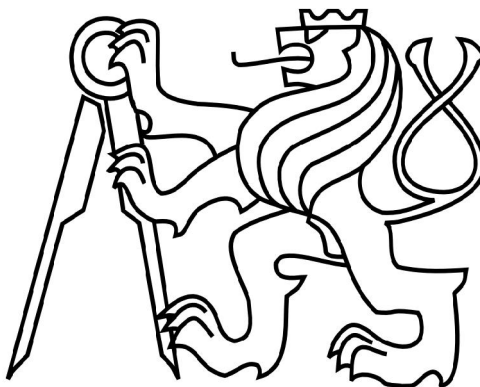


České vysoké učení technické v Praze
Fakulta elektrotechnická
Katedra počítačů



Semestrální práce

Poznámkový blok o knihách

Koldcsiter T., Hovorka V., Kovář J., Egert P., Lom M., Kulovaný J., Štětovská M.

Studijní program: Elektrotechnika a informatika, prezenční

Obor: Výpočetní technika

25. února 2011

Obsah

1.	Analýza	5
1.1	Project overview statement (POS)	5
1.1.1	Problém.....	5
1.1.2	Cíle projektu.....	5
1.1.3	Zadání/Příležitost	5
1.1.4	Výhody / prospěch projektu.....	5
1.1.5	Podmínky projektu	5
1.1.6	Obsah projektu	6
1.1.7	Kritéria úspěchu	6
1.1.8	Rizika.....	6
1.1.9	Přílohy	7
1.2	Kontextový model.....	7
2.	Rozpočet.....	8
2.1	Odhad časové náročnosti projektu	8
2.1.1	Odhad časové složitosti implementace.....	8
2.1.2	Grafické znázornění odhadů	8
2.2	Odhad rozpočtu celého projektu na začátku projektu	11
2.3	Skutečný rozpočet celého projektu	13
3.	Plán práce	15
3.1	Work breakdown structure (WBS).....	15
3.2	Přehled plnění úkolů.....	15
3.3	Ganttův model.....	17
3.4	Matice zodpovědnosti.....	18
3.5	Určení kritické cesty.....	19
4.	Návrh.....	20
5.	Infrastruktura	21
6.	Zhodnocení projektu	22
7.	Kód aplikace.....	25
8.	Testy.....	26
9.	Seznam příloh.....	28

Seznam obrázků

Obrázek 1 - Kontextový model projektu	7
Obrázek 2 - Odhad celkové časové složitosti implementace 1.kolo	8
Obrázek 3 - Odhad celkové časové složitosti implementace 2.kolo	9
Obrázek 4 - Odhad celkové časové složitosti implementace 3.kolo	9
Obrázek 5 - WBS znázorňuje dekompozici projektu na dílčí aktivity a úkoly.....	15
Obrázek 6 - Ganttův model.....	17
Obrázek 7 - Matice zodpovědnosti k 22.4.2011	18
Obrázek 8 - Určení kritické cesty projektu.....	19
Obrázek 9 - USE CASE.....	28
Obrázek 10 - USE CASE 2.....	30
Obrázek 11 - Requirements model – funkční i nefunkční požadavky	30
Obrázek 12 - RM funkční požadavky.....	31
Obrázek 13 - RM nefunkční požadavky.....	31
Obrázek 14 - Doménový model	32

Seznam tabulek

Tabulka 1 - Tabulka znázorňující odhady časové náročnosti jednotlivých částí projektu.....	10
Tabulka 2 - Tabulka znázorňující odhad rozpočtu celého projektu na začátku.....	12
Tabulka 3 - Tabulka znázorňující výpočet skutečného rozpočtu projektu.....	13
Tabulka 4 - Tabulka zobrazující výkaz práce za první iteraci.....	15
Tabulka 5 - Tabulka zobrazující výkaz práce za druhou iteraci.....	16
Tabulka 6 - Tabulka zobrazující výkaz práce za třetí iteraci.....	16
Tabulka 7 - Tabulka zobrazující výkaz práce za čtvrtou iteraci.....	16
Tabulka 8 - Tabulka zobrazující výkaz práce za pátou iteraci.....	16

1. Analýza

1.1 Project overview statement (POS)

1.1.1 Problém

Zadavatel hledá způsob, jakým by mohl psát poznámky o knihách a sdílet je s ostatními uživateli.

1.1.2 Cíle projektu

Snadné psaní poznámek o vložených knihách

- Sdílení poznámek s ostatními uživateli
- Vyhledávání pomocí tagů v poznámkách svých i poznámkách ostatních uživatelů
- Export poznámek do textového souboru

1.1.3 Zadání/Příležitost

Úkolem je vytvořit poznámkový blok o knihách. Systém bude umožňovat vytvářet knížky a poznámky k daným knížkám. Do poznámky bude možno přidávat citace z jiných knížek. Aplikace bude umožňovat vyhledávat v knihách a poznámkách. Uživatel bude mít možnost prohlížet si vybranou knihu včetně poznámek a citací u sebe na počítači, poté co program stáhne aktuální data z GoogleDocs. Knihu včetně poznámek bude možno sdílet s dalším uživatelem.

Pro sdílení daných dat budou možné dva režimy:

- 1) čtení, ve kterém vzdálený uživatel bude mít možnost pouze číst informace o knize a poznámkách k ní přiřazených,
- 2) výše uvedené čtení a uživatel bude moci zapisovat/měnit záznamy existující knihy.

1.1.4 Výhody / prospěch projektu

Díky projektu bude možné snadné vytváření poznámek o knížkách. Komunitě uživatelů bude umožněno sdílení poznámek mezi sebou a případná editace nebo přidání poznámky do poznámek jiného uživatele. Další výhodou projektu bude snadná orientace v těchto poznámkách a vyhledávání pomocí tagů. Dále bude možný export poznámek do některého z běžně používaných formátů nebo tisk dané poznámky popř. poznámek.

1.1.5 Podmínky projektu

Mezi podmínky pro realizaci projektu patří (dle specifikace zadavatele)

- Implementace v programovacím jazyce python
- Synchronizace s ostatními uživateli pomocí GoogleDocs

1.1.6 Obsah projektu

1) Základní funkčnost

Poznámky k daným knihám je možné formátovat za pomoci formátování tučné, kurzíva a podtržení. Do poznámky k daným knížkám lze vkládat citace jiných knížek. Tyto citace se zobrazí pod výpisem poznámek v dodatečném okně a v textu jsou reprezentovány formou [číslo_odkazu]. Za pomoci citace je možno odkázat se na jiné existující knížky. Při vložení citace na neexistující knihu, dojde k automatickému vytvoření této knihy a uživateli bude nabídnuta možnost dodatečného vyplnění informací o dané knize. Existující citace i informace o knize je možno upravovat. Synchronizace s Google Docs se provádí při každém načítání či vkládání knih a poznámek.

2) Vyhledávání

Aplikace umožňuje vyhledávání pomocí tagů v knihách a poznámkách. Výsledkem vyhledávání je seznam knih, které obsahují v informacích nebo poznámkách hledané informace.

3) Historie

Pro knihy včetně jejich poznámek je možno zobrazit data editace včetně uživatelů, kteří daný soubor editovali. Je možné ze seznamu načíst některou z předešlých verzí.

4) Export

Uživatel bude mít možnost vyexportovat právě prohlíženou knihu včetně poznámek a citací do souboru. Typ exportu bude upřesněn. Předpokládá se export do pdf.

5) Sdílení

Knihu včetně jejich poznámek je možno sdílet s dalšími uživateli programu. Pokud uživatel otevře knihu, ve které jsou odkazy na knihy, které nejsou v uživatelově seznamu, je do poznámky na místo standardní reprezentace [číslo_odkazu] vložena sekvence [jméno_knihy].

Pokud se soubor během editace změní, je tato informace oznámena uživateli a je mu nabídnuta možnost přepsat změněný soubor na změněnou verzi nebo zahození změn a načtení aktuální verze ze serveru.

1.1.7 Kritéria úspěchu

- seznámení a pochopení dané technologie (Python, Google docs)
- průběžná práce na projektu

1.1.8 Rizika

- možnost nespolupráce části týmu na projektu
- podcenění průběžné práce na projektu a následné nedodržení deadline

- nezvládnutí implementace za použití dané technologie, jelikož je to pro nás neznámé
- vzniknutí problému při implementaci, kdy nejsme schopni daný problém vyřešit

1.1.9 Přílohy

- USE CASE (viz. seznam příloh)
- Requirements model (viz. seznam příloh)
- Doménový model (viz. seznam příloh)

1.2 Kontextový model



Obrázek 1 - Kontextový model projektu

2. Rozpočet

2.1 Odhad časové náročnosti projektu

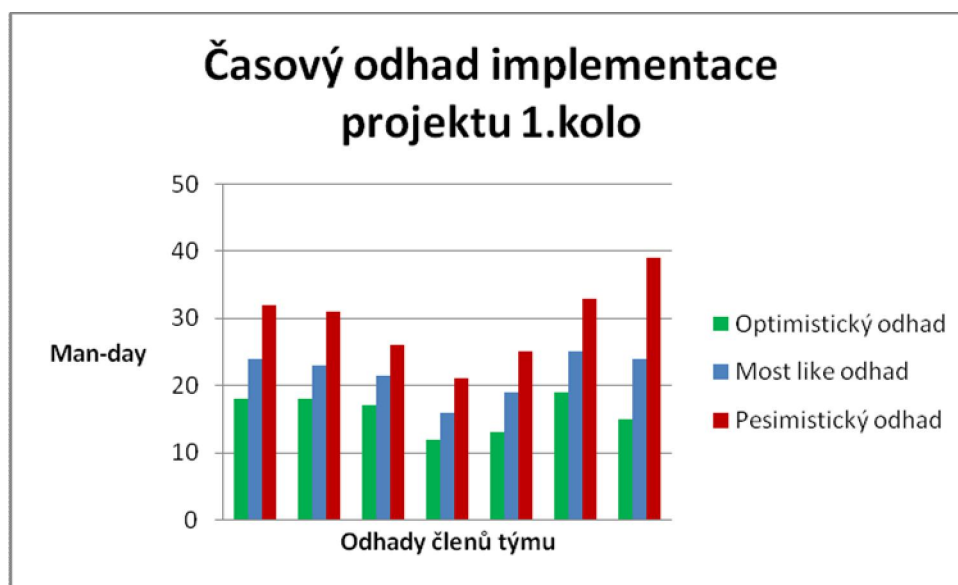
Projekt byl rozdělen na 4 základní části – analýza, návrh, administrativa a implementace. Na odhadu klíčových částí implementace se podíleli všichni členové týmu. Ostatní aktivity spojené s projektem odhadl analytik.

2.1.1 Odhad časové složitosti implementace

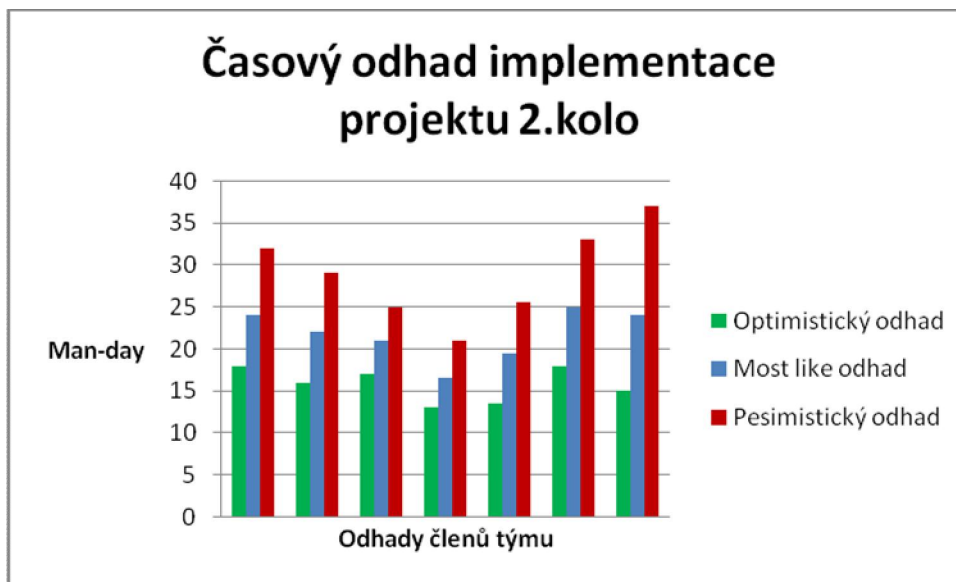
Implementace byla rozdělena do několika částí – základní funkčnost, sdílení, export, historie, vyhledávání, unit testy a GUI. Časovou složitost posledních dvou zmíněných částí odhadl analytik, odhad ostatních tzv. klíčových částí implementace probíhal metodou „Wide-Band Delphi Technique“. V každém kole (celkem 3 kola) nastínili členové týmu ke každé klíčové implementační části odhad optimistický, pesimistický a most like. Po prvním a druhém kole proběhla diskuze, proč členové týmu odhadovali předchozí časy. Konečné odhady ke každé implementační části vznikly z kola třetího pomocí aritmetického průměru a třibodové techniky. Jednotka man-day udává počet dnů, kolik každý člověk stráví na daném úkolu. Jeden man-day odpovídá 8 man-hour, kde jednotka man-hour udává, kolik čistých hodin stráví každý člen na daném úkolu. Výsledné odhady jsou znázorněny v tabulce (viz níže obrázek č.5).

2.1.2 Grafické znázornění odhadů

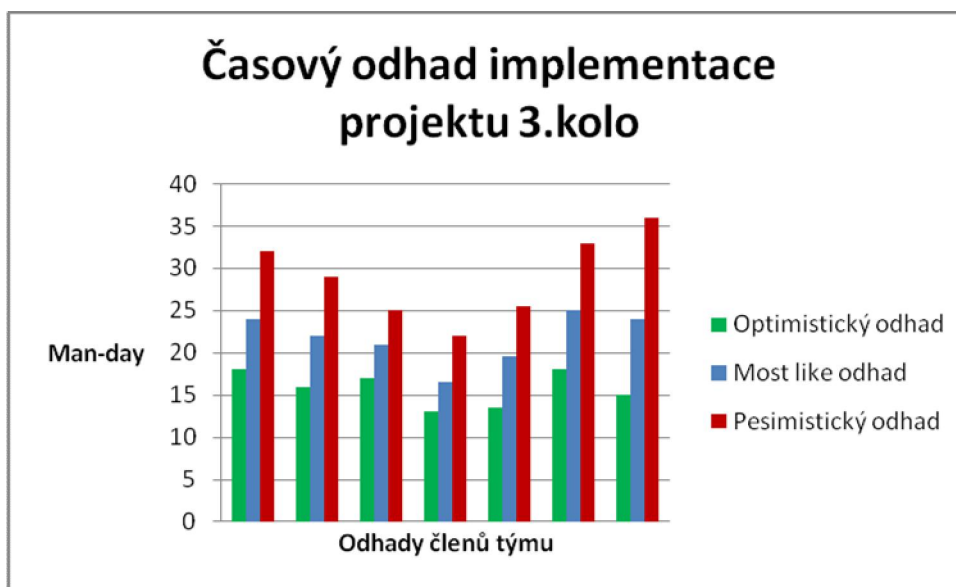
V níže uvedených grafech je znázorněn odhad časové složitosti implementace. Jedná se o součet odhadů tzv. klíčových částí implementace.



Obrázek 2 - Odhad celkové časové složitosti implementace 1.kolo



Obrázek 3 - Odhad celkové časové složitosti implementace 2.kolo



Obrázek 4 - Odhad celkové časové složitosti implementace 3.kolo

Tabulka 1 - Tabulka znázorňující odhady časové náročnosti jednotlivých částí projektu

Činnost	Časový odhad		Skutečnost	Rozdíl
	[man-day]	[man-hour]	[man-hour]	[man-hour]
Projekt celkem	41,4	331,2	372,8	41,6

Implementace	29,9	239,2	268	28,8
GUI	1,5	12	20	8
Unit testy	3,5	28	26	-2
Vyhledávání	1,9	15,2	19	3,8
Historie	2,2	17,6	23	5,4
Export	1,9	15,2	13	-2,2
Sdílení	4,9	39,2	45	5,8
Základní funkčnost	14	112	122	10
- implementace DB	4	32	33	1
- vkládání citace	3	24	28	4
- vytvoření poznámky	3	24	23	-1
- formátování poznámek	1	8	16	8
- vytvoření knížky	3	24	22	-2

Návrh	4,1	32,8	44	11,2
Synchronizace dat s DB	0,6	4,8	6	1,2
Export do souboru	0,6	4,8	5	0,2
Synchronizace sdílených dat	0,6	4,8	7	2,2
Unit testy	1	8	9	1
GUI	0,3	2,4	4	1,6
Komunikace s DB	0,4	3,2	5	1,8
Databáze	0,6	4,8	8	3,2

Administrativa	6,2	49,6	47,8	-1,8
Matice odpovědnosti	0,1	0,8	0,6	-0,2
Časový odhad projektu (3.kola)	0,6	4,8	4	-0,8
WBS	0,9	7,2	7,2	0

Dokumentace	2	16	16	0
POS	0,5	4	5	1
Odhad rozpočtu	0,7	5,6	4	-1,6
Určení kritické cesty	1	8	8	0
Ganttův model	0,4	3,2	3	-0,2

<i>Analýza</i>	1,2	9,6	13	3,4
Kontextový model	0,1	0,8	2	1,2
Use case model	0,5	4	4	0
Requierelement model	0,2	1,6	3	1,4
Doménový model	0,4	3,2	4	0,8

2.2 Odhad rozpočtu celého projektu na začátku projektu

Odhad rozpočtu byl proveden na základě Work Breakdown Structure (ohodnocení časové náročnosti jednotlivých fází projektu). Náš produkt je vytvářen v jazyce Python, který je k dispozici v Open Source licenci. Tato licence umožňuje bezplatné použití i pro komerční účely. Podrobnosti jsou přístupné na tomto odkazu: <http://docs.python.org/license.html>. Další produkt, který využíváme k tvorbě GUI aplikace, je knihovna wxWidgets. Licence nám opět umožňuje využít tuto knihovnu bezplatně (<http://www.wxwidgets.org/about/licence.htm>). Vývojové nástroje jsou tedy zdarma, což nám ušetří značné výdaje při vývoji aplikace. Totéž platí i u synchronizačního nástroje - GoogleDocs, jejichž používání je také zdarma. Hlavní části rozpočtu tedy budou náklady na analýzu a návrh aplikace a na samotnou implementaci a testování. Při tvorbě odhadu rozpočtu jsme vycházeli, jak je již výše zmíněno, z WBS, ale také z našeho odhadu časové náročnosti projektu. Hodinovou sazbu na fázi analýza/návrh jsme stanovili na 250Kč/hod a na fázi vývoj jsme stanovili 200Kč/hod. Odhad času stráveného na jednotlivých úkolech ve WBS je čistým časem (čas, kdy jsme jen pracovali na daném úkolu). Do výsledných hodin, které slouží pro výpočet rozpočtu projektu, jsme proto připočítali přibližně ještě 20% času z daných úkolů, které jsme vynásobili příslušnou hodinovou sazbou, čímž jsme získali náklady na příslušnou fázi projektu. Po sečtení nákladů všech dílčích částí jsme dostali odhadovanou finální částku, která činí **84.888 Kč**.

Tabulka 2 - Tabulka znázorňující odhad rozpočtu celého projektu na začátku

činnost	časový odhad projektu		náklady
	80% času = čistý čas [man-day]	100% času [man-hour]	
Projekt celkem	41,4	331,2	397,44

Implementace	29,9	239,2	287,04	57408
GUI	1,5	12	14,4	2880
Unit testy	3,5	28	33,6	6720
Vyhledávání	1,9	15,2	18,24	3648
Historie	2,2	17,6	21,12	4224
Export	1,9	15,2	18,24	3648
Sdílení	4,9	39,2	47,04	9408
Základní funkčnost	14	112	134,4	26880
-implementace DB	4	32	38,4	7680
- vkládání citace	3	24	28,8	5760
- vytvoření poznámky	3	24	28,8	5760
- formátování poznámek	1	8	9,6	1920
- vytvoření knížky	3	24	28,8	5760

Návrh	4,1	32,8	39,36	9840
Synchronizace dat s DB	0,6	4,8	5,76	1440
Export do souboru	0,6	4,8	5,76	1440
Synchronizace sdílených dat	0,6	4,8	5,76	1440
Unit testy	1	8	9,6	2400
GUI	0,3	2,4	2,88	720
Komunikace s DB	0,4	3,2	3,84	960
Databáze	0,6	4,8	5,76	1440

Administrativa	6,2	49,6	59,04	14760
Matice odpovědnosti	0,1	0,8	0,96	240
Dokumentace	2	16	19,2	4800
Časový odhad projektu (3. kola)	0,6	4,8	5,76	1440
WBS	0,9	7,2	8,64	2160
POS	0,5	4	4,8	1200
Odhad rozpočtu	0,7	5,6	6,72	1680
Určení kritické cesty	1	8	9,6	2400
Ganttův model	0,4	3,2	3,84	960

<i>Analýza</i>	1,2	9,6	11,52	2880
Kontextový model	0,1	0,8	0,96	240
Use case model	0,5	4	4,8	1200
Requiere ment model	0,2	1,6	1,92	480
Doménový model	0,4	3,2	3,84	960

2.3 Skutečný rozpočet celého projektu

Odhad rozpočtu byl vypočítán na 84.888, zatímco skutečný rozpočet je **95.760 Kč** po přičtení daných 20%. Rozdíl mezi skutečným a odhadovaným rozpočtem je **10.872 Kč**

Tabulka 3 - Tabulka znázorňující výpočet skutečného rozpočtu projektu

činnost	skutečnost skutečný čas [man-hour]	náklady [Kč]
<i>Projekt celkem</i>	372,8	79800

<i>Implementace</i>	268	53600
GUI	20	4000
Unit testy	26	5200
Vyhledávání	19	3800
Historie	23	4600
Export	13	2600
Sdílení	45	9000
<i>Základní funkčnost</i>	122	24400
-implementace DB	33	6600
- vkládání citace	28	5600
- vytvoření poznámky	23	4600
- formátování poznámek	16	3200
- vytvoření knížky	22	4400

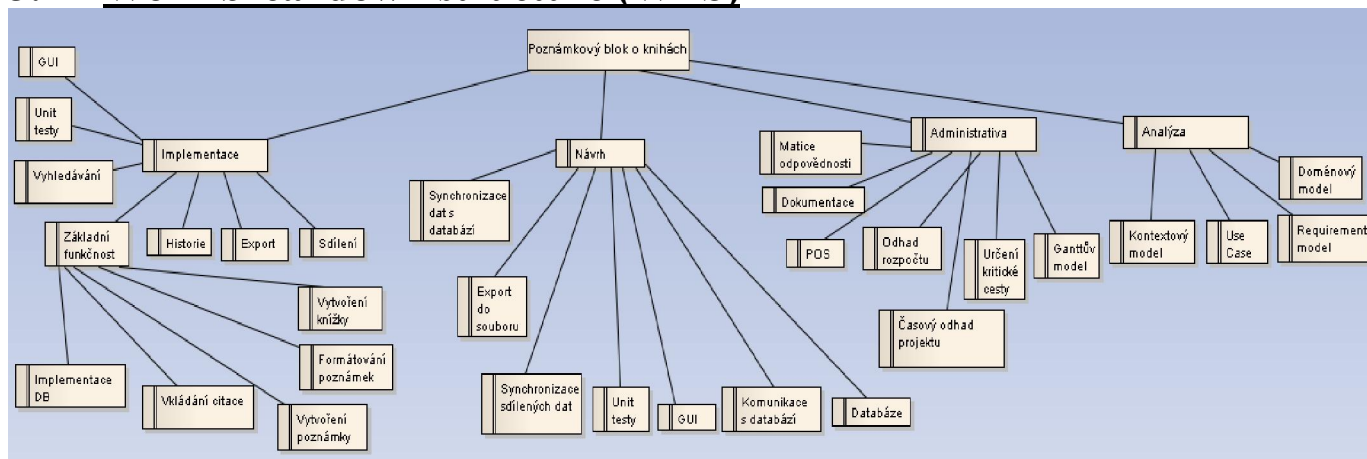
<i>Návrh</i>	44	11000
Synchronizace dat s DB	6	1500
Export do souboru	5	1250
Synchronizace sdílených dat	7	1750
Unit testy	9	2250
GUI	4	1000
Komunikace s DB	5	1250
Databáze	8	2000

<i>Administrativa</i>	47,8	11950
Matice odpovědnosti	0,6	150
Dokumentace	4	1000
Časový odhad projektu	7,2	1800
WBS	16	4000
POS	5	1250
Odhad rozpočtu	4	1000
Určení kritické cesty	8	2000
Ganttův model	3	750

<i>Analýza</i>	13	3250
Kontextový model	2	500
Use case model	4	1000
Requiereement model	3	750
Doménový model	4	1000

3. Plán práce

3.1 Work breakdown structure (WBS)



Obrázek 5 - WBS znázorňuje dekompozici projektu na dílčí aktivity a úkoly

3.2 Přehled plnění úkolů

Přehled plnění úkolů obsahuje plány a výkazy za jednotlivé iterace (jaké úkoly byly naplánovány, kolik času se s nimi mělo strávit a skutečný stav). Jako skutečné stavy jednotlivých úkolů jsme stanovili: *hotovo* (daný úkol byl naplánovaný a správně proveden – nebyla potřeba oprava), *uděláno* (buď byl daný úkol naplánovaný a proveden, avšak byla potřeba ho opravit a nebo ho byla potřeba dodělat), *doděláno* (opravení úkolu z předchozí iterace, které bylo ve stavu uděláno).

Tabulka 4 - Tabulka zobrazující výkaz práce za první iteraci

1.iterace			
naplánované úkoly	plánovaný čas [man-hour]	skutečný čas [man-hour]	stav
Matice odpovědnosti	0,8	0,6	hotovo
Časový odhad projektu (1.kolo)	1,8	1,5	uděláno
POS	4	5	hotovo
Kontextový model	0,8	2	hotovo
Requirements model	1,6	3	hotovo
WBS	7,2	5	uděláno
USE CASE	4	4	hotovo
Návrh GUI	2,4	4	hotovo
Návrh databáze	4,8	8	hotovo
Návrh komunikace s DB	3,2	5	hotovo
Komplet. dokumentace - 1	16	3,5	uděláno

Tabulka 5 - Tabulka zobrazující výkaz práce za druhou iteraci

2. iterace			
naplánované úkoly	plánovaný čas [man-hour]	skutečný čas [ma-hour]	stav
Odhad rozpočtu projektu	5,6	4	hotovo
Vytvoření knížky	24	10	uděláno
Implementace DB	32	33	hotovo
Určení kritické cesty	8	8	hotovo
WBS	7,2	7,2	doděláno
Časový odhad projektu (2.kolo)	1,5	1,5	uděláno
Doménový model	3,2	4	hotovo
GUI	12	9	uděláno
Návrh synchronizace s DB	4,8	0	uděláno
Komplet dokumentace - 2	16	8	uděláno

Tabulka 6 - Tabulka zobrazující výkaz práce za třetí iteraci

3. iterace			
naplánované úkoly	plánovaný čas [man-hour]	skutečný čas [man-hour]	stav
Vytvoření knížky	24	12	doděláno
Časový odhad projektu (3.kolo)	1,5	1	doděláno
Návrh synchronizace s DB	4,8	6	hotovo
Návrh synchronizace sdílení dat	4,8	7	hotovo
GUI	12	20	hotovo
Vytvoření poznámky	24	23	hotovo
Komplet dokumentace -3	16	12	uděláno

Tabulka 7 - Tabulka zobrazující výkaz práce za čtvrtou iteraci

4. iterace			
naplánované úkoly	plánovaný čas [man-hour]	skutečný čas [man-hour]	stav
Formátování poznámek	8	16	hotovo
Vkládání citace	24	28	hotovo
Návrh Unit testů	8	9	hotovo
Implementace sdílení	39,2	45	hotovo
Implementace vyhledávání	15,2	19	hotovo
Návrh exportu do souboru	4,8	0	uděláno

Tabulka 8 - Tabulka zobrazující výkaz práce za pátou iteraci

5. iterace			
naplánované úkoly	plánovaný čas [man-hour]	skutečný čas [man-hour]	stav
Ganttův model	3,2	3	hotovo
Návrh exportu do souboru	4,8	5	doděláno
Unit testy	28	26	hotovo
Implementace historie	17,6	23	hotovo
Komplet. dokumentace - 4	16	16	hotovo

3.3 Ganttův model



Obrázek 6 - Ganttův model

3.4 Matice zodpovědnosti

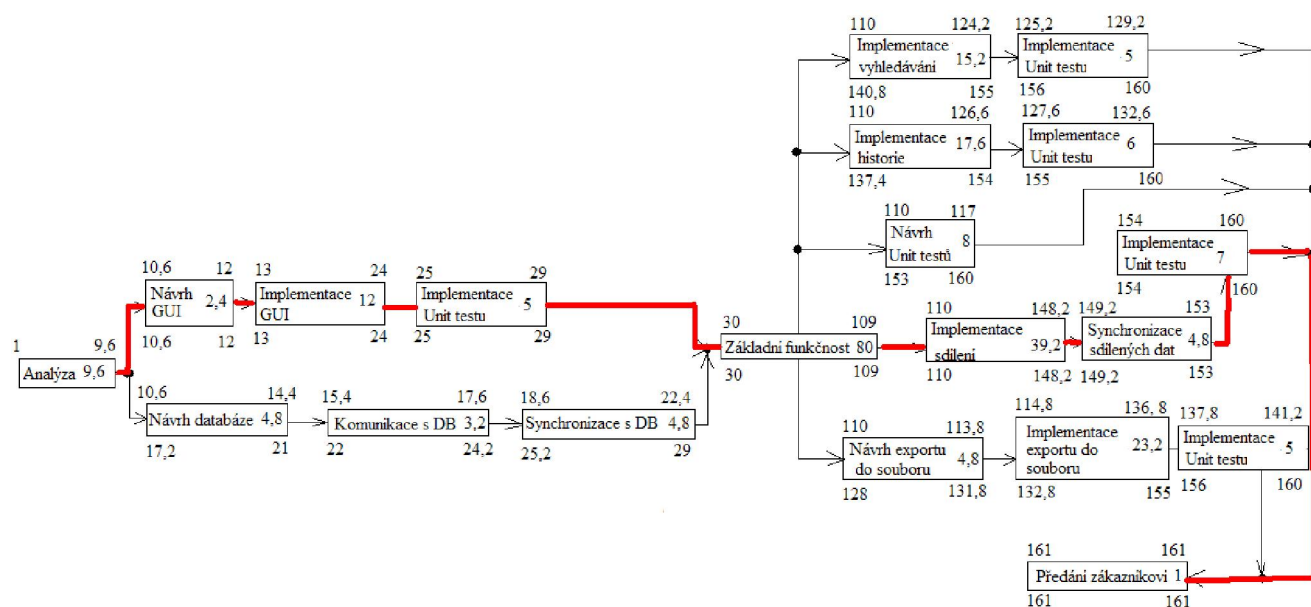
P...Pracuje, O...Odpovědný, S...Schvaluje Stav - D...Dokončeno, R...Realizován, K...Kontrolován

Úkoly	Stav	Markéta	Tomáš	Vojta	Jindra	Jirka	Michal	Petr
Project overview statement	D	P		S	P,O			
Kontextový model	D	P,O						
Use CASE model	D				S		P,O	
Návrh GUI	D							P,O
Work breakdown structure (WBS)	D	P,O						
Provedení odhadu času implementace	D	P	P	P	P,O	P	P	P
Vytvoření 1. testovací verze programu	D					P		P
Requirements model	D	P,O						
Doménový model	D			P,O				
Rozpočet projektu	D	P	P,O					
Určení kritické cesty	D	P					P,O	
Ganttův model	D		P,O				P,O	
Vyhodnocení odhadu času všech aktivit	D				P,O			
Databázová vrstva	D					P,O		
Implementace GUI	D							P,O
Unit testy	D			P	P,O			
UserGuide?	D	P,O						
Komplementace dokumentace	D	P,O	P				P	

Obrázek 7 - Matice zodpovědnosti k 26.4.2011

3.5 Určení kritické cesty

Kritická cesta nám určuje, jaké aktivity jsou kritické pro dokončení projektu. Určení kritické cesty projektu nám slouží k zaměření se na tyto stěžejní aktivity a podniknutí případných opatření k odevzdání projektu včas zákazníkovi.



Obrázek 8 - Určení kritické cesty projektu

4. Návrh

Aplikaci poznámkový blok jsme programovali v programovacím jazyku Python. Tento programovací jazyk má hlavní výhodu, že je nezávislý na platformě a nezatěžuje tolik paměť. V týmu jsme použili verzi pythonu 2.7.1.

Použili jsme knihovnu google data API, která poskytuje jednoduchý protokol pro čtení a psaní dat na web. Umožňuje uchovávat data na serveru.

Použili jsme GUI knihovnu GUI_wxWidgets.

Tato knihovna přináší následující výhody pro vývojáře:

- dobrá dokumentace a příklady užití této knihovny
- snadná implementace
- RAD designery pro Wysiwyg návrh GUI
- obsahuje pokročilé widgety již v základu
- podporuje RichText (tzn. formátování, různá písma, odkazy v textu, ...)
- podpora UNICODE

Tato knihovna také přináší následující výhody pro klienta:

- nativní vzhled aplikace (stejný vzhled jako OS)
- využívá standardní dialogy OS (výběr souboru, atd.)
- zdarma
- pokročilý dokovací systém – umožňuje uživateli upravit rozložení aplikace dle potřeby
- drag & drop

Pro export z programu jsme použili pdf knihovnu ReportLab. Tato knihovna umožňuje vytvořit dokument v PDF použitím programovacího jazyku Python. Tuto knihovnu jsme použili, jelikož se nám zdála jako jedna z nejschůdnějších cest, jak provést export do souboru.

5. Infrastruktura

Pro sdílení dokumentů a implementačních kódů jsme použili repositář SVN. Tento repositář se skládá ze složek:

- Analýza – obsahuje analytické rozborů a modely našeho projektu:
 1. určení kritické cesty
 2. odhad časové náročnosti projektu
 3. odhad rozpočtu projektu
 4. model požadavků
 5. use case model
 6. WBS
- Presentace – obsahuje prezentované dokumenty při jednotlivých iteracích
- branches
- tags
- trunk – obsahuje implementaci našeho projektu:
 1. databázová vrstva
 2. GUI
 3. model
 4. view
 5. kontrolér
 6. testy
- wiki

Pro komunikaci v týmu jsme používali wiki a mailů. Používali jsme také pro přerozdělení práce v týmu pro jednotlivé členy Issues tracker. Tento správce stavů jednotlivých úloh dává přehled o stavu jednotlivých úkolů a o jejich rozpracovanosti.

6. Zhodnocení projektu

Markéta Štětovská

role v týmu: analytik

přínos pro tým: řešení dokumentačních částí projektu

přínos projektu pro mě: rozšíření znalostí s programovacím jazykem Python, seznámení se s Google Docs, získání poznatků o GUI v Pythonu, týmová spolupráce

problémy v týmu: Žádné jsem nezaznamenala.

Jiří Kulovaný

role v týmu: programátor

přínos pro tým: řešení databáze a lokální datové mezivrstvy

přínos projektu pro mě: naučení se jazyka Python, používání vývojové podpory code.google.com, ověření, že cloudová data nejsou žádný zázrak.

problémy v týmu: Preferoval bych menší tým se silnou osobností, která by měla celý koncept "v hlavě".

Petr Egert

role v týmu: aplikační programátor

přínos pro tým: Na starosti jsem měl vytvoření uživatelského rozhraní a aplikační vrstvy programu (tzn. zajistit rozhraní pro komunikaci mezi uživatelem a databázovou vrstvou). Pro tým byly doufám přínosné mé zkušenosti s tvorbou uživatelského rozhraní, znalost programovacího jazyka Python a obecných programovacích návyků.

přínos projektu pro mě: Jako přínos práce v týmu bych naopak viděl možnost zaměřit se pouze na jednu konkrétní činnost (v mém případě programování). Celkově bych práci na projektu hodnotil jako přínosnou, zejména z důvodu prohloubení znalostí programovacího jazyka Python. Jako negativní bych označil časovou náročnost projektu, která mi přišla zbytečně velká.

problémy v týmu: Jako největší problém v týmu bych označil jeho nesehranost, která je však vzhledem k tomu, že tým existuje pouze krátce, vcelku pochopitelná.

Vojtěch Hovorka

role v týmu: analytik, tester

přínos pro tým: Hlavní přínos pro tým bylo vytvoření unit testů aplikace. Role analytika byla v mém případě více z počátku projektu, kdy se vytvářely základní dokumenty pro vznik projektu.

přínos projektu pro mě: Mezi kladné stránky týmu bych zařadil naučení se bližší spolupráce. Dále naučení se s jazykem Python. Dále naučení s SVN při práci na projektu.

problémy v týmu: Výrazné problémy v týmu jsem nezaznamenal. Většina částí projektu byla dokončena v dohodnutém termínu. Možná by na tento projekt mohla být i menší velikost týmu.

Michal Lom

role v týmu: analytik

přínos pro tým: Má část práce spočívala především na analytické části (use case, kritická cesta, Ganttův model) a poté na finální korektuře dokumentační části.

přínos projektu pro mě: Určitě jako největší plus bych hodnotil získání návyků na týmovou spolupráci (zde se kladně projevila fakt, že všichni členové týmu se znají), dále jsem ocenil získání zkušeností s Google Docs a v neposlední řadě i získání znalostí při tvorbě dokumentů pro tým.

problémy v týmu: Pokud vím, tak žádný výrazný problém v týmu nenastal a v případě potřeby jsme se vždy nějak domluvili.

Tomáš Koldcister

role v týmu: analytik

přínos pro tým: základní prostředky (SVN, GoogleCode), analytické úkoly, prezentace

přínos projektu pro mě: Pro mou osobu byl asi hlavním přínosem to, že jsem si vyzkoušel co všechno obnáší týmová spolupráce. Dále také určitě prohloubení znalostí v jazyce Python a objevení dalších možností využití GoogleDocs.

problémy v týmu: Dle mého názoru tým pracoval naprosto bez problémů.

Jindřich Kovář

role v týmu: analytik, tester

přínos pro tým: Na začátku projektu jsem působil jako analytik. V této roli jsem se podílel na zhotovení projektového dokumentu POS, na odhadu časové náročnosti projektu a jeho zpracování. V roli testera jsem se podílel na návrhu a implementaci unit testu databázové vrstvy a na zhotovení krátkého popisu navržených testů do dokumentace. Jako každý člen jsem sledoval práci svých kolegů a konzultoval s nimi svou i jejich práci.

přínos projektu pro mě: Spousta úkolů v projektu pro mě byla nová. Samotným přínosem je už jen fakt, že jsem se projektu mohl účastnit a naživo si vyzkoušet jeho realizaci. Hned na začátku projektu se ukázalo, jak je důležité jeho plánování. Jako efektivní pomocník se

ukázala webová stránka, kam se přidávaly jednotlivé úkoly a jejich stav. Bylo pohodlné přehledně vidět, kdo na jakém úkolu pracuje a v jaké je fázi. Dalším přínosem pro mě bylo testování, které jsem implementoval vůbec poprvé. Díky projektu jsem se také blíže seznámil s programovacím jazykem Python a trochu nahlédl k Google Docs.

problémy v týmu: Práce v týmu je hlavně o lidech, kteří ho tvoří. Ze zkušeností z jiných předmětů vím, že ne vždy se povede sestavit tým, jehož členové jsou spolehliví, vstřícní a komunikativní. Mě se, dle mého názoru, takový tým vytvořit podařilo. Žádné problémy v týmu jsem nezaregistroval. Na závěr bych chtěl poděkovat všem členům týmu a říci, že mi bylo potěšením v takovém týmu být.

7. Kód aplikace

8. Testy

Unit test databázové vrstvy

Test databázové vrstvy se skládá z těchto dílčích testů:

- **Vytvoření nové knihy**
Testující program se pokusí přidat novou knihu do DB. Úspěch pokusu se zjišťuje porovnáním počtu knížek před pokusem a po pokusu.
- **Update knihy**
Testující program se pokusí o změnu názvu již vytvořené knihy. Úspěch pokusu se zjišťuje porovnáním názvu před pokusem, po pokusu a návratovou hodnotou funkce *updateBook*.
- **Smazání existující knihy**
Testující program se pokusí o smazání existující knihy. Úspěch pokusu se zjišťuje kontrolou počtu knížek před pokusem, po pokusu a návratové hodnoty funkce *deleteBook*.
- **Vytvoření knihy s prázdným názvem**
Testující program se pokusí vytvořit knihu, která má prázdný název. Úspěch pokusu je zjištěn porovnáním počtu knížek před pokusem a po pokusu. Test je v pořádku, pokud se knihu přidat nepodaří.
- **Vytvoření knihy s dlouhým názvem**
Testující program se pokusí vytvořit knihu s názvem dlouhým 20tis znaků. Úspěch pokusu je zjištěn porovnáním počtu knížek před pokusem a po pokusu. Test je v pořádku, pokud se knihu přidat nepodaří.
- **Smazání neexistující knihy**
Testující program se pokusí smazat knihu, která neexistuje (respektive byla již jednou smazána). Úspěch pokusu závisí na návratové hodnotě funkce *deleteBook*. V případě hodnoty *true* – test dopadl neúspěšně, v případě *false* – úspěšně.
- **Update neexistující knihy**
Testující program se pokusí o update knihy, která neexistuje (respektive byla již jednou smazána). Úspěch pokusu závisí na návratové hodnotě funkce *updateBook*. V případě hodnoty *true* – test dopadl neúspěšně, v případě *false* – úspěšně.
- **Vytvoření nové poznámky**
Testující program se pokusí přidat novou poznámku k existující knize. Úspěch pokusu je zjištěn porovnáním počtu poznámek před pokusem, po pokusu a návratové hodnoty funkce *createNote*.
- **Vytvoření více poznámek v krátkém intervalu**
Testující program se pokusí přidat několik nových poznámek k existující knize v krátkém intervalu. Úspěch pokusu je zjištěn porovnáním počtu poznámek před pokusem, po pokusu a návratové hodnoty funkce *createNote*.
- **Smazání existující poznámky**
Testující program se pokusí o smazání existující poznámky. Úspěch pokusu se zjišťuje kontrolou počtu poznámek před pokusem, po pokusu a návratovou hodnotou funkce *deleteNote*.
- **Smazání více poznámek v krátkém intervalu**
Testující program se pokusí smazat několik existujících poznámek u existující knize v krátkém intervalu. Úspěch pokusu je zjištěn porovnáním počtu poznámek před pokusem, po pokusu a návratové hodnoty funkce *deleteNote*.
- **Update poznámky**
Testující program se pokusí o změnu již uložené poznámky (pouze header poznámky). Úspěch pokusu se zjišťuje porovnáním headru poznámky před pokusem, po pokusu a návratovou hodnotou funkce *updateNote*.

- Vytvoření poznámky s prázdným názvem

Testující program se pokusí vytvořit poznámku, která má prázdný název. Úspěch pokusu je zjištěn porovnáním počtu poznámek před pokusem, po pokusu a návratové hodnoty funkce *createNote*. Test je v pořádku, pokud se poznámku přidat nepodaří.

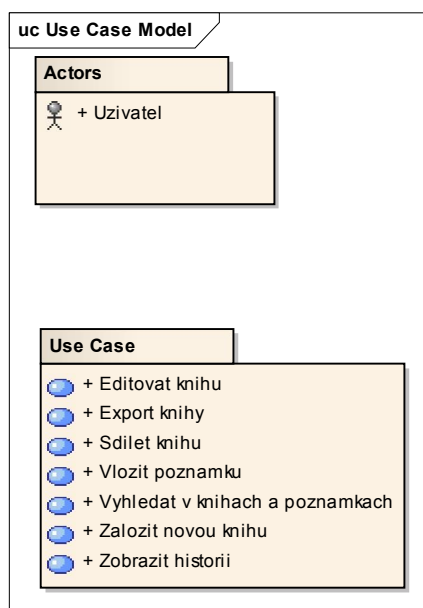
- Vytvoření poznámky s dlouhým názvem

Testující program se pokusí vytvořit poznámku s headrem dlouhým 20tis znaků. Úspěch pokusu je zjištěn porovnáním počtu poznámek před pokusem, po pokusu a návratové hodnoty funkce *createNote*. Test je v pořádku, pokud se poznámku přidat nepodaří.

Všechny testy zjistí úspěch nebo neúspěch daného pokusu, nebo zda byla vyhozena výjimka.

9. Seznam příloh

Příloha 1. USE CASE model



Obrázek 9 - USE CASE

- hlavní scénář k EDITOVAT KNIHU:

Scénář začne, když se uživatel rozhodne změnit údaje u existující knihy:

1. Uživatel vybere, co přesně chce změnit [název knihy, žánr, autor, počet stran, rok vydání]
2. Systém uloží danou změnu.
3. UC konci

- hlavní scénář k EXPORT KNIHY:

Scénář začne ve chvíli, když se uživatel rozhodne vyexportovat knihu:

1. Uživatel si vybere, jakou knihu chce vyexportovat (export probíhá včetně poznámek a citací)
2. Exportovaná kniha se uloží ve formátu .pdf
3. UC konci

- hlavní scénář k SDILET KNIHU:

Scénář začne ve chvíli, když se uživatel rozhodne sdílet knihu s jiným uživatelem:

1. Uživatel si vybere, jakou knihu chce sdílet s jiným uživatelem.
2. Uživatel vybere, v jakém módu chce knihu sdílet s jiným uživatelem (čtení nebo čtení/zápis)
3. IF čtení THEN systém nasdílí knihu vzdálenému uživateli v módu čtení
3. IF čtení/zápis THEN systém nasdílí knihu vzdálenému uživateli v módu čtení/zápis
4. UC konci

- hlavní scénář k VLOZIT POZNAMKU:

Scénář začne, když se uživatel rozhodne vložit poznámku ke knize:

1. Uživatel zadá, jakou poznámku chce vložit, popř. může vložit i citace z jiných knih
2. Systém uloží poznámku
3. UC konci

- hlavní scénář k VYHLEDAT V KNIHACH A POZNAMKACH:

Scénář začne, když se uživatel rozhodne vyhledat něco v knihách a poznámkách:

1. Uživatel zadá klíčové slovo nebo část textu, které chce vyhledat
2. Systém zobrazí daná výsledky jako seznam knih, u kterých se v jejich informacích nebo poznámkách vyskytuje hledaná informace
3. UC konci

- hlavní scénář k ZALOZIT NOVOU KNIHU:

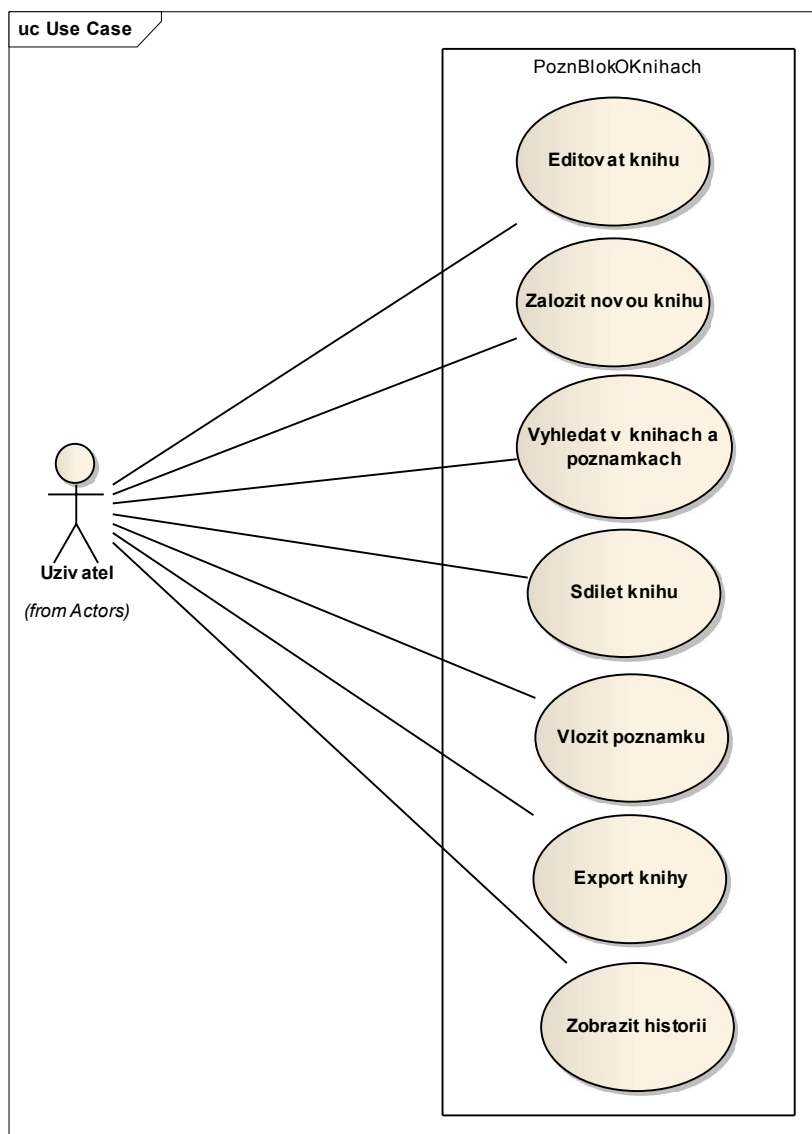
Scénář začne, když se uživatel rozhodne založit novou knihu:

1. Uživatel zadá název knihy, který slouží zároveň i jako ID, dále zadá bližší informace o knize [autor knihy, rok vydání, žánr, počet stran].
2. Systém uloží danou knihu.
3. UC konci

- hlavní scénář k ZOBRAZIT HISTORII:

Scénář začne, když se uživatel rozhodne zobrazit historii dané knihy:

- Uživatel vybere knihu, u které si chce zobrazit historii
- Systém najde danou knihu a zobrazí seznam, kdy byla kniha editována a kým
- IF uživatel chce načíst starší verzi knihy, THEN systém načte starší verzi knihy
- UC konci

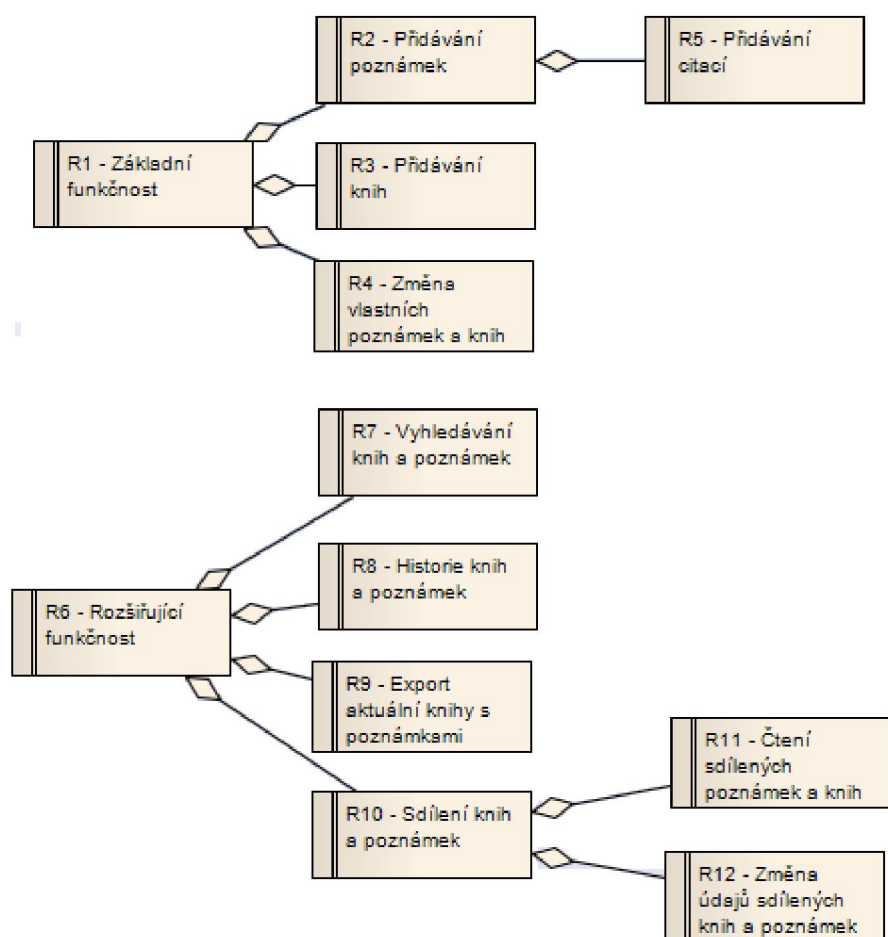


Obrázek 10 - USE CASE 2

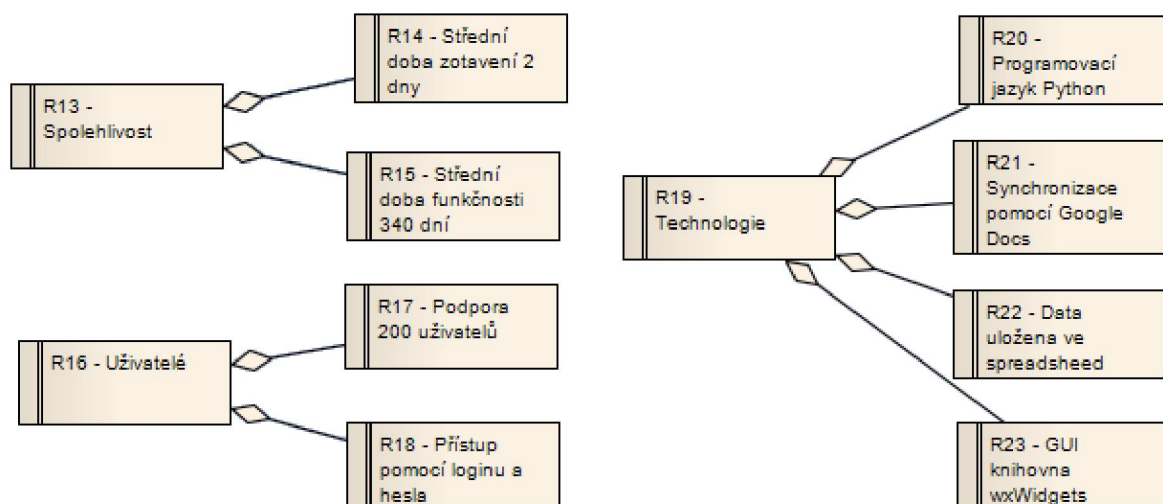
Příloha 2. Requirements model



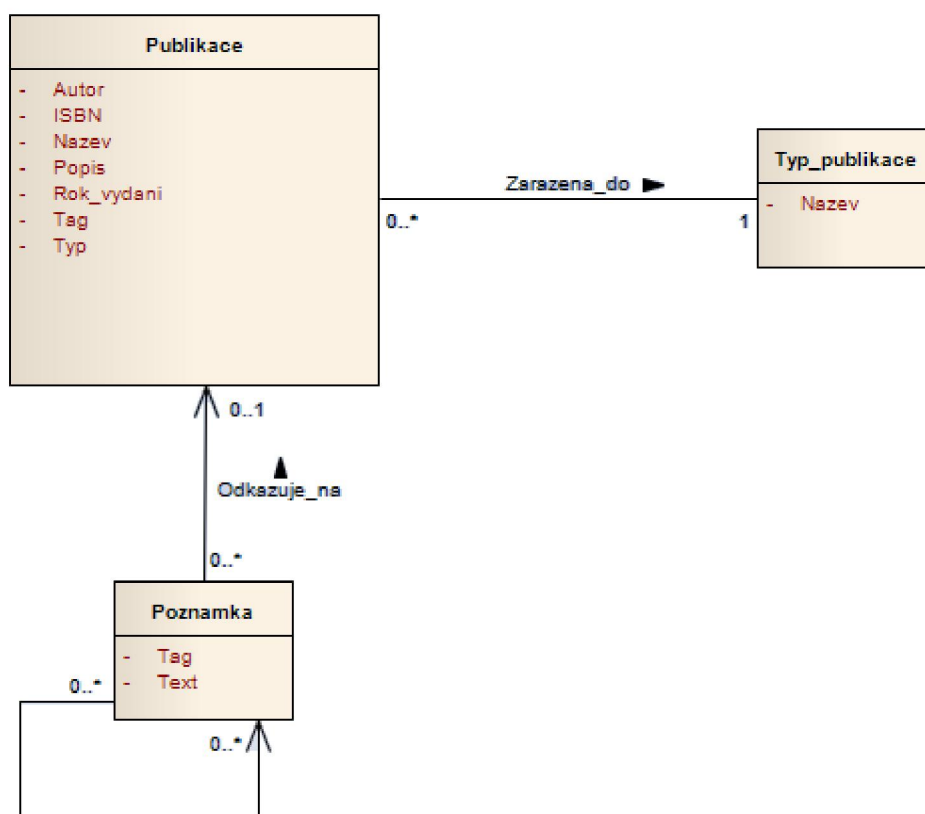
Obrázek 11 - Requirements model – funkční i nefunkční požadavky



Obrázek 12 - RM funkční požadavky



Obrázek 13 - RM nefunkční požadavky



Obrázek 14 - Doménový model