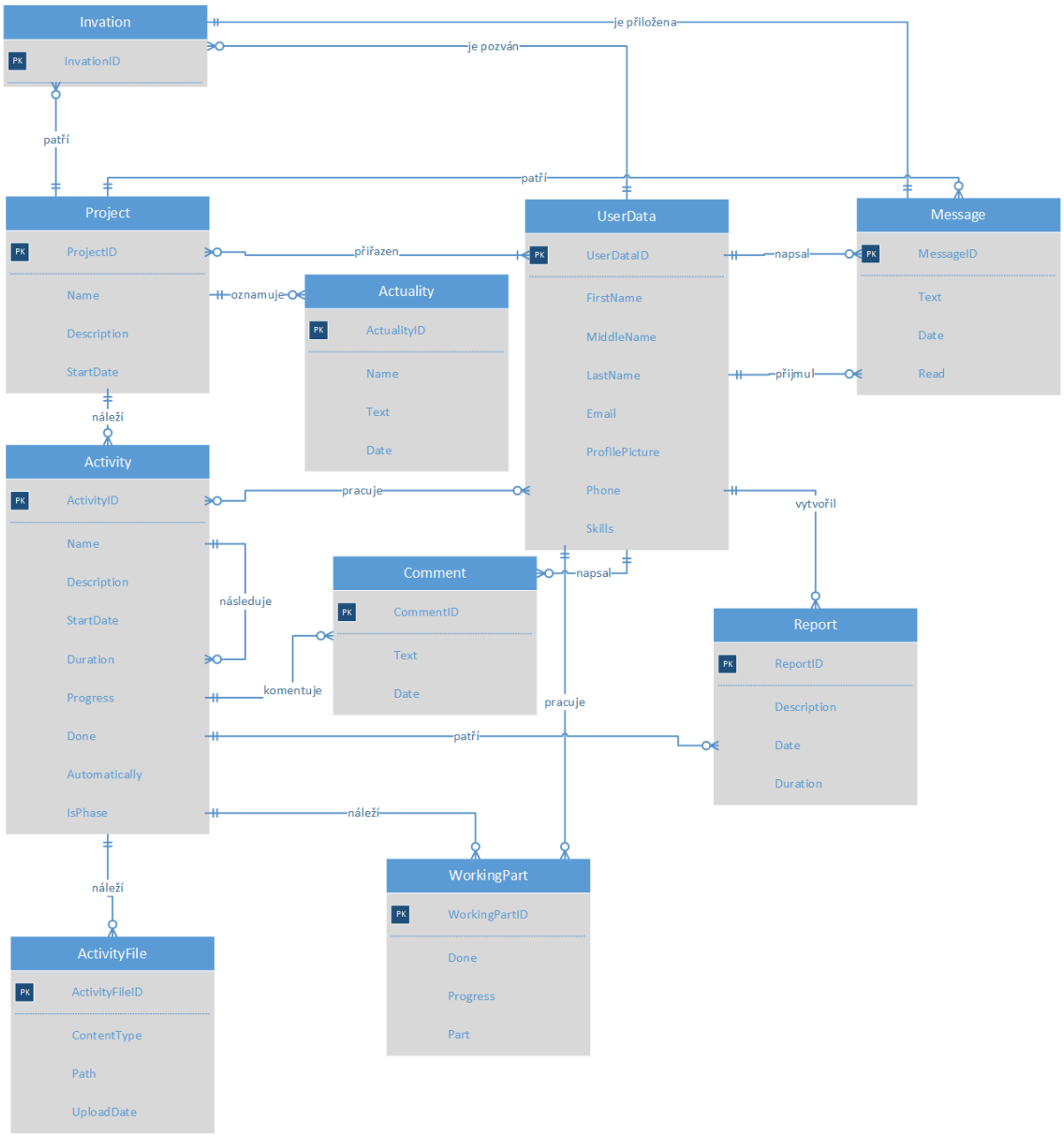
pracovníci nebudou vždy na dané aktivitě pracovat rovným dílem. Proto by měl mít manažer možnost nastavit jednotlivým pracovníkům, jakou část na aktivitě budou odpracovávat.

Přehled o aktivitách by projektový manažer neměl mít jen v informacích o projektu, ale také přehledně poskládané v Ganttově diagramu, kde by viděl, jak na tom jednotlivé aktivity jsou, která potřebuje být dokončena, aby se mohlo pracovat na další a tak dále. Pro přehlednost by měl mít možnost dělit aktivity do jednotlivých fází.

Pokud dojde při práci na projektu k nějaké změně, významné události nebo naplánování schůze projektového týmu, je nutné, aby to mohl dát projektový manažer zbytku projektového týmu na vědomí. K tomuto účelu by měl mít možnost psaní aktualit, které se ostatním pracovníkům zobrazí, a budou tedy informování o všem důležitém, co se stane.

4.3 Návrh datového modelu

Po navržení možností aplikace již víme, jak bude fungovat, a tedy i jaká data budeme potřebovat uchovávat v databázi. Ke znázornění dat jsem využil *ER diagram* (Entity–Relationship diagram), který je zobrazen na obrázku 4.2 a jehož jednotlivé entity jsou popsány dále.



Obrázek 4.2: ER diagram

Project

Entita reprezentuje samotný projekt a sdružuje entity, které do daného projektu náleží pomocí cizích klíčů. Samotná entita si uchovává pouze základní informace, tedy název projektu, popis a datum zahájení.

UserData

Entita UserData patří mezi nejdůležitější entity. Obsahuje základní informace o uživateli, jako jméno, prostřední jméno a příjmení. Dále uchovává kontaktní údaje, email a telefon, dovednosti uživatele a cestu k jeho profilové fotce. Na tuto entitu se vyskytují odkazy v dalších entitách, které potřebují identifikovat vlastníka dané entity. Každý uživatel může pracovat na více projektech a na každém projektu může pracovat více pracovníků.

Activity

Entita představují aktivitu. Aktivita si uchovává svůj název, popis, zda je dokončena, jestli je plánována ručně nebo automaticky, datum zahájení a datum ukončení. Také si uchovává, zda jsou k ní přiřazeny nějaké dílčí aktivity. Tato entita má vztah s projektem, kde každá aktivita patří do jednoho projektu.

Každá aktivita může mít ještě také jednu, nebo i více dílčích aktivit.

WorkingPart

WorkingPart reprezentuje část práce na aktivitě. Ke každé pracovní části je přiřazen uživatel, který ovšem může pracovat na více pracovních částech. Entita má také vztah k aktivitě, ke které se práce váže. Opět se váže k jedné jediné aktivitě a může být rozdělena na více části. Entita si pamatuje, kolik tvoří procent práce na aktivitě, její stav a zda byla práce dokončena.

Comment a ActivityFile

Entita Comment reprezentuje komentář k aktivitě a ActivityFile soubor. Obě entity se váží k jedné jediné aktivitě, která může mít více komentářů, tak více souborů.

Komentář si uchovává svůj text a datum, kdy byl napsán. Dále má vztah s entitou Activity, ke které náleží, a UserData podle uživatele, který komentář napsal, jelikož komentář se může vázat jen k jediné aktivitě a mohl ho napsat jen jeden uživatel.

ActivityFile si pamatuje název souboru, jeho typ, datum nahrání a cestu k němu, také se váže k jedné jediné entitě Activity.

Invation

Entita reprezentuje pozvánku pro uživatele do projektu. Váže se tedy k jednomu projektu a ke každému projektu může být rozesláno více pozvánek. Pamatuje si, kterému uživateli je zaslána, a každý uživatel může dostat více pozvánek do různých projektů.

Message

Message je entita, která reprezentuje zprávu, již si mohou uživatelé posílat mezi sebou. Entita si uchovává svůj text, datum, kdy byla napsána, a informaci, zda byla již přečtena nebo ne. Každou zprávu může napsat jeden uživatel a je určena dalšímu uživateli, tato entita má hned dva vztahy s entitou UserData. Pomocí zprávy je také možno zaslat pozvánku. Jednou zprávou lze odeslat jen jednu pozvánku.

Report a Actuality

Poslední dvě entity představují výkaz a aktualitu.

Výkaz si pamatuje svůj popis, kolik bylo stráveno času při práci a datum, kdy byl podán. Každý výkaz se vztahuje k jedné aktivitě a může ho napsat jeden uživatel.

Aktualita si uchovává pouze název a svůj text. Není potřeba si pamatovat uživatele, protože psát aktuality může pouze projektový manažer. Aktualita má vztah s entitou Project a každá aktualita může patřit jen k jednomu projektu.

Po pečlivém naplánování jednotlivých stránek a sestavení datového modelu se můžeme pustit do samotné implementace projektu, kterou jsem popsal v následující kapitole.

5 Implementace

Kapitola se zabývá implementací webové aplikace pro týmovou spolupráci a správu úkolů. Nejprve jsou popsány použité nástroje, poté následuje popis implementace datové vrstvy a nakonec jsou popsány základní funkcionality a knihovny, které k tomu byly použity.

5.1 Použité nástroje

Na začátku jsem musel vybrat vhodný jazyk a platformu, nad kterou webovou aplikaci budou vyvíjet. Rozhodl jsem se pro jazyk C#. Jedná se o objektově orientovaný jazyk, vyvinutý firmou Microsoft, která ho založila na jazycích Java a C++. V jazyce C# lze vyvíjet webové aplikace pomocí APS.NET, který jsem také zvolil pro vývoj aplikace a je podrobněji rozebrán v následující podkapitole.

5.1.1 Entity framework

Jako první krok jsem musel vytvořit databázi. K tomu jsem se rozhodl použít Entity framework, který umožňuje *ORM* (Objektově-Relační Mapování), díky kterému získáme z databáze celé objekty a ne jenom pole hodnot. Dotazy v jazyce *SQL* (Structured Query Language) nám framework v pozadí generuje sám, a tudíž s nimi přímo vůbec nemusíme pracovat. Databázi vidíme jako kolekci objektů, a můžeme tak s ní i pomocí programovacího jazyka pracovat [18].

K vytvoření databáze máme dva způsoby. První je takzvaný Database First, kdy nejprve vytvoříme databázi a framework nám k ní vygeneruje třídy a *kontext* (třída zodpovědná za interakci s daty jako objekty). Já jsem použil druhý způsob, kterému se říká Code First. Tento způsob spočívá v tom, že si napíšeme třídy, kontext a Entity Framework nám podle nich vytvoří databázové tabulky.

5.1.2 ASP.NET

ASP.NET je populární platforma pro vývoj webových aplikací od firmy Microsoft a je součástí .NET Frameworku. Díky tomu můžeme při vývoji webové aplikace využít stejnou škálu nástrojů jako při vytváření klasických aplikací. Díky .NET Frameworku, není nutné programovat aplikace jen v jazyce C#, ale můžeme si zvolit jakýkoli, který splňuje pravidla *CLS* (Common Language Specification), například Visual Basic [19].

Velkou výhodou aplikace napsané v ASP.NET je, že se neinterpretuje, ale komprimuje [19], což má za následek, že výsledná aplikace je rychlejší, než kdyby byla napsána v čistě skriptovacím jazyce. Aplikace prochází celkem dvěma kompilačními etapami. Nejprve se kód zkompiluje do přechodného jazyka *MSIL* (Microsoft Intermediate Language), zkráceně IL. První krok je příčinou toho, jak již bylo zmíněno, že v .NET Frameworku je možné použít více programovacích jazyků. Všechny jazyky se proto zkompilují do fakticky identického kódu. Druhá fáze nastane až před tím, než se stránka zavolá. V tom okamžiku se kód IL zkompiluje do nativního nízkoúrovňového strojového kódu. Tato etapa se nazývá JIT (Just-In-Time).

Aplikace ASP.NET se nekompilují při každém zobrazení stránky. Kód IL se vytvoří pouze jednou a generuje se znovu, jen pokud dojde k modifikaci zdrojových kódů. Zkompilovaný IL kód se nazývá asembly. Jsou také k dispozici nástroje, které nám dovolí předběžnou kompilaci do strojového kódu, jakmile je aplikace nasazena na ostrý server [19].

Aplikace v ASP.NET můžeme vytvářet dvojím způsobem. První je pomocí standardního modelu webových formulářů, který je však nyní na ústupu. Druhý, v dnešní době využívanější a velmi odlišný od prvního způsobu, je pomocí architektonického vzoru MVC.

Visual Studio

Webovou aplikaci v ASP.NET můžeme psát i v obyčejném textovém editoru. Takový postup je však velmi pracný a může se při něm učinit hodně chyb. Webové aplikace v ASP.NET se vyvíjejí ve Visual Studiu od firmy Microsoft.

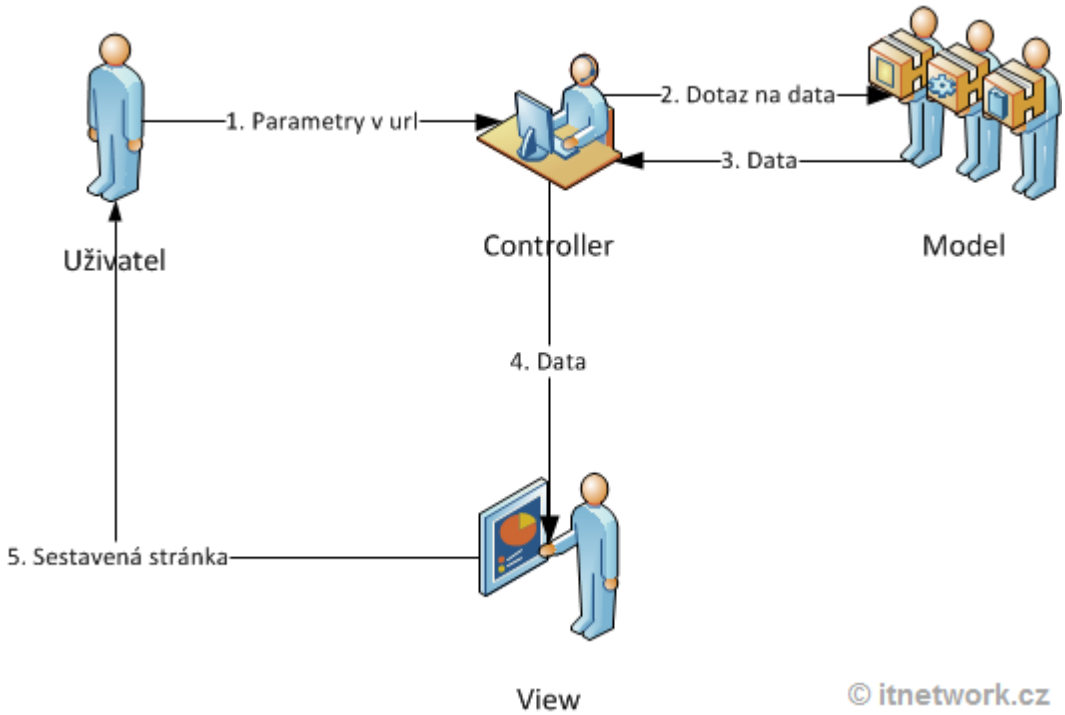
Visual Studio je profesionální vývojářské prostředí, které podporuje různé designérské sady, ladící nástroje, intellisence a tak dále. Pro nás je ale důležité, že obsahuje vlastní webový server, který je určený pro snadné testování aplikace.

Ve Visual Studiu pracujeme vždy v nějaké solution, které vždy obsahuje jeden nebo více projektů. Visual Studio nám může vytvořit náš ASP.NET projekt, kdy nám sám vytvoří základní konfigurace a my můžeme bez zbytečné námahy a všelijakých úplně základních konfigurací začít vyvíjet naši aplikaci. K dispozici je i šablona, kdy je nám vygenerována ukázková aplikace, kterou si můžeme upravovat a stavět na základech, jež nám byly přichystány. Například může být výhodné zachovat odladěný systém autentizace a autorizace.

5.1.3 MVC architektura

Základní myšlenkou této architektury je rozdělit aplikaci do tří logických částí tak, aby bylo možné každou z nich samostatně modifikovat. Zároveň by mělo platit, aby byl dopad na ostatní dvě části co nejmenší. Tyto tři části jsou Model, View a Controller, odtud zkratka MVC. Jednotlivé části si nyní jednotlivě představíme [20]:

- *Model*. Model se stará o logiku naší aplikace. Nepatří sem jen doménový model, ale také například operace s daty, validace, různá pravidla nebo databázové dotazy. Funkce modelu je přijmout nějaké parametry a na jejich základě vrátit požadovaná data. Model nemá žádné ponětí o tom, odkud požadavek na data přišel, ani jak budou následně data zpracovávána.
- *View*. View, který si můžeme přeložit jako pohled, se stará o zobrazení výstupu uživateli. Zároveň však také naslouchá akcím uživatele a na základě jeho příkazů jsou volány požadované metody v controlleru. Nejčastěji se jedná o šablonu s *HTML* (HyperText Markup Language) stránkou s možností vkládat příkazy nebo cykly v jiném jazyce. View, stejně jako model neví, odkud data, která má zobrazit, přišla, stará se jen o samotné zobrazení. Obsahuje však také, alespoň minimální, logiku. Například kontroluje, zda hodnota, kterou má zobrazit, není prázdná.
- *Controller*. Poslední část, controller, je jakýsi spojovací prvek, který drží celý systém pohromadě a propojuje jednotlivé části. Jako příklad si můžeme uvést, že uživatel zadá parametry v *URL* (Unique Resource Locator). Ty zachytí takzvaný router. Například z http://www..../Aktivita/Detail/8, rozeznáme, že voláme controller AktivitaController, požadujeme metodu Detail a id aktivity je 8. Controller požádá model o data k aktivitě. Po jejich přijetí, je controller předá do view, které je zobrazí uživateli. Podrobněji je tento životní cyklus stránky ukázán na obrázku 4.1.



Obrázek 5.1: Ukázka MVC v praxi [20]

Architektura nám také díky rozdělení na tři logické části usnadňuje vývoj aplikace a zpřehledňuje náš zdrojový kód. Přehled architektury můžeme vidět na obrázku 5.1.

5.1.4 Repozitáře

U malých aplikací nevadí, pokud z controllerů přistupujeme přímo k datům. U větších aplikací už však začne docházet například k duplicitě kódu, pokud jsme se dopustili chyby, bude ji mnohem těžší opravit a aplikace bude mnohem hůře testovatelná. Z toho důvodů se používají takzvané *repozitáře*. Jejich pomocí se oddělí logika načítání dat a jejich mapovaní na entity do modelu. Budeme tedy vždy vyžadovat data z různých míst, stejným způsobem. Controller se také nebude muset nadále starat, jestli data dostane z databáze, nebo se například načtou ze souboru. Tuhle práci nyní budou mít na starosti repozitáře [21].

V jazyce C# se repozitáře implementují pomocí *rozhraní* (interface). Rozhraní obsahuje pouze popis metod, vlastností, událostí nebo indexování. Pokud nějaká třída implementuje rozhraní, musí implementovat členy rozhraní zadané v jeho definici [22]. V controlleru nám tedy stačí volat metody pomocí rozhraní a nemusíme ani vědět, s kterou třídou momentálně komunikujeme.

5.1.5 Dependency injection

Dependency injection je návrhový vzor, jenž slouží k vytváření závislostí mezi jednotlivými částmi systému. Pokud objekt využívá služby jiného objektu, musí se starat o jeho inicializaci, následné uvolnění a tak dále. Pokud využijeme Dependency injection, jsme od této povinnosti osvobozeni a stará se o ní za nás *kontejner*, který obsahuje všechny závislosti [23].

Dependency injection jsem při implementaci aplikace použil v controllerech. Každý controller obsahuje reference na několik repozitářů respektive na jejich rozhraní, jak bylo zmíněno v minulé podkapitole, a jsou inicializovány v *konstruktoru* (metoda, která se zavolá při vytváření instance třídy). Nikde v kódu však konstruktor nevolám a nepředávám mu objekty, které implementují rozhraní daného repozitáře, ale naplní mi je kontejner, ve kterém mám zaznamenáno, za které

rozhraní se vytvoří jaký objekt. Tomuto vzoru využití Dependency injection se říká *injekce konstruktorem* [23]. Výhoda tohoto přístupu je, že kdybych se rozhodl změnit databázi, stačí mi vytvořit nové implementace repozitářů, změnit záznam v kontejneru a v celé aplikaci se začnou využívat nové metody. ASP.NET MVC však sám Dependency injection nativně nepodporuje, ale existují různé přídavky, které přidávají jeho podporu. Já jsem se rozhodl využít *Simple Injector*. Jedná se o jednoduchou knihovnu, kterou je možné využít v ASP.NET, ale také například ve Windows Phone a dalších podporovaných platformách v .NET.

5.1.6 Knihovna JQuery

Knihovna JQuery se používá pro vývoj vysoce interaktivních internetových aplikací. Klade důraz na interakci mezi HTML a jazykem JavaScript. Knihovna je rychlá, přesná a zjednodušuje tvorbu a správu událostí, animací a dalších komponentů. Dále můžeme vybírat jednotlivé elementy ve stránce a měnit jejich vlastnosti [25].

Knihovna JQueery nám také ulehčuje práci s technologií *AJAX* (Asynchronous JavaScript and XML). Využívá se například pro provádění požadavků na serveru, ale tak, aby nedošlo k přerušení ostatních úloh spuštěných ve webovém prohlížeči [25].

5.2 Implementace vybraných částí

Při implementaci jsem se setkal s několika částmi aplikace, jejichž implementace je zajímavá. Několik z nich jsem vybral a popsal v této podkapitole. Hlavní pozornost jsem věnoval částem, které usnadňují manažerům a pracovníkům jejich práci v aplikaci. Ve většině těchto částí jsem využil JavaScriptovou knihovnu JQuery.

5.2.1 Ganttův diagram

Jak bylo zmíněno, Ganttův diagram je standardní formát pro grafické znázornění informací o časovém plánu projektu a proto je potřeba, aby ho měl projektový manažer při plánování k dispozici. K ASP.NET existuje balíček pro vykreslení Ganttova diagramu, ten je však placený, proto jsem začal hledat nějakou open source JavaScriptovou knihovnu a velmi mě překvapilo, že jich existuje jen málo. Nakonec jsem se rozhodl pro knihovnu *dhtmlxGantt* [26], která podporuje jak samotné vykreslení diagramu, tak práci v něm, zahrnující vytvoření aktivit, jejich posun v čase, úpravu stavu a tak dále.

Nejprve jsem se musel podívat na dokumentaci a nastavit inicializaci, například po jakých jednotkách se má graf zobrazovat, jeho velikost a tak dále. Podstatnou částí však bylo naprogramovat metody, které umožní vykreslit aktivity z databáze. Do databáze jsem oproti původnímu návrhu musel přidat entitu *GanttLink*, která si uchovává primární klíče aktivit, již mají mezi sebou vztah. Dále si pamatuje, k jakému projektu aktivity náleží.

Nejprve jsem implementoval v controlleru *Activity* metodu *GetGanttData*, která vrátí pro knihovnu v požadovaném formátu aktivity a informace o nich. Knihovna podporuje dva formáty *JSON* (JavaScript Object Notation) a *XML* (Extensible Markup Language). Já jsem zvolil formát XML, pro každou aktivitu se vytvoří element a přidají se mu požadované atributy.

Druhá a složitější metoda se jmenuje *UpdateGanttData*, tato metoda si získá a rozdělí data z Ganttova diagramu a podle nich upraví stav v databázi.

Při implementaci metod pro práci s Ganttovým diagramem jsem se inspiroval článkem [27].

5.2.2 Síťový diagram

Nalézt vhodnou knihovnu pro vykreslení síťového diagramu bylo ještě složitější než pro Ganttův diagram. Všechny zajímavé knihovny byly opět placené, tak jsem začal hledat knihovny pro vykreslování grafů, které by se daly upravit. Nakonec jsem se rozhodl pro knihovnu *JointJS* [28], ke které jsem vytvořil prvky, jež znázorňují aktivity, milníky a jednotlivé fáze a naprogramoval metodu, která jednotlivé prvky přehledně poskládá podle vztahu mezi nimi.

5.2.3 Přidávání lidí do projektu a jejich přiřazení k aktivitám

Aplikace má být zaměřená na týmovou spolupráci. Než začne samotná spolupráce, musí projektový manažer nejprve sestavit svůj projektový tým. Proto jsem implementoval přidávání lidí do projektového týmu a jejich přidělení k aktivitám. Jednotlivé části aplikace jsem popsal níže.

Přidávání lidí do projektu

V této části režimu jsem dal k dispozici projektovému manažerovi dva seznamy. První seznam obsahuje lidi, kteří již jsou v projektu a projektový manažer je z něho může odebrat. Ve druhém seznamu se vyskytují lidé, kteří v projektu přiřazení nejsou. Není vhodné, aby se projektovému manažerovi zobrazili najednou všichni lidé, kteří jsou registrovaní a nejsou v projektu, takových lidí může být obrovské množství.

Na stránce jsem přidal textbox, do kterého může začít projektový manažer psát. Textbox je součástí formuláře, který jsem vytvořil pomocí objektu *Ajax.BeginForm*, kterému jsem nastavil obslužnou metodu *_UsersList*, která se nachází v controlleru *Project*, a bylo mu řečeno, aby vytvořil požadavek *POST* (metoda vkládá odesílaná data přímo do těla požadavku *HTTP* (Hypertext Transfer Protocol)) [25]. Dále jsem implementoval skript v jazyce JavaScript, který hlídá, zda byla na texboxu stisknuta klávesa. Pokud byla, je formulář odeslán a metoda _UserList nám naplní nový seznam uživatelů. Metoda si nejprve připraví seznam uživatelů a rozliší jej od seznamu uživatelů přiřazených k projektu. Metoda přijme text, který projektový manažer napsal do textboxu, a pokusí se text vyhledat ve jméně nebo emailu uživatelů v seznamu. Ty uživatele, u kterých požadovaný podřetězec nalezne, vrátí v novém sezamu. Díky technologii AJAX vše probíhá na pozadí, proto projektový manažer může psát do texboxu a seznam uživatelů se mu dynamicky mění.

Přiřazení lidí k aktivitě

Tuto část jsem opět implantoval pomocí dvou seznamů. V jednom se nachází uživatelé, kteří jsou již přiřazeni k dané aktivitě, ve druhém ostatní uživatelé začlenění do projektu. Oba seznamy jsem zobrazil pomocí značkovacího jazyka HTML jako obyčejné seznamy, jen jsem ke každému přidal, jako skrytý element, jeho identifikátor. Zde jsem použil funkci z knihovny JQuery *sortable*. Tato funkce nám umožní pomocí tahu myši přehazovat prvky mezi seznamy. Uživatel tedy může snadno přiřadit lidi k aktivitě nebo je z ní odebrat. U lidí, kteří jsou přiřazeni k aktivitě, se zobrazí textbox, do kterého manažer napíše, kolik práce na aktivitě mají odpracovat. Po odeslání formuláře metoda *PeopleInActivity*, která se nachází v controlleru *Activity*, zpracuje provedené změny. U aktivity odstraní přiřazené uživatele, převezme pole identifikátorů uživatelů, kteří k ní mají být přiřazení. Uživatele vyhledá k následné aktivitě přidělí a přiřadí jim práci na aktivitě.

5.2.4 Práce na aktivitě

Při práci na aktivitě jsem vytvořil pro uživatele několik prvků. První byl posuvník, pomocí kterého může uživatel upravit, na kolik procent odpracoval svojí část. Posuvník jsem vykreslil opět pomocí knihovny JQuery. Napsal jsem skript, který na pozadí hlídá, zda bylo posuvníkem pohnuto. Pokud ano, je odeslán formulář s aktuálním stavem a ten se uloží. Opět je využita technologie AJAX, takže uživatel nepozná, že proběhla komunikace se serverem.

Ke každé aktivitě jsem dal možnosti přidávat komentáře, díky nimž si mohou pracovníci na stránce nechávat poznámky nebo vzkazy. Uživatel může napsat nový komentář a stisknutím tlačítka odeslat, se odešle opět AJAX formulář na server. Nebylo by však praktické, kdyby se zobrazily všechny komentáře najednou. U aktivit, na kterých by se pracovalo delší dobu, by komentářů mohlo být mnoho a stránka by se stala nepřehlednou. Proto se nejprve zobrazí jen posledních 5 komentářů a na stránce je k dispozici tlačítko pro dalších 5. Jeho stisknutím se pomocí skriptu v jazyce JavaScript navýší o 5 skrytá proměnná, která se pomocí AJAX formuláře odešle na server a ten podle ní vrátí požadovaný počet komentářů.

U aktivit si mohou také uživatelé ukládat soubory. Ty jsou ukládány pomocí klasických formulářů. Po odeslání data přijme metoda *UploadFile* v controlleru *Activity*. Metoda přijme soubor, a pokud k aktivitě dosud nebyl nahrán žádný soubor, vytvoří pro ni novou složku. Poté zkontroluje, zda není již soubor se stejným názvem u aktivity přiřazen. Pokud ano, jen ho uloží. Pokud není, vytvoří nový objekt *AcitivityFile*, kam uloží jeho název, cestu, typ a poté ho přiřadí k aktivitě.

6 Popis jednotlivých stránek

V této kapitole popisuji jednotlivé stránky, které obsahuje výsledná aplikace. Stránky jsou rozděleny do tří kategorií, podle role uživatele v projektu.

6.1.1 Společné stránky

V projektu se vyskytují stránky, jejichž podoba není závislá na tom, zda je uživatel přihlášený nebo jakou roli v projektu má. Tyto stránky jsou tedy pro všechny případy společné.

Úvodní stránka

Na úvodní stránku se dostane každý uživatel po otevření ve webovém prohlížeči. Úvodní stránka poskytuje uživateli hlavní informace o aplikaci, například k čemu slouží a jaké obsahuje možnosti či nástroje. Nejdůležitější je však možnost registrace a přihlášení. Přihlášení je umístěno přímo na úvodní stránce a je možné dvěma způsoby. První je zadat svůj email, heslo a kliknutí na tlačítko přihlásit se. Druhá možnost, je pomocí sociální sítě Facebook.

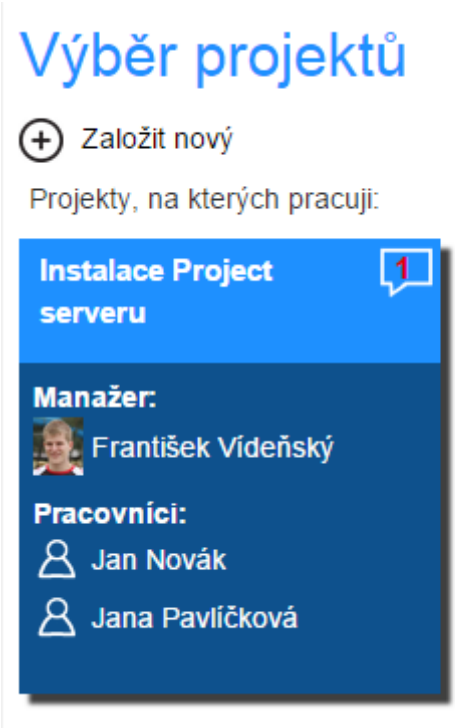
Registrace

Registraci do systému je možné provést také dvěma způsoby. První je, že se z úvodní stránky odkážeme na stránku registrace, kde vyplníme základní údaje. Povinné položky pro registraci jsou křestní jméno, příjmení, email a heslo. Heslo je požadováno dvakrát, aby se předešlo překlepům ze strany uživatele. Po vyplnění povinných údajů může uživatel nepovinně vyplnit ještě telefonní číslo, své prostřední jméno a dovednosti. Dovednosti mohou sloužit manažerům při rozhodování, komu jaký úkol přiřadí. Uživatel si také může nahrát svoji profilovou fotografii. Uživatel si také může vybrat, zda chce přeposílat zprávy na email.

Druhá možnost registrace je pomocí sociální sítě Facebook. Registrace se provede při prvním pokusu o přihlášení přes Facebook, uživateli se vytvoří účet, ke kterému uživatel doplní další potřebné údaje a poté bude registrace dokončena.

Výběr projektu

Po úspěšném přihlášení do systému je uživatel přesměrován na stránku s výběrem projektu. Na stránce se nachází tlačítko založit nový projekt a dva druhy lístečků s projekty. První lístečky reprezentují projekty, které uživatel založil, je zobrazen jeho název, popis, pracovníci, kteří se na něm podílejí, a počet nových zpráv v projektu. Druhý typ jsou projekty, k nimž je uživatel přiřazen jako pracovník, lísteček obsahuje stejné atributy jako první, a ještě navíc jméno projektového manažera. Po výběru projektu je uživatel přesměrován na jeho hlavní stránku.



Obrázek 6.1: Ukázka výběru projetu

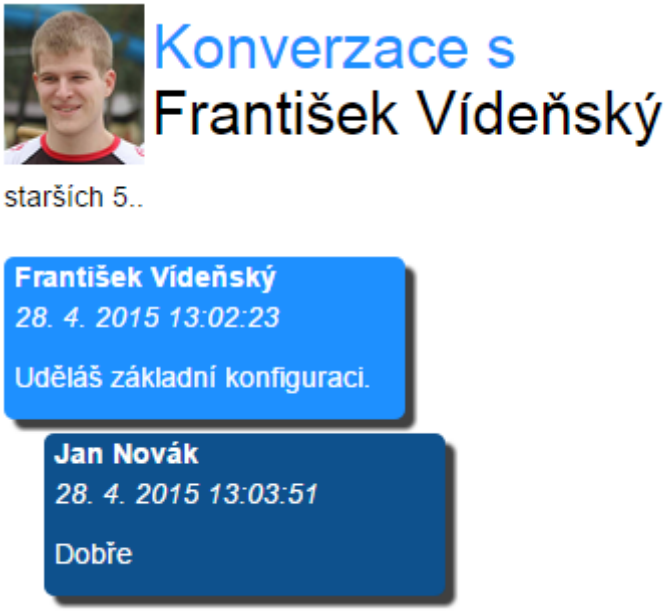
Založení nového projektu

Při založení nového projektu je nutné vyplnit povinné položky - název projektu a jeho datum zahájení. Volitelné pole je popis projektu, kde může být vyplněn krátký, ale i podrobnější popis nového projektu. Tlačítkem vytvořit se vytvoří nový projekt s vyplněnými údaji a uživatel, jenž ho vytvořil, se automaticky stává jeho manažerem a je přesměrován na hlavní stránku projektu. Druhá možnost založení projektu je načtení importovaného projektu z MS Project 2013 nebo ProjectLibre.

Moje zprávy

Do sekce moje zprávy se uživatel dostane po kliknutí na ikonu zprávy, která se nachází v pravém horním rohu obrazovky. Číslo na ikonce zobrazuje počet nepřečtených zpráv. Na stránce se nachází seznam zpráv seřazených do konverzací, u každé je informace, kdy přišla poslední zpráva, s kým je konverzace vedena, a část jejího obsahu. Zprávu je možné otevřít, čímž se nám zobrazí její obsah a zpráva se označí jako přečtená. Je také možné posílat nové zprávy. Uživatel si může vybrat příjemce zprávy ze seznamu uživatelů, kteří aktuálně pracují na projektu.

Zprávy se dělí do dvou kategorií. První, klasické zprávy, jsou popsány výše. Druhá jsou systémové zprávy. Na systémové zprávy nelze odpovídat a slouží k informování uživatele, že byl pozván do projektu, kde je možnost pozvání přijmout nebo odmítnout. Nebo je pomoci nich uživatel informován, že byl přiřazen k aktivitě a tak dále.



Obrázek 6.2: Ukázka konverzace

Možnosti profilu

V pravém horním rohu, vedle ikony se zprávami, se nachází jméno uživatele s profilovou fotografií. Pokud nebyla profilová fotografie nahrána, bude zobrazena výchozí fotografie. Po kliknutí na jméno nebo fotografii je zobrazeno vysouvací menu s možnostmi upravit profil, změna projektu a odhlásit se. V úpravě profilu je možné upravit všechny položky, které uživatel vyplnil při registraci. Při výběru možnosti změna projektu je uživatel přesměrován na stránku výběr projektu. Pokud uživatel vybere možnost odhlásit se, bude odhlášen ze systému a přesměrován zpět na úvodní stránku.

6.1.2 Stránky pro pracovníky

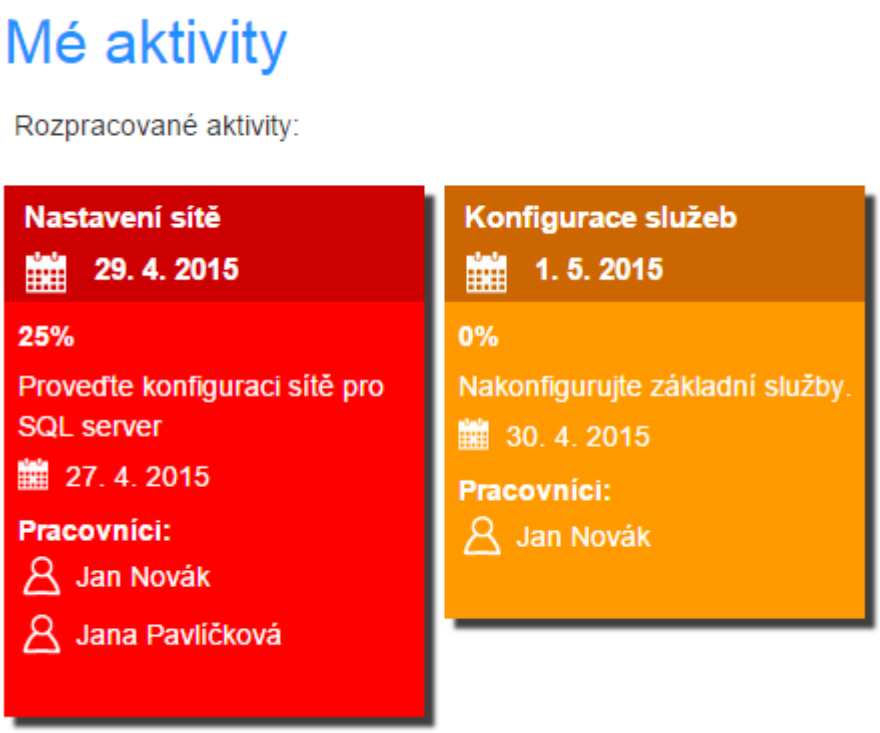
Stránky pro pracovníky jsou zaměřené na týmovou spolupráci. Zajišťují komunikace mezi členy týmu a informují pracovníka o jeho aktuálních úkolech.

Hlavní stránka projektu

Po výběru projektu je pracovník přesměrován na hlavní stránku. Na stránce nalezne základní informace o projektu a vypsané aktuality od projektového manažera. Dále se mu zobrazí urgentní aktivity, které musí být z aktivit, ke kterým je pracovník přiřazen, co nejdříve dokončeny.

Přehled aktivit

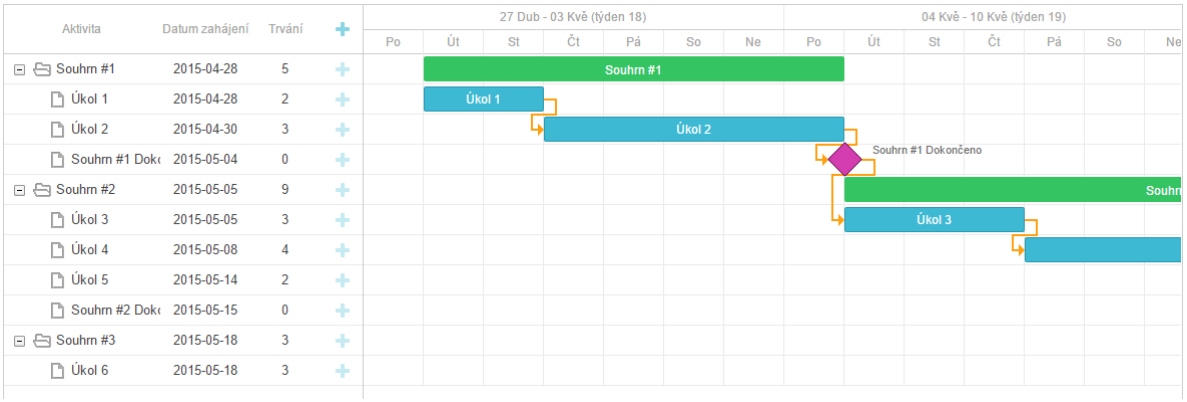
V přehledu aktivit se pracovníkovi zobrazí všechny aktivity, na kterých pracuje, seřazeny podle data dokončení. Aktivity jsou zobrazeny v podobě lístečku a obsahují název aktivity, její popis, procenta, kolik je odpracováno. Na lístečku se také nachází seznam lidí, kteří na aktivitě aktuálně pracují. Lístečky jsou barevně rozlišeny podle času, kolik zbývá do jejich ukončení. Pokud zbývají tři nebo dva dny k dokončení, je lísteček žlutý, pokud zbývá už jeden jediný den, je červený.



Obrázek 6.3: Ukázka přehledu aktivit

Ganttův diagram

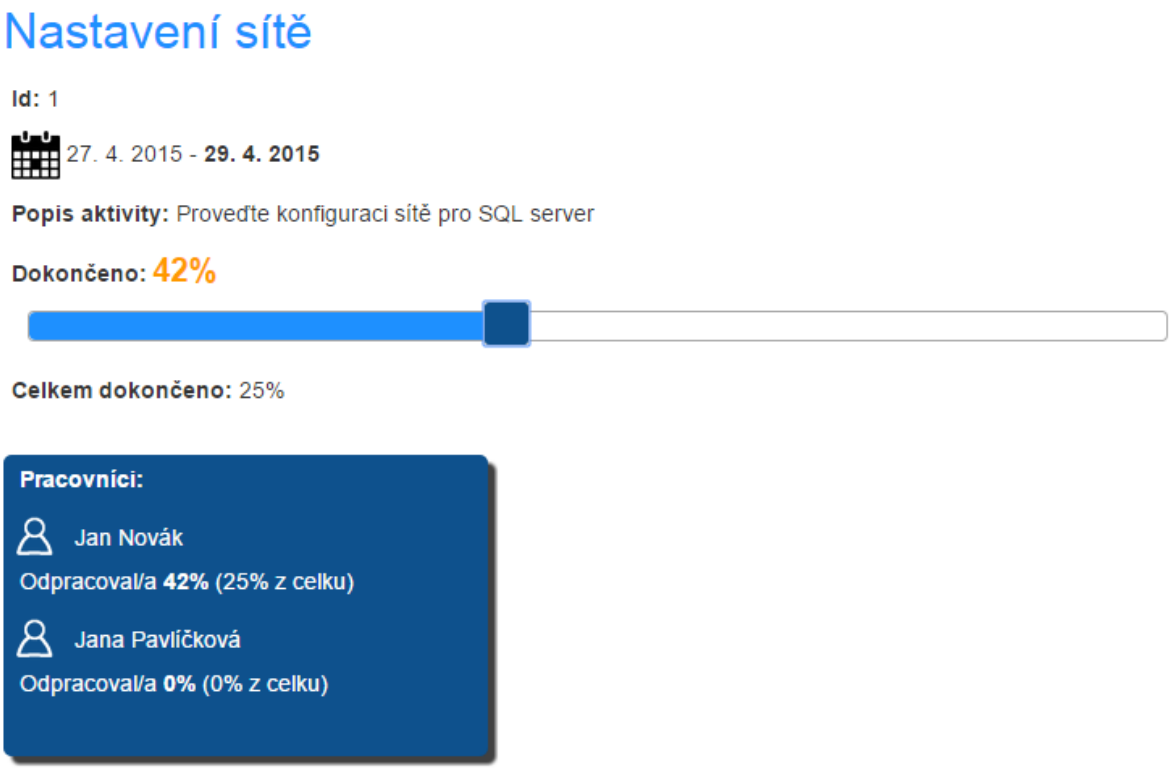
Přehled všech aktivit v projektu si může pracovník zobrazit pomocí Ganttova diagramu. Aktivity zde si může jen prohlížet a otevřít si jejich detail. Není možné zde provádět jakékoliv úpravy.



Obrázek 6.4: Ukázka Ganttova diagramu

Detail aktivity

V detailu aktivity je znovu zobrazen její název, popis a stav. Pomocí posuvníku může uživatel měnit, na kolik procent odpracoval svůj díl na aktivitě. Dále jsou u každé aktivity k dispozici komentáře, pomocí kterých mohou pracovníci nechat vzkazy pro další členy týmu. K dispozici je také nahrávání nejrůznějších souborů, aby je uživatelé nemuseli nahrávat na externí uložiště a vždy je měl i zbytek projektového týmu u dané aktivity k dispozici.



Obrázek 6.5: Ukázka práce na aktivitě

Výkazy

V sekci výkazy vidí pracovník své výkazy seřazené v tabulce podle jednotlivých aktivit. Při vytváření nového výkazu je nutné vyplnit popis práce, kterou pracovník na aktivitě vykonal, a zadat dobu, jak dlouho mu práce trvala.

6.1.3 Stránky pro projektového manažera

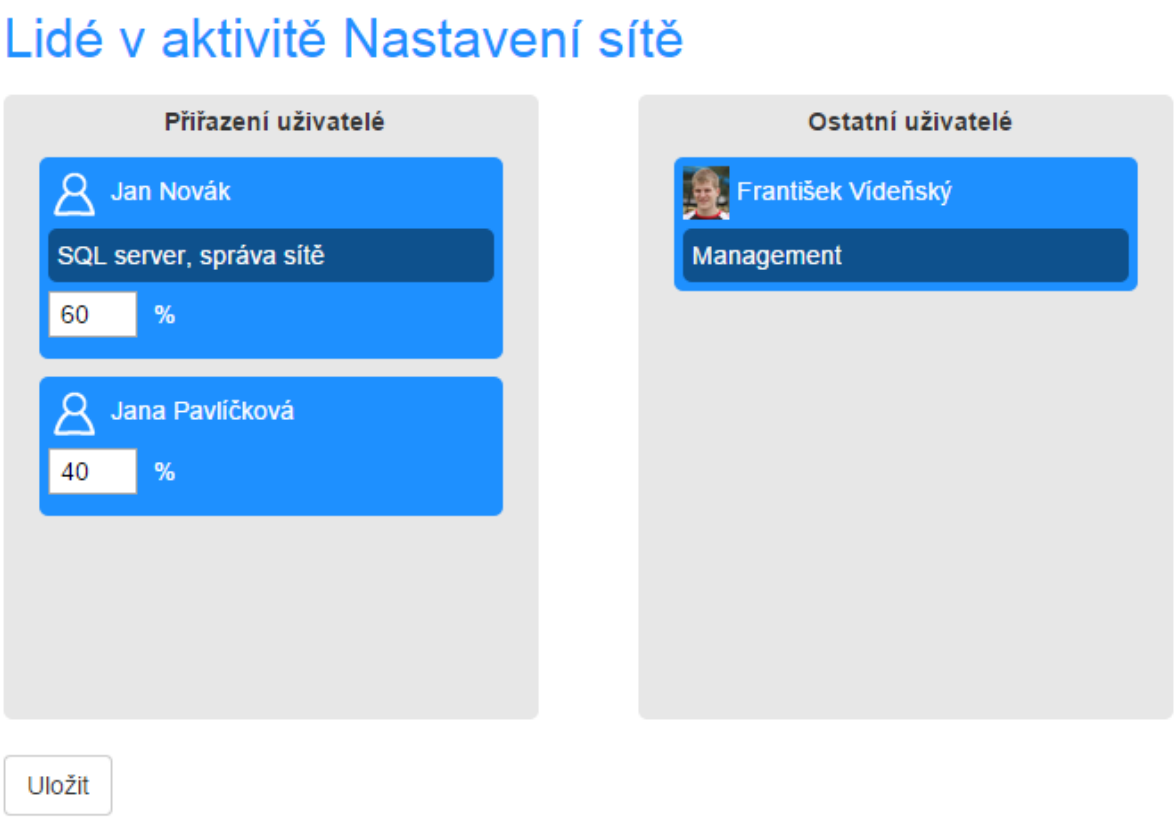
Projektový manažer má velký výběr stránek. Musí mít možnost upravit informace o projektu, plánování aktivit, přiřazení jednotlivých lidí a tak dále.

Úprava projektu

V sekci úprava projektu má projektový manažer možnost upravit informace, které zadal při jeho vytvoření. Tedy jeho název, datum zahájení, nebo může upravit či doplnit jeho popisek.

Přidání lidí do projektu

Na stránce přidání lidí do projektu je k dispozici vyhledávač, který vyhledává lidi v systému podle dvou atributů, podle jména uživatele, nebo podle jeho emailu. Pokud projektový manažer nalezne hledanou osobu, přidá ji k projektu pomocí tlačítka pozvat. Počet osob v projektu není nijak omezen, projektový manažer proto může přidávat lidi, dokud si nesestaví celý projektový tým. Pozvanému uživateli je zaslána zpráva s pozvánkou.



Obrázek 6.6: Ukázka přiřazení lidí k aktivitě

Aktivity

Aktivity jsou projektovému manažerovi zobrazeny v Ganttově diagramu. S jeho pomocí může aktivity posouvat v časovém horizontu a vytvářet mezi nimi vztahy. Tedy po které aktivitě následuje jaká aktivita. Může také založit novou aktivitu pomocí ikonky plus. Pokud klikne na ikonku plus u aktivity, také se přesměruje na stránku pro vytvoření aktivity, ale jako předchůdce se vyplní daná aktivita. Při vytváření nové aktivity je požadováno zadat její název, popis, datum zahájení práce, nebo nechat zaškrtnutou možnost automatického plánování a dobu trvání. Při zadání doby trvání má projektový manažer dvě možnosti. První je, že ví, jak dlouho bude dokončení aktivity trvat a dobu zadá, druhou je přijmout nabídku využití metody PERT, kde zadá optimistický, nejpravděpodobnější, a pesimistický odhad doby trvání a hodnota se automaticky dopočítá podle vzorce 2.1. K aktivitě je možné přidat její předchůdce, pokud je vybráno automatické plánování, vypočítá se datum zahájení.

Po vytvoření aktivity je možné přiřadit k ní pracovníky, jsou nabídnutí jen ti lidé, kteří byli přiřazeni k projektu. Každému uživateli musíme přiřadit určitý díl práce na aktivitě. Pokud pracovník má ve svém profilu vyplněné dovednosti, jsou zde projektovému manažerovi k dispozici, aby mu pomohly při rozhodování, koho k aktivitě přiřadí. Opět je pracovníkovi zaslána zpráva, že byl k dané aktivitě přiřazen a jakým dílem se na práci bude podílet. Aktivity lze mazat nebo upravit. I když manažer na aktivitě nepracuje, může k ní přidávat komentáře a nahrávat soubory.

Projektový manažer také může aktivity do projektu nahrát hromadně pomocí možnosti nahrát aktivity. Zde má k dispozici šablonu, do níž vyplní potřebné údaje, a aktivity se do projektu přiřadí.

Síťový diagram

V síťovém diagramu jsou zobrazeny aktivity, milníky a fáze. Je zde také vyznačena kritická cesta červenou barvou. Dvojím kliknutím na jakýkoliv element si může projektový manažer zobrazit jeho

informace jako nejdříve možné zahájení, ukončení, nejpozději přípustné zahájení, ukončení a dále celkovou a volnou rezervu.

Aktuality

Pokud projektový manažer potřebuje něco sdělit celému projektovému týmu, může využít systém aktualit. Při vytváření nové aktuality je požadováno napsat její název a text. V přehledu aktualit jsou zobrazeny oba atributy a ještě datum, kdy byla aktualita napsána. Každou aktualitu je možné upravit nebo vymazat.

Výkazy

Sekce s výkazy slouží k tomu, aby měl projektový manažer přehled, kdo a kolik práce vykonal na jaké aktivitě. Výkazy si lze zobrazit podle aktivit, kdy u každého pracovníka jsou vypsané výkazy. Nebo si můžeme zobrazit výkazy od jednotlivých uživatelů, kde jsou v tabulce zobrazeny aktivity, na nichž pracoval, a jednotlivé výkazy k nim.

7 Ověření funkčnosti

Ověření funkčnosti jsem prováděl pro odhalení chyb v aplikaci. Proto jsem si vytvořil sadu úkolů a snažil se odhalit různé krajní případy, ke kterým by mohlo při užívání dojít. Důkladně jsem se zaměřil na hlídání přístupových práv, tak aby se pracovník nedostal k možnostem manažera nebo nemohl upravovat úkol, ke kterému není přiřazen. Ověřoval jsem také, aby se uživatel nemohl dostat do projektu, na němž nepracuje.

Dále jsem musel vyzkoušet, zda program správně počítá všechny náležitosti a správně nalezne kritickou cestu. K tomu jsem využil možnosti importu projektu z MS Project. Projekt se správně načetl a kontroloval jsem vypočítané hodnoty jako nejdříve možný začátek/konec, nejpozději přípustný začátek/konec, celkovou a volnou rezervu. Všechny hodnoty u všech aktivit se vypočítaly stejně jako v MS Projectu. Dále jsem dělal na projektu v obou programech stejné změny, jako posunutí jedné aktivity, a testoval, jestli se správně přepočítají aktivity, které navazují, zda se přepočítá délka fáze a také zda se přepočítají dřív zkoumané hodnoty. Ve všech těchto úkolech jsem dosáhl stejných výsledků jako v programu MS Project a protože se jedná o známý, profesionální, software pro podporu plánování a řízení projektů, mám jistotou, že jsou výsledky správné.

Ověření od uživatelů

S mou aplikací bylo seznámeno 5 lidí z okruhu přátel a rodiny ve věkovém rozmezí 19 a 49 let. Již od začátku jsem dostal několik doporučení na změnu, jako úprava některých odstínů barev, změna písma.

Po dokončení implementace jsem vytvořil sadu úkolů, která měla prověřit jak ještě jednou základní funkcionalitu, tak hlavně jak se uživatelé v programu orientují a zda se jim dobře pracuje. Ověření od uživatelů dopadlo dobře, úkoly splnili a nebyly zaznamenány žádné problémy v orientaci.

8 Možnosti dalšího rozšíření

Výsledná aplikace je sice dokončená, ale vždy se najdou další návrhy, co by se dalo vylepšit nebo co by se mohlo přidat úplně nového. V této kapitole jsem popsal několik zajímavých nápadů, které by aplikaci zdokonalily. Stěžejní body možného rozšíření jsou:

- Možnost upravit projekt v síťovém diagramu.
- Rozdělit projektový tým do více skupin.
- Vnořovat fáze a vytvářet mezi nimi vztahy.
- Export projektu.
- Vytvoření chatovacích místností.
- Práce s materiálními zdroji.

Projektový manažer může nyní měnit čas zahájení aktivit, jejich délku a vztahy přímo v Ganttově diagramu. Síťový diagram však nyní slouží jen k zobrazení stavu projektu, ale není možné provádět jakékoliv úpravy. Pokud si při prohlížení síťového diagramu projektový manažer všimne nějaké chyby, nebo ho napadne, že by chtěl něco změnit, musí se přesunout do Ganttova diagramu. Proto by bylo dobré umožnit projektovému manažerovi také upravovat aktivity i v tomto zobrazení.

Na náročnějších projektech může pracovat velmi rozsáhlý projektový tým. Bylo by tedy vhodné, aby projektový manažer mohl projektový tým rozdělit na menší skupiny a například poslat této skupině hromadou zprávu nebo napsat aktualitu jen pro ně.

MS Project umožňuje vnořovat souhrnné úkoly do sebe a vytvářet mezi nimi vztahy. Má aplikace umí dělit aktivity do fází, což však při náročnějších projektech může být svazující a méně přehledné, proto pokud bych pokračoval ve vývoji, určitě bych umožnil vnořování fází do sebe a vytváření vazeb, jako umožňuje MS Project.

Aplikace umožňuje import z programů MS Project a ProjectLibre, pokud bych pracoval na dalších rozšířeních, přidal bych také export do těch aplikací.

Pracovníci si mohou k aktivitě nahrávat soubory a přidávat komentáře. Tento systém však není stavěný pro větší diskuze. Z tohoto důvodu by nejlépe každá aplikace nebo alespoň každý projekt měl mít svou chatovací místnost, ve které by mohly probíhat dlouhé a přehledné diskuze nad problémy.

Při plánování a řízení projektu se mimo lidské zdroje využívají také zdroje materiální. To aplikace nyní neumožňuje vůbec. Bylo by tedy dobré, kdyby projektový manažer mohl počítat i s materiálními zdroji a mít k dispozici přehled, kdy je jaký zdroj využíván, a aby aplikace hlídala, aby nebyla v jednu chvíli přesažena kapacita zdroje.

Určitě by se daly vymyslet další nápady na změny, ale při dalším vývoji bych se prvně zaměřil na výše popsané.

9 Závěr

V práci jsem se věnoval návrhu aplikace pro týmovou spolupráci a správu úkolů. Představil jsem problematiku projektového řízení a nástroje, které se používají při časovém plánování projektu. Díky získaným poznatkům z představených problematik jsem vytvořil kritéria, podle nichž jsem zanalyzoval škálu funkcí, které nabízejí dostupné nástroje, dále jsem hodnotil uživatelské rozhraní a přenositelnost dat. Na základě výsledku analýzy jsem vytvořil návrh vlastní aplikace.

Výsledkem práce je aplikace, která nabízí projektovému manažerovi všechny potřebné nástroje pro plánování projektu, pracovníkům přehled svých úkolů spolu s nástroji pro jejich úpravu a komunikaci v rámci projektového týmu.

Za přínos této práce považuji, že jsem se dozvěděl, jak probíhá projektové řízení, a naučil jsem se technikám projektového plánování. Dále jsem se naučil tvořit rozsáhlejší webové aplikace a získal zkušenosti s jazykem C#.

Aplikace může být rozšířena o spoustu dalších možností, které můžeme například najít u komerčních aplikací. Některé z možností jsem popsal v předchozí kapitole.

Výsledná aplikace vyhovuje požadavkům ze zadání a splňuje všechna stanovená kritéria. Lze ji využít při práci na projektech ve všech odvětvích.

Literatura

[1] ROSENAU, Milton D. *Řízení projektů: příprava a plánování, zahájení, výběr lidí a jejich řízení, kontrola a změny, vyhodnocení a ukončení*. Vyd. 1. Praha: Computer Press, 2000, xiv, 344 s. ISBN 80-7226-218-1.

[2] ŘEHÁČEK, Petr. *Projektové řízení podle PMI*. 1. vyd. Praha: Ekopress, 2013, 123 s. ISBN 978-80-86929-90-3.

[3] SCHWALBE, Kathy. *Řízení projektů v IT*. Vyd. 1. Překlad David Krásenský. Brno: Computer Press, 2007, 720 s. ISBN 978-80-251-1526-8.

[4] Pojmy v projektovém řízení. *Icontio* [online]. 2010 [cit. 2015-01-21]. Dostupné z: http://www.smartgov.cz/projektove-rizeni/pojmy-v-projektovem-rizeni.aspx

[5] DOLEŽAL, Jan, Pavel MÁCHAL a Branislav LACKO. *Projektový management podle IPMA*. 2., aktualiz. a dopl. vyd. Praha: Grada, 2012, 526 s. Expert (Grada). ISBN 978-80-247-4275-5.

[6] WBS - (Work Breakdown Structure). *ManagementMania.com* [online]. © 2011-2013 [cit. 2015-03-01]. Dostupné z: https://managementmania.com/cs/work-breakdown-structure

[7] Manažer (Manager). *Management* Mania [online]. © 2011-2013 [cit. 2015-01-21]. Dostupné z: https://managementmania.com/cs/manazer

[8] FOTR, Jiří a Lenka ŠVECOVÁ. *Manažerské rozhodování: postupy, metody a nástroje*. 2., přeprac. vyd. Praha: Ekopress, 2010, 474 s. ISBN 978-80-86929-59-0.

[9] Tým, skupina, komunita. *Vedeme* [online]. © 2005 - 2015 [cit. 2015-01-21]. Dostupné z: http://www.vedeme.cz/pro-vedeni/kapitoly-vedeni/66-tymy/150-tym-skupina-komunita.html

[10] GRASSEOVÁ, Monika, Radek DUBEC a Roman HORÁK. *Procesní řízení ve veřejném sektoru: teoretická východiska a praktické příklady*. Vyd. 1. Brno: Computer Press, 2008, v, 266 s. ISBN 978-80-251-1987-7.

[11] KORECKÝ, Michal a Václav TRKOVSKÝ. *Management rizik projektů: se zaměřením na projekty v průmyslových podnicích*. 1. vyd. Praha: Grada, 2011, 583 s. Expert (Grada). ISBN 978-80-247-3221-3.

[12] Metody plánování projektů. *Promis* [online]. 2004 [cit. 2015-01-23]. Dostupné z: http://promis.econ.muni.cz/lecture/2/2/3/

[13] *ProjectLibre* [online]. ©2014 [cit. 2015-02-28]. Dostupné z: http://www.projectlibre.org/

[14] *Trello* [online]. ©2015 [cit. 2015-02-28]. Dostupné z: https://trello.com/

[15] Software pro řízení projektů| Microsoft project. *Microsoft Office* [online]. ©2015 [cit. 2015-02-28]. Dostupné z: https://products.office.com/cs-cz/project/project-and-portfolio-management-software

[16] Primavera P6 Enterprise Project Portfolio Management Release 8.4. *Oracle* [online]. [2015] [cit. 2015-02-28]. Dostupné z: http://www.oracle.com/us/products/applications/primavera/p6-enterprise-project-portfolio-management/product-update/overview/index.html

[17] *Software pro řízení projektů - Easy Project* [online]. [2015] [cit. 2015-02-28]. Dostupné z: http://www.easyproject.cz/

[18] ČÁPKA, David. 6. díl - Scaffolding a Entity Framework v ASP.NET MVC. In: *Itnetwork.cz* [online]. @2015 [cit. 2015-03-28]. Dostupné z:

http://www.itnetwork.cz/asp-dot-net-mvc-tutorial-scaffolding-entity-framework-editor-clanku

[19] MACDONALD, Matthew, Adam FREEMAN a Mario SZPUSZTA. *ASP.NET 4 a C# 2010: tvorba dynamických stránek profesionálně*. Vyd. 1. Překlad Jan Pokorný. Brno: Zoner Press, 2011, 880 s. Encyklopedie Zoner Press. ISBN 978-80-7413-131-8.

[20] ČÁPKA, David. 1. díl - Úvod do MVC architektury v ASP.NET. In: *Itnetwork.cz* [online]. © 2015 [cit. 2015-03-21]. Dostupné z: http://www.itnetwork.cz/asp-dot-net-uvod-do-mvc-architektury

[21] The Repository Pattern. *MSDN - Microsoft Developer Network* [online]. @2015 [cit. 2015-04-01]. Dostupné z: https://msdn.microsoft.com/cs-cz/library/ff649690.aspx

[22] Interfaces (C# Programming Guide). *MSDN - Microsoft Developer Network* [online]. @2015 [cit. 2015-04-01]. Dostupné z: https://msdn.microsoft.com/en-us/library/ms173156.aspx

[23] FOWLER, Martin. Inversion of Control Containers and the Dependency Injection pattern. In: *Martin Fowler* [online]. 23. 1. 2004 [cit. 2015-04-01]. Dostupné z: http://www.martinfowler.com/articles/injection.html

[25] MARGORÍN, Marián. *JQuery bez předchozích znalostí*. Vyd. 1. Brno: Computer Press, 2011, 253 s. ISBN 978-80-251-3379-8.

[26] Editable JavaScript Gantt Chart - dhtmlxGantt. *JavaScript UI Library, AJAX components & HTML5 Framework - DHTMLX* [online]. @2015 [cit. 2015-04-29]. Dostupné z: http://dhtmlx.com/docs/products/dhtmlxGantt/

[27] Gantt Chart for ASP.NET MVC with dhtmlxGantt. In: *JavaScript UI Library, AJAX components & HTML5 Framework - DHTMLX* [online]. 14. 5. 2014 [cit. 2015-04-03]. Dostupné z: http://www.dhtmlx.com/blog/using-dhtmlx-gantt-chart-asp-net-mvc/

[28] *JointJS - the HTML 5 JavaScript diagramming libraly* [online]. © 2009-2015 [cit. 2015-04-29]. Dostupné z: http://jointjs.com/

Seznam příloh

Příloha A – obsah CD

Příloha A

Obsah CD

- /src/ - složka obsahuje zdrojové soubory
- /doc/- tato práce ve formátu .pdf a .docx
- /doc/manual/ - manuál k aplikaci ve formátu .pdf a .docx