

České vysoké učení technické v Praze
Fakulta elektrotechnická

katedra počítačové grafiky a interakce

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: Antonina Lebedeva

Studijní program: Softwarové technologie a management
Obor: Web a multimedia

Název tématu: Modul správy lidských zdrojů pro systém Redmine

Pokyny pro vypracování:

Vytvořte plugin do systému Redmine pro správu lidských zdrojů na projektech, které firma realizuje. Cílem je především umožnit:

- A. Zobrazit projekty, na kterých konkrétní zaměstnanec firmy pracuje a bude pracovat.
- B. Zobrazit zaměstnance, kteří pracují a budou pracovat na vybraném projektu.
- C. Sledovat situace, kdy na projektu je nedostatek kapacit, nebo kdy zaměstnanec není dostatečně vytížen. Na tyto situace bude aplikace určenou osobu upozorňovat. Vytížení jednotlivých pracovníků v čase bude graficky znázorňováno.


Práci průběžně analyzujte, implementujte, testujte, nasazujte a dokumentujte. Dále zvolte adekvátní množství projektů, zaměstnanců, úkolů, testerů a proveďte uživatelské testování použitelnosti. Výsledky testování zapracujte.

Seznam odborné literatury:

- [1] LARMAN, Craig a Chris RUPP. Applying UML and patterns: introduction to object-oriented analysis and design and interactive development. 3rd ed. New Jersey: Prentice-Hall, 2005, xviii, ISBN 01-314-8906-2.
- [2] <http://www.redmine.org/>

Vedoucí: Ing. Martin Komárek

Platnost zadání: do konce letního semestru 2015/2016


prof. Ing. Jiří Žára, CSc.
vedoucí katedry




prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 24. 3. 2015

České vysoké učení technické v Praze
Fakulta elektrotechnická
Katedra počítačů



Bakalářská práce

Aplikace pro správu lidských zdrojů pro systém Redmine

Antonina Lebedeva

Vedoucí práce: Ing. Martin Komárek

Studijní program: Softwarové technologie a management, Bakalářský

Obor: Web a multimedia

21. května 2015

Poděkování

Rada bych poděkovala vedoucímu své bakalářské práce panu Ing. Martinu Komárkovi za cenné rady a poznámky při tvorbě této práce. Děkuji také Ing. Pavlovi Lauko za pomoc při gramatické kontrole práce, Ing. Stanislavu Kocandě za pomoc při návrhu a vývoji aplikace a také své rodině a příteli Yurovi za podporu během mého studia.

Prohlášení

Prohlašuji, že jsem práci vypracoval samostatně a použil jsem pouze podklady uvedené v příloženém seznamu.

Nemám závažný důvod proti užití tohoto školního díla ve smyslu §60 Zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Praze dne 20. 5. 2015

.....

Abstract

The goal of this thesis is to create an application for project management and human resources that should communicate with the Redmine system. As part of the thesis deals with the analysis of existing and similar solutions, design, requirements specification and I perform analysis of the development of plugins for the Redmine system. Also in this work included user interface design, implementation and testing with users.

Abstrakt

Cílem této bakalářské práce je vytvoření aplikace pro správu projektů a lidských zdrojů, která by měla komunikovat s systémem Redmine. V rámci bakalářské práce se zabývám analýzou existujících a podobných řešení, návrhem a specifikací požadavků, provádím analýzu vývoje pluginů pro systém Redmine. Také součástí této práce návrh uživatelského rozhraní, jeho realizace a testování s uživateli.

Obsah

1	Úvod	1
2	Popis problému a návrh aplikace	3
2.1	Motivace	3
2.2	Popis problému	3
2.3	Specifikace cíle řešerše	4
2.4	Analýza existujících řešení	4
2.4.1	Rešerše	4
2.4.2	Kandidáti	6
2.4.3	Diskuse s zákazníkem	9
2.4.4	Výsledek řešerše	9
2.5	Návrh aplikace a požadavky	9
2.5.1	Funkční požadavky	9
2.5.2	Obecné požadavky	10
3	Příprava k implementaci	11
3.1	Vývojové prostředí a systémové požadavky	11
3.2	Programovací jazyk Ruby	11
3.3	Framework Ruby on Rails	11
3.4	Struktura aplikace Redmine	12
3.4.1	Základní struktura adresáře	12
3.4.2	Databázový model	13
4	Analýza a návrh GUI	15
4.1	Prvotní návrh GUI	15
4.2	Připomínky zákazníka	17
4.3	Případy užití	18
4.4	Konečný návrh GUI a shrnutí	21
4.4.1	Nový design	21
4.4.2	Shrnutí	22
4.5	Návrh databázového modelu	24
4.5.1	Teoretické informace	24
4.5.2	Popis tabulek	25
4.5.3	Popis vztahu a kardinalit	27

5	Realizace	29
5.1	Plán realizace	29
5.2	Vývojové prostředí	30
5.2.1	VirtualBox	30
5.2.2	Git	30
5.2.3	Redmine	30
5.2.4	RubyGems	30
5.2.5	Rake	32
5.2.6	Rails	32
5.2.7	jQuery	33
5.3	Iterace	33
5.3.1	1.iterace [16.03. - 23.03.2015]	33
5.3.2	2.iterace [23.03. - 30.03.2015]	35
5.3.3	3. Iterace [30.03. - 06.04.2015]	36
5.3.4	4. Iterace [06.04. - 13.04.2015]	37
5.3.5	5. Iterace [13.04. - 20.04.2015]	38
5.3.6	6. Iterace [20.04. - 04.05.2015]	39
6	Testování	41
6.1	Test s uživateli	41
6.1.1	Cíle testu	41
6.1.2	Požadovaná skupina participantů	41
6.1.3	Screener	41
6.1.4	Pre a Post-test dotazníky	41
6.1.5	Nastavení testu	42
6.1.6	Seznam úkolů	42
6.1.7	Průběh testu s uživateli	43
6.1.8	Vyhodnocení testování	43
7	Závěr	47
	Literatura	49
A	Seznam použitých zkratk	51
B	Dotazníky	53
B.1	Screener	53
B.2	Pre-test dotazník	54
B.3	Post-test dotazník	54
C	Instalační a uživatelská příručka	55
C.1	Instalace	55
C.2	Uživatelská příručka	55
D	Obsah přiloženého CD	59

Seznam obrázků

2.1	Ukázka aplikace Harvest Forecast	6
2.2	Ukázka aplikace Resource Guru	7
2.3	Ukázka aplikace Resource Guru 2	7
2.4	Ukázka aplikace EasyRedmine	8
4.1	Stránka Kapacity, podstránka Tým	15
4.2	Stránka Kapacity, podstránka Projekty	16
4.3	Stránka Alokace, podstránka Tým	16
4.4	Stránka Alokace, podstránka Projekty	16
4.5	Diagram případů užití	19
4.6	Nový design, stránka Tým	21
4.7	Nový design, stránka Projekty	22
4.8	Ukázka Modálního okna, akce Přidat kapacitu uživatele	22
4.9	Databázový model	25
C.1	Screenshot aplikace - stránka Tým	56
C.2	Screenshot aplikace - stránka Tým	57
C.3	Screenshot aplikace - stránka Tým, formulář pro přidávání kapacity	57
C.4	Screenshot aplikace - stránka Tým, seznam kapacit	58

Seznam tabulek

2.1	Rešerše	5
4.1	Požadavky	23
5.1	Plán iterací	29

Kapitola 1

Úvod

Hlavní impulz pro výběr tématu mé Bakalářské práce mi poskytla zkušenost práce ve firmě zabývající se vývojem software a moje pozorování jejího řízení zevnitř.

V každé firmě, až už to začínající nebo již s mnohaletou zkušeností, vzniká problém s odpovídajícím rozdělováním lidí a jejich pracovních zdrojů na projekty s ohledem na dodržení rozpočtu a smluvních termínů.

Pro nejlepší efektivitu firmy by měli projektoví manažeři sledovat pracovní vytížení zaměstnanců. Často ale nastávají situace, kdy jeden zaměstnanec pracuje na několika projektech a nestíhá odpovídající měrou věnovat čas každému z projektů, ve stejné době, jiný pracovník té samé firmy má na starosti méně časově náročné úkoly a není dostatečně vytížen. Dokonce ani ten nejlepší projektový manažer nezvládne takový úkol vyřešit bez softwarové pomůcky. A proto, aby tomu zabránili, projektoví manažeři se uchylují k užití různých programů. Bohužel je ale těžké najít softwarovou aplikaci vyhovující ve všech požadovaných parametrech.

Požadavků na takovou aplikaci je hodně, ale nejzásadnější je jednoduchý, nepřetížený interface, integrace se současnou aplikací pro vedení projektu a automatické oznámení pro situaci, kdy je pracovník úplně uvolněn.

Kapitola 2

Popis problému a návrh aplikace

2.1 Motivace

Moji motivaci k vypracování této bakalářské práci je přání udělat užitečný nástroj, který by měl usnadnit práci projektovým manažerům a rozšířit funkčnost již existujícího systému pro řízení projektů. Myšlenka vytvoření takové aplikace patří mému kolegovi z práce, jenž plní povinnosti projektového manažera a potřebuje pomoc s tímto netriviálním úkolem. Dále se bude zmiňován termínem „zákazník“.

2.2 Popis problému

Podle zadání této bakalářské práce cílem je vytvořit aplikaci, která jednoduše umožní zobrazit projekty, na kterých konkrétní zaměstnanec firmy pracuje a bude pracovat, a také naopak zobrazit zaměstnance, kteří pracují a budou pracovat na vybraném projektu. Aplikace umožní sledovat situace, kdy na projektu je nedostatek kapacit, nebo kdy zaměstnanec není dostatečně vytížený, a bude na tyto situace správce aplikace upozorňovat. Aplikace bude komunikovat se systémem Redmine, ve kterém se budou zadávat informace o alokacích jednotlivých zaměstnanců na konkrétní projekty. Součástí této práce je také sběr a analýza požadavků a případů užití.

Pro pochopení náplně této bakalářské práce je nutné vědět, čím je systém Redmine a jaké existují možnosti komunikace s tímto systémem.

Redmine je serverová webová aplikace pro správu projektu a úkolů. Je napsán v jazyce Ruby s využitím frameworku *Ruby on Rails*. Je to multiplatformní a multidatabázový systém. Redmine je open source software a šíří se podle *GNU General Public License v2 (GPL)*.

Tento produkt poskytuje následující funkce: podpora více projektů, flexibilní systém řízení přístupů založený na rolích, flexibilní systém pro správu úkolů, Ganttův diagram a kalendář, snadné sledování času u jednotlivých projektů, podpora pluginů a mnoho dalších.

Pro splnění cíle této práci existují 2 řešení: napsat webovou aplikaci která bude komunikovat s systémem Redmine pomocí API nebo vytvořit plugin pro tento systém. Rozebereme každou možnost podrobněji.

Webová aplikace

Redmine poskytuje některé ze svých dat přes REST API. Toto rozhraní API umožňuje přístup a základní CRUD operace (vytvářet, aktualizovat, mazat). API podporuje jak XML a JSON formát. Pro implementaci aplikaci je výběr z programovacích jazyků

Plugin

Píše se ve frameworku Ruby on Rails jako plugin pro aplikaci v Ruby – Redmine. Zobrazuje se jako součást systému, využívá stejnou databázi. Nejvíce rozšířený způsob přidat funkčnost systému Redmine. Na oficiálních stránkách Redmine je kompletní seznam dostupných pluginů.

2.3 Specifikace cíle řešerše

Cílem řešerše je najít buď plugin nebo webovou aplikaci pro pohodlnou a jednoduchou správu zaměstnanců a projektů, která by co nejvíce splňovala očekávání zákazníka a následně rozšířit ten program o chybějící vlastnosti a funkce nebo případně co nejlépe specifikovat a rozšířit původní zadání. Je nutné položit důraz na schopnost aplikace komunikovat s systémem Redmine kvůli tomu, že výsledný program by měl pracovat s již stávajícími projekty a uživateli.

2.4 Analýza existujících řešení

Vzhledem k tomu, že oficiální stránky Redmine obsahují úplný archív pluginů, není potřebné hledat je v jiných zdrojích. Pro vyhledávání webové aplikace a pluginu jsem používala stejná klíčová slova a jsou to: *project management, resources management, workload, capacity management*.

2.4.1 Rešerše

Pro přehledně zobrazení výsledků hledání níže předvádím tabulku 2.1. Žlutě jsou zdůrazněny vlastnosti, které by výsledný program měl obsahovat.

		Harvest Forecast	Recource Guru	EasyRedmine	Agile Dwarf	WorkLoad ¹	Redmine Pipeline Plugin
Plugin do Redmine		NE	NE	NE/ANO	ANO	ANO	ANO
Licence (Open source)		NE	NE	ANO	ANO	ANO	ANO
Gantt Chart	Podle lidí	ANO	ANO	ANO		ANO	
	Podle projektu	ANO					
	Drag&Drop úkolů mezi lidí		ANO	ANO			
	Grafické roztažení po časové ose	ANO	ANO	ANO			
Agile Board				ANO	ANO		
Nastavení denní kapacity	Jako výchozí hodnota	ANO	ANO				
	Na každý den v týdnu		ANO				
RoadMap				ANO			
Attendance list				ANO			
Waiting list			ANO				
Capacity calculation	Při uvážení due to date	ANO	ANO				
	Při uvážení práce o víkend		ANO				
Zobrazení vytíženosti	Na časové ose	ANO	ANO	ANO			
	Na současný měsíc (v %)		ANO				ANO
	Na současný týden (v %)		ANO				ANO
	Pro každého zaměstnance		ANO				ANO
	Na celý tým		ANO				ANO
Time left	Podle lidí						ANO
	Podle projektu	ANO				ANO	ANO
	Podle druhu úkolu						ANO

Tabulka 2.1: Rešerše

Vysvětlení pojmů z tabulky :

Gantt Chart - druh pruhového diagramu pro grafické znázornění naplánování posloupnosti činností v čase.

Agile Board - implementace metody řízení projektu Kanban a její druhu.

Nastavení kapacity na den - možnost nastavit každému zaměstnanci denní kapacitu v rozmezí 1-24.

RoadMap - je plán, který přiřazuje dlouhodobě a krátkodobě cíle konkrétním technologickým řešením a pomáhá při splnění těchto cílů.

Attendance list - seznam každodenní docházky zaměstnanců.

Capacity calculation - výpočet časově náročnosti úkolů v hodinách na každý den.

Zobrazení vytíženosti - zobrazení vytíženosti zaměstnance při zvážení jeho pracovní kapacity.

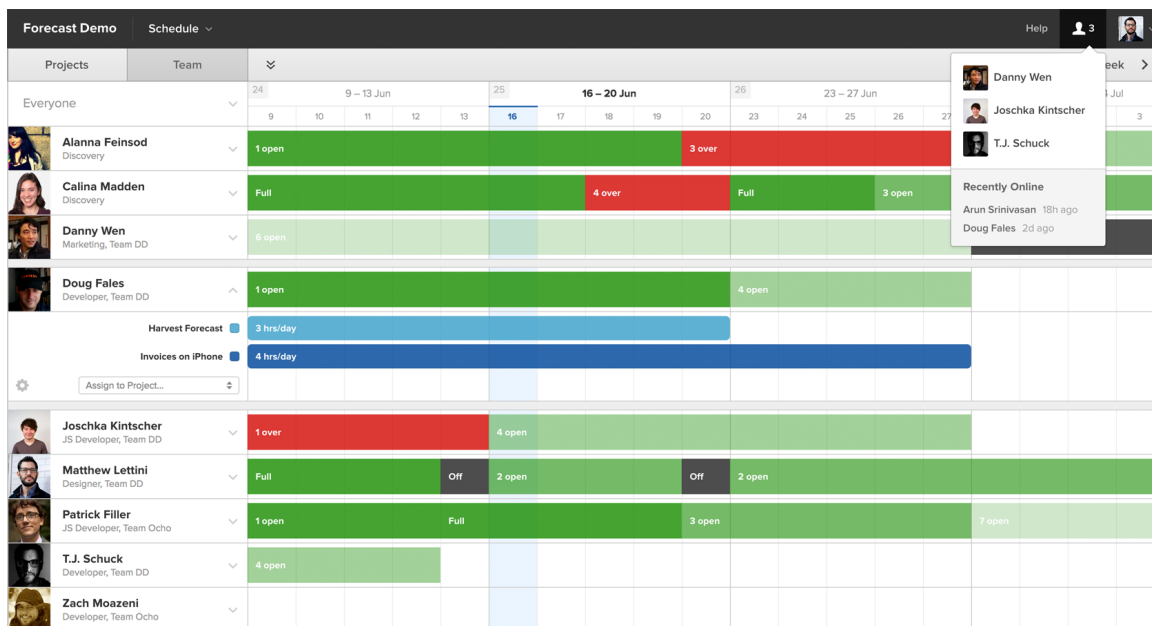
Time left - zbylý čas na dotyčný úkol. Vypočítá se jako rozdíl mezi odhadovaným časem na úkol a již zapsaným časem.

¹Nepodařilo se zprovoznit, tabulka vyplněna podle dokumentace k pluginu

2.4.2 Kandidáti

Z výše uvedené tabulky vyplývá, že co nejvíce požadovaných vlastností obsahují 3 aplikace: *Harvest Forecast*, *Recurse Guru* a *EasyRedmine*. Dále podrobněji popíšu každou z nich.

Harvest Forecast



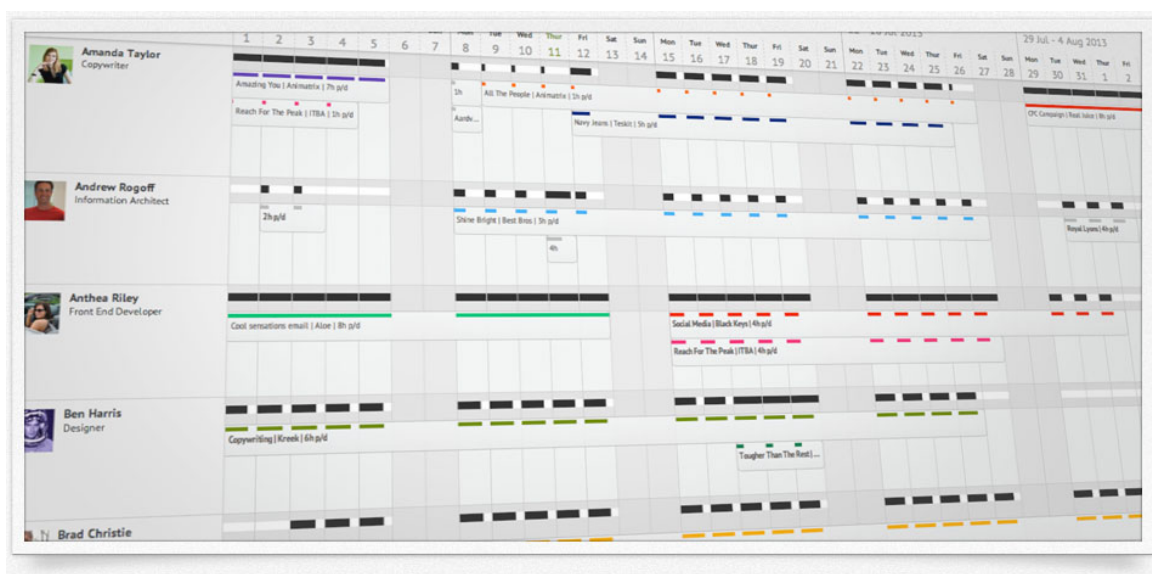
Obrázek 2.1: Ukázka aplikace Harvest Forecast

Jednoduchý software pro plánování času v týmu. Má nepřetížené a intuitivní uživatelské rozhraní. Hezká vizualizace plánů pro každého člena týmu na stejné časové ose. Není už potřeba měnit v nastavení projektu délku jeho trvání, snadno to uděláte přes grafické ovládání.

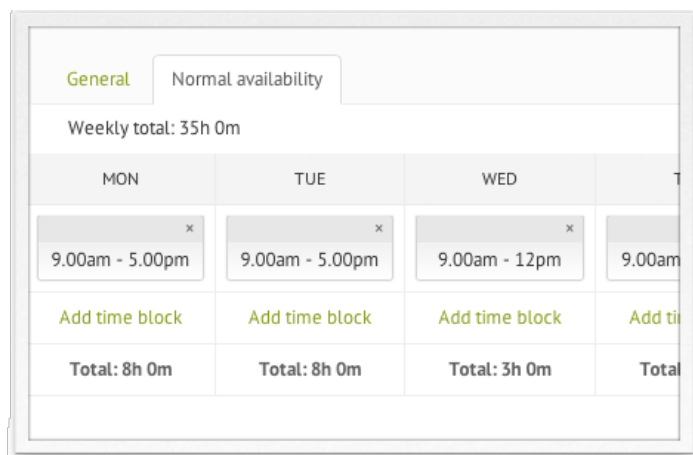
Přes všechny své výhody má několik závažných nedostatků. Především nepodporuje spojení s systémem Redmine. Nevýhodou taky je, že je placený a není open source. Znamená to, že nemám možnost rozšířit a doplnit tento program pro svůj cíl. Tato varianta se nehodí.

Resource Guru

Tato webová aplikace se ve většině funkcí shoduje s softwarem Harvest Forecast. Má ale několik užitečných funkcí navíc. Podporuje Drag&Drop zadání mezi zaměstnanci přímo na stejné časové ose. Toto vlastnost bych chtěla zdůraznit, protože docela usnadňuje práci projektového manažera.



Obrázek 2.2: Ukázka aplikace Resource Guru



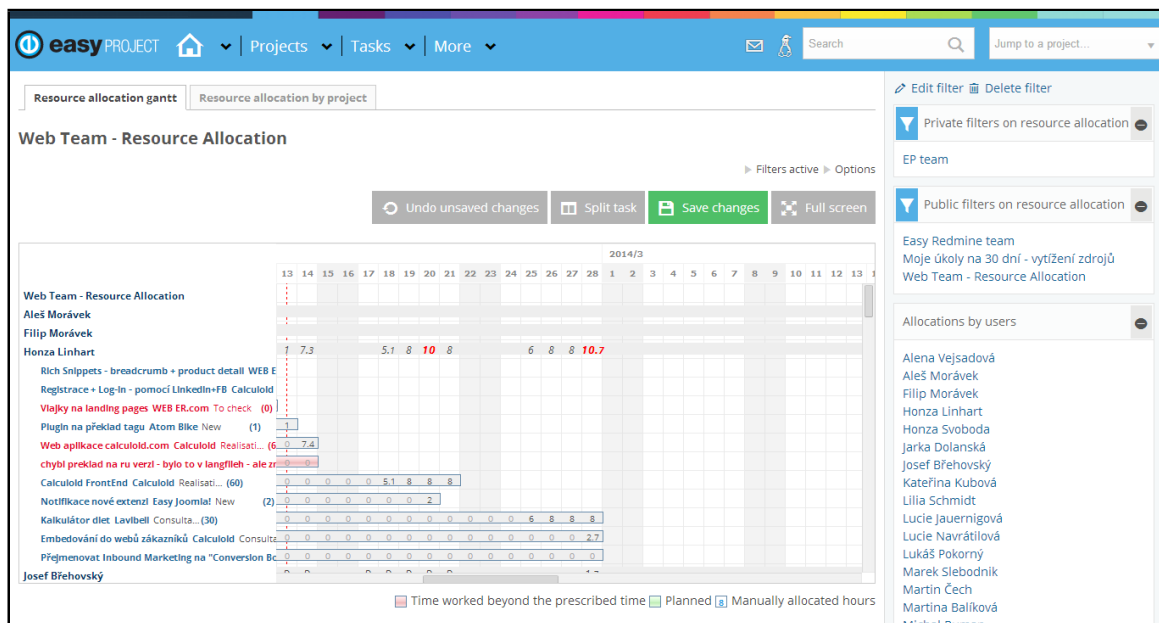
Obrázek 2.3: Ukázka aplikace Resource Guru 2

Další funkcí hodnou pozornosti je univerzální nastavení kapacity zaměstnance. Je možné buď nastavit rovnoměrnou kapacitu na týden nebo zvlášť na každý den v týdnu a s časovými úseky. viz. obrázek 2.3

Také tato aplikace umí počítat s tím, že někteří zaměstnanci mohou pracovat i o víkend. Jde o nastavit při uvedení osobní kapacity zaměstnance. Poslední odlišností, na kterou bych chtěla upozornit, je zobrazení vytíženosti zaměstnanců různými způsoby. Aplikace umí spočítat časovou dostupnost zaměstnance na každý den a ukázat tyto hodnoty na časové ose, vyčíslit hodnotu vytíženosti každého člena týmu nebo celého týmu v procentech na současný týden nebo měsíc.

Bohužel, ten software má stejné nedostatky jako předchozí kandidát - Harvest Forecast a proto je odmítnut.

EasyRedmine



Obrázek 2.4: Ukázka aplikace EasyRedmine

Komplexní software pro řízení projektu. Postaven na základě open source software Redmine. EasyRedmine přináší nejen více uživatelské přívětivě rozhraní, ale i umožňuje používat dodatečně funkce jako „řízení lidských zdrojů“, „správa financí“, „systém oznámení“ a mnoho dalších.

Firma realizující tento software poskytuje 2 varianty jeho používání: on hosting řešení (celý software se provozuje na hostingu firmy), on-premises řešení (zakoupeny software se provozuje na serveru odběratele). V případě druhé varianty uživatel má na výběr buď nainstalovat EasyRedmine načisto nebo nainstalovat EasyRedmine jako plugin k již fungujícímu Redmine. EasyRedmine podporuje přidání pluginů, které jsou také placené.

Zajímavý plugin pro moji práci je plugin pro EasyRedmine jménem "Resource Management, Utilization". Dotyčný plugin umí přiřazovat úkoly mezi zaměstnance přes Drag&Drop, má přehled o vytíženosti pracovníků za současný týden a měsíc, časovou osu, ukazující pracovní zátěž v porovnání s kapacitou zaměstnance a upozorňuje na případy, kdy zaměstnanec vytížen více než jeho denní kapacita.

Proto abych mohla rozvíjet tuto variantu je nutné získat EasyRedmine a plugin Resource Management, Utilization. Při zakoupení produktů firma poskytne zdrojové kódy pod licenci GNU Generál Public License v2 (GPL). Cena EasyRedmine pro maximálně 30 uživatelů je €399, cena pluginu je €299. Podle mě je to hlavní nevýhodou této varianty.

2.4.3 Diskuse s zákazníkem

Potom, co jsme provedli řešerši je logické projednat a dohodnout její výsledky se zákazníkem. Proto jsem provedla s ním diskuse.

Zákazník se snažil podrobněji popsat svůj koncept. Z mnou nabízených variant se nejbližší k jeho původní myšlence zdály varianty 1 a 2, t.j. Harvest Forecast a Resource Guru. Třetí varianta (EasyRedmine + plugin) byla odmítnuta kvůli nákladům, přetíženému a složitému uživatelskému rozhraní a zbytečné detailnosti. Hlavní myšlenka je mít celkový, i když pak méně přesný, přehled o pracovních možnostech a jejich rozdělení mezi jednotlivými projekty.

2.4.4 Výsledek řešerše

Rešerše existujících řešení ukázala, že pluginů pro systém Redmine, splňujících aspoň základní koncept zadání není mnoho a většinou nejsou doladěny, nemají dokumentaci, nejsou stále udržovány a nepodporují většinu požadovaných funkcí, přesto mohou sloužit jako základní platforma a výukový materiál. Také podle mého hledání neexistuje veřejně dostupná webová aplikace, komunikující s systémem Redmine a realizující základní požadavky. Existuje ale několik webových aplikací, zaměřených na správu projektu a lidských zdrojů. Téměř všechny jsou placené a nelze je rozvíjet a předělávat jejich funkčnost z toho důvodu, že vývojáři těchto aplikací neposkytují své zdrojové kódy. S ohledem na to, že pro svou práci nemůžu použít cizí zdrojový kód, zbývá se jen inspirovat existujícími aplikacemi, konkrétně Harvest Forecast a Resource Guru, a samostatně realizovat zamýšlenou funkčnost formou pluginu pro systém Redmine.

2.5 Návrh aplikace a požadavky

Na základě provedené rešerše vznikly následující funkční a nefunkční neboli obecné požadavky.

2.5.1 Funkční požadavky

1. Aplikace bude vypisovat seznam uživatelů systému Redmine a jim přiřazených projektů. Znamená to, že pod každým jménem uživatele bude seznam projektů, do nichž je přiřazen.
2. Aplikace bude vypisovat seznam projektu systému Redmine a do nich přiřazených uživatelů.
3. Aplikace bude umožňovat zadávat plánovaný čas, který pracovník by měl věnovat svému projektu každý týden.
4. Aplikace bude mít časovou osu. Ta osa bude rozdělena po týdnech a bude se začínat aktuálním datem.
5. Aplikace bude zobrazovat alokaci pracovní doby zaměstnance pro každý z jeho projektů na časové ose.

6. Aplikace bude mít možnost se posouvat po časové ose směrem "zpět", t.j. do minulých dat, a směrem "vpřed", t.j. do budoucích dat.
7. Každému z existujících v systému Redmine projektu bude přiřazena vlastní barva.
8. Alokace, zmíněná v bodě číslo 5, by měla se zobrazovat v podobě pásu barvy projektu, do kterého patří. Také by tu měla být uvedena hodnota alokaci v hodinách.
9. Aplikace bude sumarizovat alokaci na každém projektu pro každého zaměstnance.
10. Aplikace bude umožňovat nastavit týdenní pracovní kapacitu každému zaměstnanci. Bude možnost uvést datum do kterého tato konkrétní týdenní pracovní kapacita bude platit.
11. Aplikace bude ukazovat na časové ose rozdíl mezi plánovaným časem (alokací) z požadavku číslo 9 a kapacitou zaměstnance z požadavku číslo 10.
12. Aplikace by měla upozorňovat na případy, kdy týdenní plánovaný čas na úkoly překračuje denní kapacitu pracovníka. T.j. pokud hodnota z požadavků číslo 11 je více než 0.

2.5.2 Obecné požadavky

1. Aplikace bude implementována formou pluginu pro systém Redmine verzi 2.x.
2. Plugin bude využívat framework Ruby on Rails.
3. Plugin bude přístupný jenom přes systém Redmine, který je přístupný pomocí webového prohlížeče.

Kapitola 3

Příprava k implementaci

V této kapitole jsou popsána vývojová prostředí, struktura aplikace, databázový model systému Redmine a další věci, které je důležité znát při tvorbě pluginu pro systém Redmine.

3.1 Vývojové prostředí a systémové požadavky

Pro pohodlnější práci nad pluginem pro systém Redmine je potřeba mít nainstalován tento systém samotný. Jak už víme, Redmine je aplikace napsaná s využitím frameworku Ruby on Rails. Z toho plyne, že pro fungování této aplikace je nutné mít nainstalovaný Ruby interpreter odpovídající verze, kterou můžete zjistit na oficiálních stránkách Redmine, manažer závislostí pro Ruby aplikaci Bundler. Dalé je nutné mít databázi. Redmine podporuje MySQL, PostgreSQL, Microsoft SQL Server a SQLite. Redmine je multiplatformní aplikace a je možné ji provozovat na většině Unix, Linux, Mac a Windows operačních systémech. V našem případě jsem zvolila operační systém Ubuntu ve verzi 14.04 a databázový systém MySQL.

3.2 Programovací jazyk Ruby

Ruby je dynamický, reflexní, interpretovaný vysokoúrovňový programovací jazyk pro rychlé a snadné objektově orientované programování. Tento jazyk má na operačním systému nezávislou realizaci multithreading, přesnou dynamickou typizaci, garbage collector, a mnoho dalších funkcí.

3.3 Framework Ruby on Rails

Ruby on Rails, nebo zjednodušeně Rails a RoR, je framework pro vývoj webových aplikací napsaný v programovacím jazyce Ruby. Ruby on Rails poskytuje návrhový vzor Model-View-Controller pro webové aplikace, a umožňuje integraci s webovým a databázovým serverem. Je to open source software a šíří se pod licenci MIT.

Vzor Model-View-Controller odděluje části aplikace zodpovědné za čtení a ukládání dat včetně manipulace s nimi (model), za zobrazení grafického rozhraní aplikace (view) a za část přijímající vstupy od uživatele a řídící zobrazení dat na výstupu (controller).

Základní vlastností je abstraktní přístup k datům v databázi pomocí mapování záznamů z relační databáze na objekty. Takový přístup je realizován pomocí návrhového vzoru Active-Record¹. Každá tabulka se mapuje na jednu nebo více tříd na základě principu "convention over configuration" (konvence má přednost před konfigurací). Jedna z těchto konvencí je - název tabulky by měl být v množném čísle, a název třídy - v jednotném. Atributy tabulky jsou mapovány za běhu na atributy instance Ruby, t.j. „řádky“ v databázi se převedou na instance objektů, „sloupce“ na jejich atributy.

Také je důležité říct pár slov o složce View MVC vzoru. V Rails View komponenta se jmenuje Action View a obsahuje v sobě logiku, potřebnou pro zobrazení data Modelu. Konečný výstup HTML je kompozice třech Rails prvků: Templates, Partials, Layouts.

Templates

Šablony Action View mohou být psané několika způsoby. Má-li soubor šablony.erb rozšíření, pak se používá směs ERB (Embedded Ruby) a HTML. Pokud má soubor .builder rozšíření pak je použita knihovna Builder :: XmlMarkup.

Rails podporuje více systémů šablon a používá příponu souboru aby mezi nimi rozlišoval. Například soubor HTML doplněný pomocí šablonovacího systému ERB bude mít .html.erb jako příponu souboru.

Partials

Partial templates - obvykle jen nazývané "partials"- jsou dalším nástrojem pro rozdělení procesu renderingu do více zvládnutelných kousků. S partials, můžete vytahovat kusy kódu z šablon do jednotlivých souborů, a také je znovu používat v šablonách.

Layouts

Definují jak má být obsah umístěn na stránce. Dynamicky tvořená stránka může obsahovat vnoření z několika stran, a to i bez použití tabulek a rámců, jen pomocí layout API.

Rails má v sobě tři prostředí - jedno pro každou z etap životního cyklu vývoje aplikace: development (vývoj), test (testování) a production (provoz), pro každou z nichž se vytváří samostatná databáze.

3.4 Struktura aplikace Redmine

3.4.1 Základní struktura adresáře

Redmine, jak již bylo zmíněno, je Ruby on Rails aplikation a tudíž má standartní základní strukturu adresáře jako každá RoR aplikace.

¹ORM framework, vestavěný v framework Ruby on Rails

- **app** - tato složka organizuje komponenty aplikace. Má v sobě podadresáře, které obsahují šablony (app/views), helpery (app/helpers), kontrolery (app/controllers), a backend byznys logiku (app/models)
- **app/view** - taky zahrnuje podadresáře pro každý z kontroleru, ve kterém jsou soubory šablon pro každou z action dotyčného kontroleru
- **components** - v minulosti udržovala součásti malé samostatné aplikace, která spojovala view, model a controller
- **config** - tento adresář obsahuje malé množství konfiguračního kódu, který aplikace bude potřebovat, včetně konfigurace databáze (v database.yml), strukturu Rails prostředí (environment.rb), a směrování příchozích webových požadavků (routes.rb)
- **db** - konfigurace databázových tabulek je definována zde
- **doc** - tady je celá RubyDoc generovaná dokumentace
- **lib** - adresář pro umístění užitečných knihoven
- **log** - zde naleznete chybové hlášky. Rails vytváří skripty, které vám pomohou spravovat různé chybové hlášky. Najdete tady samostatné protokoly pro server (server.log) a pro každé prostředí Rails (development.log, test.log a production.log)
- **public** - stejně jako veřejné adresář pro webový server, tento adresář udržuje soubory, které se nemění, jako jsou soubory JavaScript (public/javascripts), grafiky (public/images), styly (public/stylesheets), a HTML soubory (public)
- **script** - tento adresář obsahuje skripty pro spuštění a správu různých nástrojů, které budete používat s Rails
- **test** - zde jsou uloženy testy aplikace
- **tmp** - Rails používá tento adresář pro udržování dočasných souborů jako session uživatele a PID soubory
- **vendor** - adresář pro pluginy třetích stran

3.4.2 Databázový model

Jelikož náš plugin bude komunikovat s daty v systému Redmine, je užitečné znát strukturu a vztahy mezi databázovými tabulkami. Oficiální stránky Redmine naštěstí poskytují kompletní databázové schéma, ale bez detailní dokumentace. Toto schéma je k dispozici v referencích [?]

Kapitola 4

Analýza a návrh GUI

4.1 Prvotní návrh GUI

Při tvorbě návrhu GUI jsem se ho snažila udělat co jednodušší a s důrazem na konečného uživatele. Jako inspirace mi posloužily už zmíněné aplikace Harvest Forecast a ResourceGuru.

Plugin má 2 základní rozdělení: to jsou Kapacity a Alokace. Stránka „Kapacity“ je určena pro zadávání týdenní pracovní kapacity zaměstnanců a také týdenní kapacity projektů, t.j. nutné množství pracovních hodin na projekt. Stránka „Alokace“ je určena pro správu alokací času pracovníků na projektech a zobrazení rozdílu mezi kapacitou a alokací. Každá z těchto dvou stránek na další rozdělení na „Tým“ a „Projekty“. „Tým“ ukazuje data z pohledu lidí, t.j. kapacitu a alokaci každého zaměstnance. Podstránka „Projekty“ obsahuje jiný pohled - podle projektů, jsou tam kapacity projektů a alokace lidí na projektech.

Následující wireframy¹ ukazují výše popsané stránky:

Kapacity		Alokace						
Tým	Projekty	+						
←	Aktuální týden	→	2 - 6 Únor	9 - 13 Únor	16 - 20 Únor	23 - 27 Únor	2 - 6 Březen	9 - 13 Březen
Antonína Lebedeva	25 hodin	Volno						
Marcella Novakova		32 hodin					20 hodin	
Peter Prochazka	40 hodin	Volno					35 hodin	

Obrázek 4.1: Stránka Kapacity, podstránka Tým

¹návrh definující funkci a obsah stránek webu

Kapacity		Alokace						
Tým	Projekty	+						
←	Aktuální týden	→	2 - 6 Únor	9 - 13 Únor	16 - 20 Únor	23 - 27 Únor	2 - 6 Brezen	9 - 13 Brezen
PROJEKT 1			50 hodin					
Nový projekt 2			60 hodin		50 hodin			
Zakazka 3			45 hodin		35 hodin			

Obrázek 4.2: Stránka Kapacity, podstránka Projekty

Kapacity		Alokace						
Tým	Projekty	+						
←	Aktuální týden	→	2 - 6 Únor	9 - 13 Únor	16 - 20 Únor	23 - 27 Únor	2 - 6 Brezen	9 - 13 Brezen
Antonina Lebedeva >			1 hod. volne		Volno			
Marcella Novakova >			10 hod. volne		cely vytizeny			
Peter Prochazka v			3 hod. volne		Volno		3 hod. pres	
Nový projekt 2			20 hod		20 hod			
Zakazka 3			17 hod		18 hod			

Obrázek 4.3: Stránka Alokace, podstránka Tým

Kapacity		Alokace						
Tým	Projekty	+						
←	Aktuální týden	→	2 - 6 Únor	9 - 13 Únor	16 - 20 Únor	23 - 27 Únor	2 - 6 Brezen	9 - 13 Brezen
PROJEKT 1 >			1 hod. volne					
Nový projekt 2 >			10 hod. volne		10 hod. volne			
Zakazka 3 v			3 hod. volne		20 hod. volne		27 hod. volne	
Peter Prochazka			17 hod		volno		18 hod	
Antonina Lebedeva			25 hod		volno			

Obrázek 4.4: Stránka Alokace, podstránka Projekty

Na těchto stránkách se také vyskytují další akce:

- Tlačítko „+“ – odkazuje na stránku s jednoduchým formulářem pro vložení kapacity nebo alokace
- Tlačítka „«“ a „»“ – slouží pro posouvání po časové ose do budoucna a do minulosti
- Tlačítko „Aktuální týden“ – přemístí aktuální týden na začátek časové osy
- Jméno zaměstnance je odkazem na stránku s seznamem všech kapacit/alokací odpovídajících danému člověku. Na této stránce je možné editovat -předvyplněný formulář, a mazat jeho záznamy o kapacitě/alokaci
- Vedle každého zaměstnance nebo projektu je tlačítko „>“ - slouží pro otevření/zavření dodatečných informací o alokaci

Z uvedených wireframů vyplývá několik funkčních požadavků, které nebyly uvedené v kapitole 2.5.1:

1. Přepínání mezi stránkami „Tým“ a „Projekty“ pomocí záložek
2. Posunutí aktuálního týdne na začátek osy tlačítkem
3. Označení aktuálního týdne na ose
4. Tlačítko Ukázat/Skrýt dodatečnou informaci

4.2 Připomínky zákazníka

Výše ukázané wireframy byly představené a popsané zákazníkovi. Poznamenal tento design docela jednoduchým a upozornil na nutnost většinu dat zadávat ručně, což může způsobovat omyly a překlepy. Některé stránky byly vyhodnoceny jako zbytečné, stránky „Alokace“ a „Kapacity“ by bylo možné spojit. Přejít na novou stránku pro přidání kapacity/alokaci nebo pro správu seznamu kapacit/alokaci uživatele/projektu byl označen jako nepohodlí pro uživatele. V průběhu diskuse, zákazník upozornil na velmi důležitou situaci, kdy zaměstnanec pracuje vždy stejný čas a bylo by zbytečně zadávat časově údaje „od“ a „do“. Chtělo by to mít výchozí nastavení kapacity, ale s možností přidávat nulovou kapacitu na období volna. Níže jsou sepsané nové požadavky od zákazníka:

1. Funkce „Vytvořit novou kapacitu/alokaci“ bude zobrazena v modálním okně ²
2. Funkce „Správa kapacit/alokaci zaměstnance/projektu“ bude také v podobě modálního okna

²V grafickém uživatelském rozhraní modálním oknem se nazývá okno, které blokuje práci uživatele s rodičovskou aplikací pokud uživatel nezavře okno

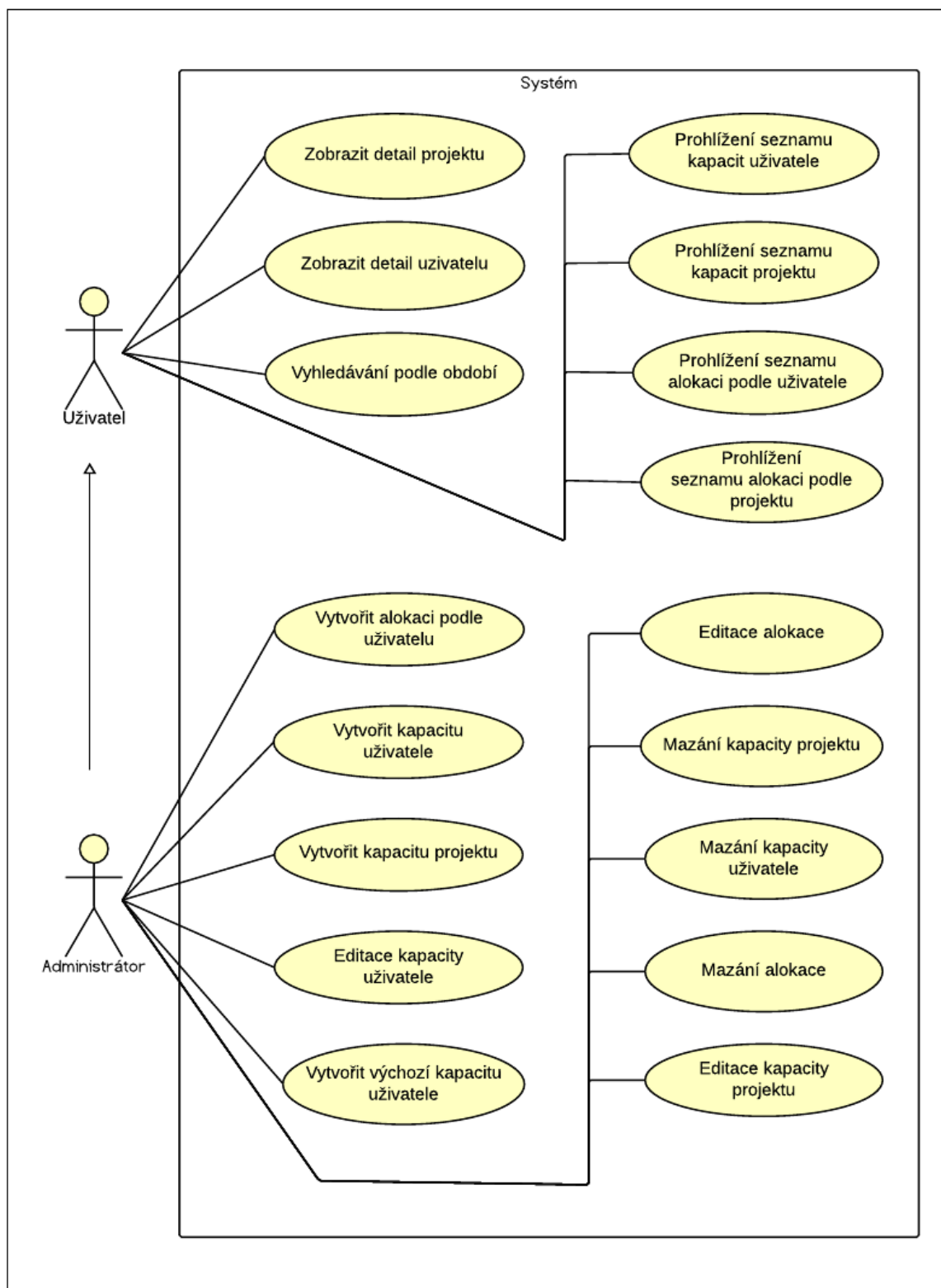
3. Stránky „Kapacity“ a „Alokace“ sloučit do jedné stránky s dvěma podstránkami „Tým“ a „Projekty“
4. Vytvořit možnost posouvání po časové ose na určitý datum nebo ukazovat určité období jako „tento měsíc“ „minulý měsíc“
5. Přizpůsobit počet týdnů na časové ose šířce okna prohlížeče
6. Výchozí nastavení kapacity pro zaměstnance
7. Přidat možnost Drag&Drop pruhu, odpovídajících za zobrazení kapacity/alokace na časové ose
8. Přidat možnost roztažení pruhu kapacity/alokaci po časové ose
9. Omezení ve výběru dat v kalendáře na stránce vytvoření a editace

4.3 Případy užití

Diagram případů užití je určen pro znázornění operaci, které mohou vykonávat uživatelé systému. Používá následující elementy:

- **Hranice systému** – obdélník s názvem v horní části a s elipsami uvnitř.
- **Aktér** – figura, popisuje objekty vstupující do vztahů s systémem. Například uživatel.
- **Případ užití** – elipsa, označuje funkci systém.
- **Relace «include»** - čerchovaná šipka, vyjadřuje situaci, kdy jeden případ užití zahrnuje v sobě další případ užití nebo více.
- **Relace «extend»** - čerchovaná šipka, vyjadřuje situaci, kdy jeden případ užití může rozšiřovat jiný.
- **Zobecnění/dědění** – šipka s bílým koncem, zobrazuje vztah dědění mezi aktéry nebo případy užití. Používá se pro zjednodušení diagramu.

Na následujícím obrázku 4.5 je zobrazen digram případů užití. Jsou tam 2 aktéři: administrátor a libovolný uživatel systému, dokonce nepřihlášený. Administrátor se dědí od uživatele, to znamená že může vykonávat ty samé akce jako uživatel, ale má navíc svoje akce, které obyčejný uživatel nemá právo provádět. Dále jednotlivě popisuje každý případ užití:



Obrázek 4.5: Diagram případů užití

1. **„Zobrazit detail projektu“** – na stránce „Projekty“ je možnost si rozkliknout podrobnější informací o projektu, takovou jako kapacita projektu a alokace podle uživatelům.
2. **„Zobrazit detail uživatele“** – podobná funkce, ale na stránce „Tým“. Zobrazuje kapacitu uživatele a jeho alokace podle projektům.
3. **„Vyhledávání podle období“** – v podstatě, to je posouvání po časové ose na určitý datum: začátek minulého měsíce nebo uživatelem zvolený datum.
4. **„Prohlížení seznamu kapacit uživatele“** – tato akce dostupná jen ze stránky „Tým“ vypíše všechny kapacity zvoleného uživatele od nejnovější.
5. **„Prohlížení seznamu kapacit projektu“** – akce dostupná na stránce „Projekty“ vypíše kapacity zvoleného projektu od nejnovější.
6. **„Prohlížení seznamu alokaci podle uživatele“** – funkce se nachází na stránce „Tým“ vypíše všechny alokaci daného uživatele od nejnovější.
7. **„Prohlížení seznamu alokaci podle projektu“** – podobná funkce, nachází se na stránce „Projekty“ vypíše všechny alokaci daného projektu od nejnovější.
8. **„Vytvořit alokaci podle uživatele“** – na začátku se vybere uživatel, pak projekt ze seznamu mu přiřazených projektů. Akce dostupná ze stránky „Tým“.
9. **„Vytvořit alokaci podle projektu“** – opačná akce, na začátku se vybere projekt, pak uživatel ze seznamu mu přiřazených uživatelů. Akce dostupná ze stránky „Projekty“.
10. **„Vytvořit kapacitu uživatele“** – akce na stránce „Tým“ vytvoření nové kapacity uživatele.
11. **„Vytvořit výchozí kapacitu uživatele“** – akce na stránce „Tým“ velmi podobná akci „Vytvořit kapacitu uživatele“ s tím rozdílem, že není políčko „Do“.
12. **„Vytvořit kapacitu projektu“** – akce na stránce „Projekty“ vytvoření nové kapacity projektu.
13. **„Editace kapacity uživatele“** – standardní akce editace kapacity uživatele, bez možnosti měnit uživatele.
14. **„Mazání kapacity uživatele“** – standardní akce mazání.
15. **„Editace kapacity projektu“** – standardní akce editace kapacity projektu, bez možnosti měnit projekt.
16. **„Mazání kapacity projektu“** – standardní akce mazání.
17. **„Editace alokace“** – standardní akce editace alokace, bez možnosti měnit projekt a uživatele.
18. **„Mazání alokace“** – standardní akce mazání.

4.4 Konečný návrh GUI a shrnutí

4.4.1 Nový design

Na základě připomínek a upozornění byl navržen nový design, který byl pak úspěšně schválen zákazníkem. Pro realizaci požadavku číslo 3 z bodu 4.2, t.j. sloučení stránek „Kapacity“ a „Alokace“ jsem se rozhodla použít původní návrh stránky „Alokace“ a rozšířit ho o informace o kapacitách a možnosti tuto kapacitu přidávat.

Většina funkčnosti zůstává, popíšu změny a doplnění.

1. Tlačítko „**Nova kapacita**“ – pro vytvoření nové kapacity pro zaměstnance v případě stránky „Tým“ a pro projekt na stránce „Projekty“
2. Tlačítko „**Nova alokace**“ – pro vytvoření nové alokaci zaměstnance na projektu. Toto tlačítko je na obou stránkách stejné.
3. **Pruh**, zobrazující kapacitu uživatele/projektu v části doplňkové informace.
4. Tlačítko „**Přidat do projektu**“ na stránce „Tým“ – pro vytvoření alokace pro daného uživatele.
5. Tlačítko „**Přidat uživatele**“ na stránce „Projekty“ – pro vytvoření alokace na daném projektu.
6. **Rozbalovací menu** pro výběr období zobrazujícího na časové ose.
7. Formuláře pro editaci a vytváření se zobrazují v podobě „**Modálního**“ okna.

Níže jsou uvedeny wireframy zobrazující popsané změny:

Tým		Projekty					
<input type="button" value="Nova kapacita"/> <input type="button" value="Nova alokace"/>		<input type="text" value="Minuly mesic"/>					
<input type="button" value="Aktualni tyden"/>		2 - 6 Únor	9 - 13 Únor	16 - 20 Únor	23 - 27 Únor	2 - 6 Brezen	9 - 13 Brezen
Antonina Lebedeva	>	1 hod. volne		Volno			
Marcella Novakova	>		10 hod. volne			cely vytizeny	
Peter Prochazka	>	3 hod. volne	Volno	3 hod. pres		35 hod. volne	
Celkova kapacita :		40 hod	0 hod			35 hod	
<input type="button" value="Pridat do projektu"/>							
Novy projekt 2		20 hod		20 hod			
Zakazka 3		17 hod		18 hod			

Obrázek 4.6: Nový design, stránka Tým

Tým		Projekty					
<input type="button" value="Nova kapacita"/> <input type="button" value="Nova alokace"/>		<input type="text" value="Minuly mesic"/>					
<input type="button" value="Aktualni tyden"/>		2 - 6 Únor	9 - 13 Únor	16 - 20 Únor	23 - 27 Únor	2 - 6 Brezen	9 - 13 Brezen
PROJEKT 1	>			1 hod. volne			
Novy projekt 2	>		10 hod. volne			5 hod. volne	
Zakazka 3	∨	3 hod. volne	20 hod. volne	27 hod. volne		45 hod. volne	
Celkova kapacita :		45 hod		35 hod			
<input type="button" value="Pridat uzivatele"/>							
Peter Prochazka		17 hod	volno	18 hod			
Antonina Lebedeva		25 hod		volno			

Obrázek 4.7: Novy design, stránka Projekty

Tým		Projekty					
<input type="button" value="Nova kapacita"/> <input type="button" value="Nova alokace"/>		<input type="text" value="Minuly mesic"/>					
<input type="button" value="Aktualni tyden"/>		2 - 6 Únor	9 - 13 Únor	16 - 20 Únor	23 - 27 Únor	2 - 6 Brezen	9 - 13 Brezen
Antonina Lebedeva	>						
Marcella Novakova	>					cely vytizeny	
Peter Prochazka	∨	3				35 hod. volne	
Celkova kapacita :		35 hod					
<input type="button" value="Pridat do projektu"/>							
Novy projekt 2							
Zakazka 3							

Pridat novou kapacitu

Uzivatel

Kapacita

Tyden od

Tyden do

Obrázek 4.8: Ukázka Modálního okna, akce Přidat kapacitu uživatele

4.4.2 Shrnutí

Jako výsledek jsem dostala seznam požadavků a je potřeba je roztrždit podle priority realizace. Pro tento účel jsem vyrobila tabulku, která je uvedena níže. Některé požadavky uvedené v tabulce jsou výsledkem sloučení dvou či třech menších požadavků.

Vysvětlení priorit:

- **Vysoká** – požadavky, které jsou základem aplikace, tvoří její byznys logiku.
- **Střední** – požadavky, usnadňující používání základních funkcí aplikace.
- **Nízká** – požadavky, které zvyšují pohodlí použití aplikace.

#	Požadavek	Priorita
1	Výpis seznamu uživatelů a projektů, do kterých jsou přiřazeny	Vysoká
2	Výpis seznamu projektů a do nich přiřazených uživatelů	Vysoká
3	Přidání alokace zaměstnanců na projektu	Vysoká
4	Zobrazení časové osy, rozdělenou po týdnech	Vysoká
5	Posouvání po časové ose do budoucích a minulých období	Vysoká
6	Zobrazení trvání alokace a její hodnoty v podobě pruhu na časové ose	Vysoká
7	Nastavení týdenní kapacity pracovníků	Vysoká
8	Nastavení týdenní kapacity projektů	Vysoká
9	Zobrazení trvání kapacity a její hodnoty v podobě pruhu na časové ose	Vysoká
10	Zobrazení rozdílu mezi kapacitou a alokací na pruhu na časové ose	Vysoká
11	Upozornění na překročení kapacity	Vysoká
12	Výpis seznamu kapacit uživatele s možností editace a mazání	Vysoká
13	Výpis seznamu kapacit projektu s možností editace a mazání	Vysoká
14	Výpis seznamu alokací s možností editace a mazání	Vysoká
15	Omezení práv podle role	Vysoká
16	Každému projektu bude přiřazena vlastní barva	Střední
17	Možnost posouvat aktuální týden na začátek časové osy	Střední
18	Zvýraznění aktuálního týdne na časové ose	Střední
19	Možnost ukázat/skrýt dodatečnou informaci	Střední
20	Posunutí na určitý datum nebo období na časové ose	Střední
21	Stránky pro editaci a vytvoření v podobě „Modálního“ okna	Střední
22	Možnost výchozího nastavení kapacity	Střední
23	Možnost přidat alokaci pro určitý projekt (tlačítko „Přidat uživatele“)	Střední
24	Možnost přidat alokaci pro určitého uživatele (tlačítko „Přidat do projektu“)	Střední
25	Přizpůsobit množství týdnů na časové ose k šířce prohlížeče	Nízká
26	Drag&Drop pruhu na časové ose	Nízká
27	Možnost roztážení pruhu na časové ose	Nízká
28	Omezení výběru datumů v kalendáři při editace/vytvoření kapacity/alokace	Nízká

Tabulka 4.1: Požadavky

Při vývoji této aplikace se budu řídit inkrementální a hlavně iterativní metodikou vývoje

softwaru. Inkrementální přístup spočívá v tom, že celý projekt se rozděluje na menší části a tím se zjednodušuje možnost zavedení změn během procesu vývoje. Iterativní přístup rozděluje vývoj na iterace, které definují požadavky a termín ke kterému by měly být hotové. Po každé iteraci vznikne dokončený funkční produkt, bude otestován před odevzdáním, ale nebude mít úplnou funkcionalitu. V další kapitole nastíním plán jednotlivých iterací.

4.5 Návrh databázového modelu

4.5.1 Teoretické informace

Před realizací je třeba navrhnout databázový model aplikace. Databázový model v sobě zachycuje tabulky jednotlivých entit, jejich atributy a vztahy mezi nimi. Krátce popíšu základní termíny spojené s pojmem “Databázový model”:

- **Databázová tabulka** neboli jen **tabulka** – základní objekt databáze (entita), představuje dvourozměrnou tabulku s pevným počtem sloupců a slouží k uložení dat.
- **Sloupec** – definuje sloupec v tabulce a zároveň je jejím atributem, t.j. popisuje vlastnosti entity.
- **Záznam** – je řádek tabulky s určitými hodnotami, odpovídajícími typu sloupců.
- **Primární klíč** – atribut(sloupec) jednoznačně identifikující záznam tabulky.
- **Cizí klíč** – sloupec definuje, že jeho hodnota je odkazem na hodnotu sloupce jiné tabulky. Velmi často se odkazuje na primární klíč jiné tabulky.
- **Kardinalita vztahu** vlastnost vztahu, která určuje počet instancí entit ve vztahu. Existují 3 základní kardinality:

1:1 - každé entitě odpovídá maximálně jedna další entita (každý člověk má pouze jedno rodné číslo)

1:N - první entitě odpovídá více než jedna druhá entita, druhé entitě odpovídá maximálně jedna první entita (čtenář si půjčí více knih, ale 1 kniha je čtena 1 čtenářem)

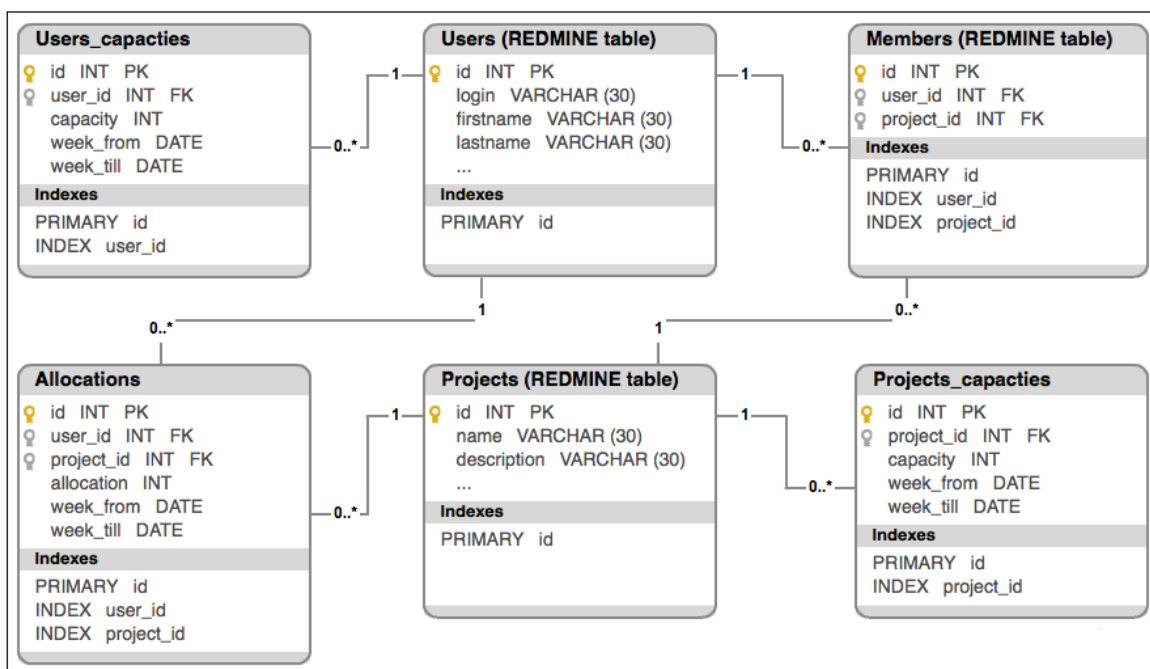
M:N - první entitě odpovídá více než jedna druhá entita, druhé entitě odpovídá více než jedna první entita (učitel vyučuje v několika třídách a ve třídě učí několik učitelů)

Na dále představeném obrázku 4.9 je vidět datový model naší aplikace. Chtěla bych zopakovat, že framework Ruby on Rails má své pravidla a konvence. Jedna z nich je takzvaná „Convention over Configuration“, to znamená, že pokud se programátor bude řídit konvencemi Rails, nebude nucen psát hodně konfigurací, občas ji vůbec nepotřebuje. Standardně ActiveRecord používá konvenci pojmenování pro stanovení souvislosti mezi modely a databázovými tabulkami. Pravidla pojmenování jsou takové:

- **Databázová tabulka** – jméno je v plurálu, slova jsou rozdělené podtržítkem.
- **Třída modelu** – název je v singuláru, používá se styl „CamelCase“.

Také ActiveRecord používá konvenci pojmenování pro sloupce v databázových tabulkách, v závislosti na účelu těchto sloupců. Pravidla jsou:

- **Cizí klíč** se jmenuje podle vzorce `singularized_table_name_id`. To jsou pole, která hledá ActiveRecord při vytváření spojení mezi modely.
- **Primární klíč** - ve výchozím nastavení ActiveRecord používá sloupec s názvem `id` a typem `int` jako primární klíč tabulky.



Obrázek 4.9: Databázový model

4.5.2 Popis tabulek

Tabulka „Users“

Již existující tabulka v databázi Redmine. Popisuje uživatele systému Redmine, obsahuje vlastnosti takové jako:

- `id` (primární klíč)
- `login` (přihlašovací jméno)
- `firstname` (jméno)
- `lastname` (příjmení)
- a další

Z této tabulky nás zajímají vlastnosti `firstname`, `lastname` a `id`.

Tabulka „Project“

Již existující tabulka v databázi Redmine. Tato tabulka reprezentuje projekt v systému. Má atributy:

- `id` (primární klíč)
- `name` (název)
- `description` (popis)
- a další

Hlavně nás budou zajímat `name` a opět `id`.

Tabulka „Member“

Již existující tabulka v databázi Redmine. Tabulka popisuje vztah mezi entitami `User` a `Project`. Každý uživatel systému Redmine může být přidán do projektu. Obsahuje jen 3 políčka a to jsou:

- `id` (primární klíč)
- `user_id` (cizí klíč do tabulky `Users`)
- `project_id` (cizí klíč do tabulky `Projects`).

Tato tabulka je nutná nám pro výpis seznamu uživatelů a jejím přidělených projektu a pro opačný výpis - projektu a uživatelů do nich přiřazených.

Tabulka „Users _capacities“

Tato tabulka představuje týdenní kapacitu zaměstnance, t.j. počet hodin, které může věnovat práci. Má následující atributy:

- `id` (primární klíč)
- `zaměstnanec` - `user_id` (cizí klíč do tabulky `Users`)
- hodnotu kapacity - `capacity`
- týden od - `week_from`
- týden do - `week_till`

Týden reprezentován jako typ `date` a je pondělím tohoto týdne.

Tabulka „Projects _capacities“

Tabulka slouží pro ukládání dat o kapacitě projektu, t.j. kolik hodin týdně projekt potřebuje pro splnění a podle rozpočtů. Sloupce této tabulky jsou:

- id (primární klíč)
- project_id (cizí klíč do tabulky „Projects“)
- hodnotu kapacity - capacity
- týden od - week_from
- týden do - week_till

Týdny od a do mají stejný typ a význam jako v tabulce „Users _capacities“

Tabulka „Allocations“

To je tabulka pro uložení informace o alokaci zaměstnance na projektu. Má následující vlastnosti:

- id (primární klíč)
- user_id (cizí klíč do tabulky „Users“)
- project_id (cizí klíč do tabulky „Projects“)
- hodnotu alokace - allocation
- týden od - week_from
- týden do - week_till

week_from a week_till mají stejný typ a význam jako v tabulce „Users _capacities“

4.5.3 Popis vztahu a kardinalit

Vztah „Users - Users _capacities“

To je vztah mezi uživateli a kapacitou uživatele. Jedna se o vztah typu 1:N. Znamená to, že každý uživatel může mít libovolné množství kapacit, které budou se lišit svým obdobím. A jedna kapacita naopak může mít jen jednoho uživatele.

Vztah „Projects - Projects _capacities“

Vztah mezi projektem a kapacitou projektu. Je podobný vztahů „Users - Users _capacities“. Typ toho vztahů je 1:N. To jest každý projekt může mít několik kapacit v různém období, ale kapacita má jen jeden projekt.

Vztah „Users - Allocations“

Jedná se o vztah mezi entitami uživatel a alokace. Jeho kardinalita je 1:N. To znamená, že každý uživatel může mít hodně alokaci definovaných týdnů od a do. Alokace ale má jen jednoho uživatele.

Vztah „Projects - Allocations“

Tabulka projektu spojená s tabulkou alokací vztahem typu 1:N. Podobný se vztahem „Users - Allocations“ Jedna alokace může mít jen jeden projekt, ale projekt může mít více alokací.

Vztah „Users - Members“

Vztah mezi uživateli a vazební tabulkou „Members“ Kardinalitu má 1:N. To jest jeden záznam tabulky „Members“ může mít jen jeden záznam v tabulce uživatele, ale jeden uživatel může mít několik záznamů v tabulce „Members“

Vztah „Projects - Members“

Stejně jako v předchozím vztahu „Users - Members“ kardinalita je 1:N a pro jeden řádek tabulky projektů je 0 až N odpovídajících řádků v tabulce „Members“ a naopak pro jeden řádek „Members“ je jen jeden možný řádek projektu.

Kapitola 5

Realizace

5.1 Plán realizace

Zvolila jsem postupovat iterativní metodou vývoje softwaru a proto je třeba naplánovat si každou iteraci. Po každé iteraci na výstupu bude poskytnuta na kontrolu určitá část funkcionality aplikace a pak budou pro danou iteraci sebrány připomínky a případná hlášení chyb od zákazníka. Tyto poznámky budou součástí zadání dalších iterací. Interval pro iterace jsem zvolila 1 týden, protože podle mého odhadu to je optimální doba pro realizaci 3-4 požadavků a čím rychleji zákazník dostane požadavky na otestování, tím rychleji budu znát jeho názor a budu podle něj schopna opravit aplikace. Z toho vyplývá, že aplikace bude dokončena rychleji.

V níže představené tabulce 5.1 jsou rozepsané jednotlivé iterace. Každá iterace má vlastnosti:

- **Číslo iterace** – pořadové číslo iterace.
- **Požadavky** – požadavky k iteraci nebo čísla funkčních požadavků z tabulky 4.1, které je potřeba splnit. Pokud požadavky z dané iterace budou splněné dříve než v očekávaném termínu, je třeba pokračovat na požadavcích z další iterace.
- **Termín splnění** – určuje datum, ke kterému požadavky musí být splněné.

Číslo iterací	Požadavky	Termín splnění
1. iterace	Číslo 1, 4, 5, 7, 9, 12	23.03.2015
2. iterace	Číslo 2, 3, 6, 8	30.03.2015
3. iterace	Číslo 10, 11, 13, 14	06.04.2015
4. iterace	Číslo 15, 16, 17, 18, 19	13.04.2015
5. iterace	Číslo 21, 22, 23, 24	20.04.2015
6. iterace	Číslo 20, 25, 27 + doladění verzi k testování použitelnosti + příprava k testování použitelnosti	04.05.2015
7. iterace	Testování použitelnosti + zpracování výsledků testování + oprava nalezených chyb	11.05.2015
8. iterace	Příprava k odevzdání Bakalářské práce	18.05.2015

Tabulka 5.1: Plán iterací

5.2 Vývojové prostředí

5.2.1 VirtualBox

Pro vývoj pluginu jsem použila virtualizační nástroj VirtualBox s nainstalovanou OS Ubuntu 14.04. Virtuální stroj jsem zvolila kvůli tomu, že má možnost tvořit snapshoty¹ a pomocí nich se vrátit k uloženému stavu systému. To je velmi užitečná funkce, protože při práci s neznámou aplikací, programovacím jazykem a frameworkem vždy může nastat neočekávaná situace, ze které se bude třeba vrátit do stabilní verze systému.

5.2.2 Git

Abych měla možnost zachovávat svůj progress v kódu, používám vzdálený repositář a decentralizovaný systém pro verzování *git*. Stejně pro udržení stabilní verze projektu používám větvi:

- **master** – je defaultní větev, držím v ni stabilní verzi aplikace.
- **další větev či větví** - pro realizaci nové funkcionality bez rizika pokazit již fungující části aplikace.

Když nová funkčnost bude hotová a otestována, mužů ji přidat k stabilní části pomocí sloučení větev - příkaz `git merge`.

5.2.3 Redmine

Jako systém pro řízení projektu a úkolů(požadavků) používám systém Redmine firmy, ve které pracuji. Tento systém velmi usnadňuje práci s požadavky, obsahuje možnost nastavit prioritu každému požadavku, předpokládaný datum splnění, procento splnění atd. Taky mám možnost přidělovat hotové úkoly na otestování zákazníkovi a dostávat od něho poznámky v podobě komentářů k úkolům.

5.2.4 RubyGems

RubyGems je systém pro správu balíků pro programovací jazyk Ruby, který poskytuje standardní formát pro programy a knihovny Ruby. RubyGems je velmi podobný systémům pro správu balíku jako je *apt-get*, *yum* a *npn*. Lokálně se nazývá gem a jeho balíčky jsou gemy, které v sobě obsahují:

- **Kód**
- **Dokumentaci**
- **Gemspec** – popis balíčků, informace o autorovi

¹Snapshot je snímek stavu systému

Instalace a použití

Poslední verzi RubyGems je možné stáhnout z oficiálních stránek produktu. Aplikace Redmine aktivně používá gem a proto ho také plugin může používat pro stahování, instalaci, mazání a obnovení potřebných balíčků. RubyGems má příkaz `gem`, který může:

- Vyhledávat balíčky – `gem search regulární_výraz`
- Instalovat je – `gem install název_balíčku`
- Odinstalovat – `gem uninstall název_balíčku`
- Vypsát seznam nainstalovaných gemů – `gem list`
- atd.

Pokud nainstalujete balíčky výše uvedeným způsobem, budete je mít jen u sebe lokálně. To znamená, že při nasazení aplikace nebo, v mém případě, pluginu na jiný počítač, bude potřeba znovu instalovat všechny nutné balíčky. Pro takové případy existuje manažer závislosti gemů v ruby aplikacích - *Bundler*. Je to také gem(balíček) a proto se snadno instaluje příkazem:

```
gem install bundler
```

V případě Rails verze 3 je Bundler součástí frameworku, takže ho není potřeba instalovat. Bundler má následující příkazy:

- `bundle init` – vytvoří soubor Gemfile s výchozím zdrojem pro stažení gemů
- `bundle install` – nainstaluje balíčky z souboru Gemfile
- `bundle show` – vypíše seznam nainstalovaných balíčků přes bundle

Aplikace Redmine taky používá *Gemfile* a *bundler* a proto *Gemfile* pluginu nemá obsahovat zdroj pro stažení, *Gemfile* aplikace již má tento zdroj. Definice zdroje pro stažení balíčků může vypadat tak:

```
source 'https://rubygems.org'
```

Soubor Gemfile kromě URL zdroje obsahuje seznam gemů, potřebných pro aplikace, včetně jejich verzi. Každý řádek je jeden gem, čárkami se nerozdělují. Používá se syntaxe RubyGems. Vypadá to tak:

```
gem 'název_balíčku', '>=číslo_verze'
```

Podrobnější specifikace je možno najít na oficiálních stránkách Bundler a RubyGems.

5.2.5 Rake

Rake je samostatný nástroj Ruby, nahrazující unixovou utilitu "Make" a používající soubor *Rakefile* a `.rake` pro vytvoření seznamu úkolů. Název Rake pochází z Ruby Make. Ve frameworku Rails Rake se používá pro běžné administrativní úkoly, zejména ty, které jsou na sobě vzájemně závislé.

Je možné získat seznam dostupných příkazů Rake, které často závisí na aktuálním adresáři. Každý úkol má popis, usnadňující Vaše hledání. Příkaz vypadá takto:

```
rake --tasks
```

Dále vypíšu příkazy, které budu používat nejčastěji:

- `rake about` - poskytuje informace o verzích Ruby, RubyGems, Rails, komponentách Rails, složce aplikace, názvu aktuálního Rails prostředí, adaptéru databáze vaší aplikace a verzi schématu.
- `rake redmine:plugins:migrate` - provede migrace² pluginů do databáze aplikace "Redmine".
- `rake routes` - vypíše všechny routy³ aplikace.

5.2.6 Rails

Tento framework a všechny jeho závislosti je možné stáhnout pomocí RubyGems příkazem:

```
gem install rails
```

Framework Ruby on Rails má následující užitečné příkazy:

- `rails console` - umožňuje interakci s Rails aplikací z příkazového řádku. V podstatě rails console používá IRB. To je užitečné pro testování rychlých nápadů a upravování dat na serveru, aniž by to ovlivnilo webovou stránku. Má zkrácenou variantu: `rails c`
- `rails server` - spustí malý webový server s názvem WEBrick, dodávaný s Ruby. Standardně se používá port 3000, ale je možné ho nastavit ručně pomocí parametru `-p`. Také existuje zkrácená forma příkazu: `rails s`
- `rails generate` - používá šablony na vytváření mnoha věcí. Tento příkaz vypíše seznam dostupných generátorů. Má alias `rails g`

Rails má Redmine plugin generátor, který umí vygenerovat plugin, jeho modely, kontroler atd. Po spuštění příkazu:

```
rails generate redmine_plugin název_pluginu
```

vygeneruje se celá struktura pluginu, podobná struktuře rails aplikaci. V souboru `init.rb` se nachází informace o pluginu : název, jméno autorů, popis atd. A také informace o rozmístění stránky pluginu a další jeho vlastnosti.

²Migrace je to způsob strukturované a uspořádaně měnit databázi

³Routa je relace mezi URL aplikace a spojením – `controller#action`

5.2.7 jQuery

jQuery je knihovna jazyka JavaScript, zaměřuje se na interakci mezi JavaScript a HTML. Tato knihovna pomáhá snadno přistupovat k jakémukoliv elementu DOM, k jeho atributům a obsahu a manipulovat s nimi. jQuery poskytuje pohodlné API pro práci s AJAX.

jQueryUI

jQueryUI je rozmanitá sada interakcí s uživatelským rozhraním, efektů, widgetů⁴ a motivů postavena na knihovně jQuery. jQueryUI již zahrnuje v sobě výše zmíněnou knihovnu jQuery.

Ve svoji práci budu používat knihovnu jQueryUI, protože vystavené požadavky předpokládají interakce na straně klienta. Daná knihovna velice usnadní implementaci a tím ušetří čas.

5.3 Iterace

5.3.1 1.iterace [16.03. - 23.03.2015]

Zadání

Realizovat požadavky číslo 1, 4, 5, 7, 9, 12 tedy:

1. Výpis seznamu uživatelů a projektů, do kterých jsou přiřazeny
2. Zobrazení časové osy, rozdělenou po týdnech
3. Posouvání po časové ose do budoucích a minulých dat
4. Nastavení týdenní kapacity pracovníků
5. Zobrazení trvání kapacity a její hodnoty v podobě pruhu na časové ose
6. Výpis seznamu kapacit uživatele s možností editace a mazání

Analýza a návrh

Pro výpis seznamu uživatelů a projektů, do nichž jsou uživatelé přiřazení, použiji záznamy z tabulky aplikace Redmine „Members“.

Časová osa bude mít počáteční datum a délku. Počáteční datum buď bude bráno z parametrů URL nebo, pokud parametr není, bude použit datum začátku aktuálního týdne. Posouvání po časové ose bude nahrávání stejně stránky, ale s parametrem počátečního data o 2 týdny více nebo méně.

Pro realizaci možnosti přidávat týdenní kapacitu uživatelů (bod č. 4) je potřeba vytvořit novou tabulku „UsersCapacity“ pomoci rails migrace, model pro tuto tabulku a kontroler, ve kterém implementovat akce „new“ a „create“. Taky je zapotřebí vytvořit kontroler pro hlavní

⁴Je to malá aplikace s omezenou funkcí, která může být instalována a provedena v rámci webové stránky koncovým uživatelem

stránku, na které se budou zobrazovat seznamy uživatelů/projektu, časová osa, kapacity, alokace atd. Tento kontrolér bude mít zatím jen jednu akci „index“.

Pruh, zobrazující kapacitu na časové ose, bude HTML blok s atributy počátku a délky. Hodnoty těchto atributů budou vypočtené na základě údajů z tabulky „UsersCapacity“ a následně zpracované pomocí JavaScript pro nastavení správných pozic. Tyto HTML bloky budou nastýlovány pomocí CSS.

Stránka s výpisem všech kapacit uživatele bude obsahovat řádky s informací o kapacitě, tedy hodnota kapacity, datum začátku její platnosti a datum konce, a také ikony pro smazání a editaci. Pro realizaci je potřeba implementovat akce „show“, „edit“, „update“ a „destroy“ kontroleru „UsersCapacityController“. Akce „show“ a „edit“ mají vlastní šablony a akce „update“ a „destroy“ jen provádějí změny v databázi a následně přesměrují na hlavní stránku.

Implementace

Realizace bodů 1, 2, 3, 5, 6 proběhla bez problémů. V průběhu realizaci bodů č. 4 jsem se narazila na to, že kapacity mohou překrývat což jde proti hlavní logice aplikace. Proto jsem musela implementovat dodatečnou kontrolu údajů před zanesením do databáze. Pokud se data budou křížit, údaje nebudou uloženy do databáze a uživatel bude upozorněn hláškou. Také v tomto bodě pro snadné zadávání data jsem použila widget knihovny jQueryUI „datepicker“. Datum se vyplní po kliknutí na příslušný den v interaktivním kalendáři. Abych se zbavila chyb se zadáváním dat, omezila jsem výběr dnů v kalendáři jen na pondělí a také ošetřila případ, že „týden do“ nemůže být méně než „týden od“ a naopak „týden od“ nemůže být více než „týden do“.

K seznamu uživatelů na hlavní stránce (bod č. 1) jsem se rozhodla přidat uživatele, kteří mají aspoň jednu žádanou kapacitu, přestože nemají přiřazené projekty.

Jméno uživatele na stránce „Tým“ je odkazem na seznam všech jeho kapacit (bod č. 6). Tento seznam by byl velmi dlouhý a tím nepřehledný, proto bylo rozhodnuto použít stránkování. Pro tento účel jsem použila gem „will_paginate“. Návod k tomuto balíčku je možno nalézt na stránce jeho repozitáře Github https://github.com/mislav/will_paginate/wiki.

Testování

Realizované požadavky byly otestované mnou ručně a pak také ručně otestované zákazníkem.

Závěr

Splnit požadavky dané iterace se mi podařilo. Zákazník ale upozornil na následující nedostatky:

1. Při přechodu ze stránky „Tým“ do stránky „Projekty“ a zpět by se posunutí na časové ose mělo zachovávat.
2. Formuláře by se po neúspěšném odeslání neměly vymazat. Po úspěšném odeslání formuláře by se mělo dojít k přesměrování zpět na původní stránku.

Připomínky číslo 2 a 3 zatím v úvahu brát nebudou, protože implementace formulářů v modálních oknech předpokládá vyřešení těchto nedostatků.

5.3.2 2.iterace [23.03. - 30.03.2015]

Zadání

Realizovat požadavky číslo 2, 3, 6, 8 tedy:

1. Vypis seznamu projektu a do nich přiřazených uživatelů
2. Přidání alokace zaměstnanců na projektu
3. Zobrazení trvání alokace a její hodnoty v podobě pruhu na časové ose
4. Nastavení týdenní kapacity projektů

Analýza a návrh

Seznam projektů budu vypisovat podobně seznamu uživatelů, použiji záznamy tabulek „Members“ a „ProjectsCapacity“. Kvůli tomu, že budou dvě stránky : „Tým“ a „Projekty“ se objevuje nutnost realizace záložek. Budou to 2 odkazy, stylizované pod záložky, a oba tyto odkazy budou vést na stejnou stránku, ale s jinými parametry URL.

Přidání alokace udělám stejně jako přidávání kapacity. Proto vytvořím migraci pro tabulku „Allocations“, model „Allocation“ a kontroler s akcemi „new“ a „create“.

Alokace a kapacitu projektu budu vykreslovat úplně stejně jako kapacitu uživatele. S tím rozdíl, že bloky pro zobrazení alokace budou mít trochu jiný styl.

Realizace nastavení týdenní kapacity projektu se ničím neliší od realizace nastavení kapacity uživatele. Provedu to velmi podobně.

Implementace

Pro správné zobrazení aktivní záložky jsem použila další gem „active_link_to“. Návod k tomuto balíčku je možné nalézt na stránce jeho repozitáře Github https://github.com/comfy/active_link_to.

Přidávání alokaci jsem realizovala tak, že na stránce „Tým“ tlačítko „Nová alokace“ vede do formuláře, ve kterém je na začátku třeba vybrat uživatele a pak se zobrazí seznam projektů, do kterých je přiřazen. Na stránce „Projekty“ to se chová opačně: nejprve výberu projekt a pak uživatele. Realizovala jsem to pomocí technologie AJAX. Ve formulářích jsem přidala omezení pro hodnotu kapacity nebo alokace. Tato hodnota nemůže být záporná, t.j. méně než 0 a více než 60. Realizace ostatních požadavků proběhla podle plánu.

Testování

Otestovala jsem a funkce byly též otestované zákazníkem.

Závěr

Požadavky byly splněny včas. Zákazník nahlásil jen několik připomínek:

1. Ve formuláři pro přidávání alokace v seznamu uživatelů nebo projektu by se měli vyskytovat jenom ti, kteří mají aspoň jeden přiřazený projekt/uživatele.
2. Odstranit omezení kapacity a alokace na 60.
3. Stejná poznámka k formulářům jako v předchozí iteraci.

5.3.3 3. Iterace [30.03. - 06.04.2015]

Zadání

Realizovat požadavky číslo 10, 11, 13, 14, tedy:

1. Zobrazení rozdílu mezi kapacitou a alokací na pruhu na časové ose
2. Upozornění na překročení kapacity
3. Výpis seznamu kapacit projektu s možností editace a mazání
4. Výpis seznamu alokací s možností editace a mazání

Analýza a návrh

Pro vykreslování rozdílu mezi kapacitou a alokací bych chtěla použít stejné algoritmy jako pro vykreslování kapacity a alokace. Proto je potřeba v hlavním kontroleru připravit data těchto bloků, a konkrétně hodnotu rozdílu, datum začátku a datum konce. Pro každého uživatele nebo projekt se postupně projde seznam kapacit a odečte se od něho alokace.

Upozornění na překročení kapacity bude zvýrazněno barvou a textem bloku, reprezentujícího záporný výsledek.

Výpis seznamu kapacit projektů a alokací s možností editace a mazání bude provedeno stejně jako výpis seznamu kapacit uživatele z 1.iteraci. Název projektu na stránce „Projekty“ bude odkazem na stránku s seznamem jeho kapacit (bod č. 3) a jméno uživatele/projektu v bloku dodatečné informace by vedlo na stránku s výpisem seznamu alokací pro daného uživatele a projekt (bod č. 4).

Závěr

Implementace a testování proběhly bez problémů. Splnit zadání se podařilo, zákazník ale přidal poznámku:

Pokud má uživatel volno, t.j. nulovou kapacitu a nemá v tomto období alokaci, tak rozdíl by měl vypadat stejně jako blok zobrazující nulovou kapacitu.

5.3.4 4. Iterace [06.04. - 13.04.2015]

Zadání

Realizovat požadavky číslo 15, 16, 17, 18, 19 a také opravit některé připomínky z předchozích iterací, tedy:

1. Omezení prav podle role
2. Každému projektu bude přiřazena vlastní barva
3. Možnost posouvat aktuální týden na začátek časové osy
4. Zvýraznění aktuálního týdne na časové ose
5. Možnost ukázat/skrýt dodatečnou informaci
6. Při přechodu ze stránky „Tým“ do stránky „Projekty“ a zpět by mělo zachovávat posunutí na časové ose
7. Ve formuláři pro přidávání alokace v seznamu uživatelů nebo projektů by se měli vyskytovat jenom ti, kteří mají aspoň jeden přiřazený projekt/uživatele

Analýza a návrh

Role pluginu jsou navrženy tak, že uživatel aplikace s vlastností „admin“ má práva na všechny akce, ostatní mají práva jen na zobrazení a prohlížení kapacit, alokaci a jejich rozdílu. V aplikaci Redmine role jsou udělané tak, že každý uživatel na různých projektech může mít různá práva v rámci toho projektu. Plugin má možnost vystavit akce, a pak admin aplikace může upravovat role, které budou mít práva na jejich vykonání. Ale náš plugin nezávisí na projektu, a proto se nám tato varianta nehodí. Řešením je kontrolovat před vykonáním akce, které jsou dostupné jenom administrátorovi, má-li na ní přihlášený nebo anonymní uživatel práva. Jako doplnění je potřeba skrýt přístup k vykonání těchto akcí, t.j. nezobrazovat tlačítka pro přidání, editaci a mazání záznamů.

Pro přidání barev projektům, výběru seznam barev a pak, podle ID projektu přiřadím mu barvu. Bude to realizováno pomocí JavaScript.

Tlačítko pro posunutí aktuálního týdne na začátek časové osy bude nahrávat stránku bez parametrů v URL pro začátek osy, t.j. začátek se nastaví na aktuální týden.

Zvýrazním aktuální týden tak, že najdu blok, reprezentující aktuální týden a přidám mu css třídu.

Rozbalování dodatečné informace bude provedeno pomocí knihovny jQueryUI. Tato knihovna má velmi vhodný efekt „Toggle“. HTML blok, který se bude skrývat a ukazovat se zpět, musí mít atribut `id`. Také je nutné mít něco, co bude spouštět tento efekt, v našem případě to je ikonka šipky, která taky má mít své `id`. Pak, pomocí JavaScriptu, propojím příslušné bloky a ikonky.

Implementace

Při implementaci jsem se nenarazila na problémy. Realizace proběhla podle návrhu.

Závěr

Testování bylo provedené mnou a zákazníkem. Po testování zákazníkem, jsem obdržela následující poznámky:

1. Šipečky u rozbalovacích informací by se měly otáčet směrem dolů
2. Odstranit prompt u rozbalovacího menu ve formuláři pro přidávání alokace

5.3.5 5. Iterace [13.04. - 20.04.2015]

Zadání

Realizovat požadavky číslo 21, 22, 23, 24 a opravit některé připomínky, tedy:

1. Stránky pro editaci a vytvoření v podobě „Modálního“ okna
2. Možnost výchozího nastavení kapacity
3. Možnost přidat alokaci pro určitý projekt (tlačítko „Přidat uživatele“)
4. Možnost přidat alokaci pro určitého uživatele (tlačítko „Přidat do projektu“)
5. Pokud má uživatel volno, t.j. nulovou kapacitu a nemá v to období alokaci, tak rozdíl by měl vypadat stejně jako blok, zobrazující nulovou kapacitu
6. Šipečky u rozbalovacích informací by se měly otáčet směrem dolů

Analýza a návrh

Pro realizace „Modálního“ okna použiju widget knihovny jQueryUI „dialog“. To je HTML objekt, který vypadá jako dialogové okno a má vlastnost „modal“. Po kliknutí na příslušné tlačítko, obsah stránky z odkazu tlačítka se načte do modálního okna. Formuláře budou používat technologie AJAX pro zobrazování chyb.

Rozšířím tabulku „UsersCapacities“ o sloupec `default` typu `boolean` pro realizaci výchozí kapacity. Formulář na vytváření kapacity uživatele se taky obohatí o prvek checkbox pro označení bude-li kapacita výchozí. Rozdíl mezi obyčejnou kapacitou a výchozí je v tom, že výchozí kapacita nemá datum do. Každý uživatel může mít jen jednu výchozí kapacitu, proto při uložení nové kapacity bude provedena kontrola na existence výchozí kapacity u příslušného uživatele a taky kontrola na překrytí jinými kapacitami. Také je potřeba ošetřit případ kdy se do rozmezí výchozí kapacity přidá nová nevýchozí kapacita. V tomto případě existující výchozí kapacita se rozdělí na interval rozdílu mezi začátky výchozí a nové kapacity, interval nové kapacity a nový interval výchozí kapacity.

Tlačítko „Přidat uživatele“ se bude umisťovat v doplňkové informace na stránce „Projekt“ a bude vést na formulář s předvyplněným polem `project_id`. Díky tomu, že projekt již vybrán zůstane zvolit uživatele, hodnotu alokaci a její období.

Funkčnost tlačítka „Přidat do projektu“ bude udělaná podobně. S tím rozdílem, že se rozmísťovat bude na stránce „Tým“ a předvyplněné bude políčko `user_id`.

Implementace

Implementace proběhla bez významných problémů.

Testování

Otestovala jsem několik základních případů přidávání výchozí kapacity a přidávání dalších kapacit do jejích intervalů. Zákazník také ověřil funkčnost.

Závěr

Požadavky dané iterace byly splněné včas a bez významných chyb. Dále následují doplnění k realizovaným funkcím od zákazníka:

1. U formulářů pro přidávání alokace pro určitý projekt nebo uživatele přidat prompt
2. Možnost editace výchozí kapacity
3. Možnost vytvoření výchozí kapacity z již existujících nevýchozích kapacit, pokud ještě není zadaná výchozí
4. Obnovení stránky po opuštění modálu s seznamem kapacit/alokaci
5. Změna označení a obarvení bloku rozdílů na stránce „Projekty“
6. Stránkování změnit na tlačítko „Ukázat další“, které bude vypisovat zbytek seznamu

5.3.6 6. Iterace [20.04. - 04.05.2015]

Zadání

Realizovat požadavky číslo 20, 25 a opravit některé připomínky, tedy:

1. Posunutí na určitý datum nebo období na časové ose
2. Přizpůsobit množství týdnů na časové ose šířce prohlížeče
3. Přeložit celý plugin
4. U formulářů pro přidávání alokaci pro určitý projekt nebo uživatele přidat prompt
5. Možnost editace výchozí kapacity
6. Možnost vytvoření výchozí kapacity z již existujících nevýchozích kapacit, pokud ještě není zadaná výchozí
7. Obnovení stránky po opuštění modálu s seznamem kapacit/alokací
8. Změna označení a obarvení bloku rozdílů na stránce „Projekty“
9. Stránkování změnit na tlačítko „Ukázat další“, které bude vypisovat zbytek seznamu

Analýza a návrh

Požadavek číslo 20, tedy posunutí na určité období či datum, bude realizován takto: na hlavní stránce přibude rozbalovací seznam, ve kterém budou možnosti: „minulý měsíc“, „tento měsíc“ a „určitý datum“. Po zvolení položky „určitý datum“ se objeví pole, ve kterém je možné žádat datum ručně nebo pomocí widget „datepicker“, hned po zadání data se daná stránka obnoví s novou hodnotou parametrů počátečního data. Ostatní položky také obnoví stránku s novým parametrem, hodnota tohoto parametre je určena současným měsícem, t.j. položka „tento měsíc“ se posune časovou osu na týden počátku současného měsíce, a položka „minulý měsíc“ na první týden minulého měsíce.

Přizpůsobení množství týdnů k šířce prohlížeče znamená, že na základě šířky okna prohlížeče bude vypočten maximální počet týdnů, t.j. sloupců. Hlavním problémem je, že vykreslení časové osy se děje na serverové straně a zjistit šířku je možné jen na klientské. Proto řešením je na začátku vykreslení stránky dozvědět její šířku, vypočítat vhodné množství týdnů a poslat AJAX požadavek na server pro šablonu časové osy s novým počtem sloupců a následně ji vykreslit na stránce.

Pro přeložení aplikace jsem použila knihovnu I18n, která jde spolu s frameworkem Ruby on Rails od verzi 2.2. Tato knihovna se zabývá překladem a lokalizací. Používá se lokalizační soubor s překlady ve složce `config/locales`, který má příponu `.yml`.

Z výchozí kapacity bude možnost vytvořit obyčejnou po odškrtnutí vlastnosti „Výchozí kapacita“ a následném zadání „týdnu Do“.

Stejně tak z obyčejné kapacity bude možnost vytvořit výchozí. Tohle se bude dovolovat jen v případě kdy uživatel zatím nemá jinou výchozí kapacitu.

Body č. 4 a 7 nevyžadují popis.

Změna označení a obarvení bloku rozdílu na stránce „Projekty“ znamená, že oproti zobrazení na stránce „Tým“, blok s kladnou hodnotou, označující nedostatek alokací, bude označen červenou barvou a blok se zápornou, označující naopak přebytek alokací, bude zelený. Protože nedostatek lidí na projektu je z pohledu vedení projektu horší situace než přebytek. Na stránce „Tým“ červený blok ukazuje na vytíženost přes jeho kapacitu.

Knihovna pro snadné stránkování „will_paginate“ nepodporovala zobrazení v modálním oknu a proto zákazník nabídnul následující řešení: vypisovat jen 3 prvních záznamů a přidat tlačítko „Ukázat další“, po kliknutí na které zobrazí se zbytek seznamu. Bude to realizované pomocí AJAX požadavků na server pro vykreslení celého seznamu.

Závěr

Implementace proběhla bez závažných problémů. Otestováno bylo ručně. Touto iterací se skončila implementační část. Ostatní požadavky a to jsou:

1. Drag&Drop pruhu na časové ose
2. Možnost roztažení pruhu na časové ose
3. Omezení výběru dat v kalendáři při editaci/vytvoření kapacity/alokace

budou realizované později po dohodě se zákazníkem

Kapitola 6

Testování

Vzhledem k rozsáhlejšímu obsahu je 7. iterace popsána v této samostatné kapitole. Dále bude popsána příprava, průběh a zpracování výsledků testování použitelnosti aplikaci s uživateli.

6.1 Test s uživateli

6.1.1 Cíle testu

Cílem tohoto testování je zjistit postupy potenciálních uživatelů aplikace při plnění specifikovaných úkolů. Takové testování umožní odhalit nedostatky a problémy uživatelského rozhraní a také ověřit funkčnost aplikace, t.j. jestli operace projdou bez chyb.

6.1.2 Požadovaná skupina participantů

Cílovou skupinu uživatelů tvoří lidé, běžné používající systém Redmine jako systém pro řízení projektu. Z toho vyplývá, že jsou průměrní či pokročilejší uživatelé počítačů. Někteří z nich by měli být projektoví manažeři. Požadovaný počet participantů je pět.

6.1.3 Screener

Je to dotazník, sloužící k vybraní participantů. Obsahuje veřejnou a neveřejnou část. Veřejná část se skládá z otázek a určena pro potenciální participanty. Neveřejná je ukazatelem ideálního rozvržení odpovědi. Obě částí najdete v přílohách [B.1](#)

6.1.4 Pre a Post-test dotazníky

Pre-test dotazník se používá pro sběr doplňujících informací o již vybraném kandidátu na testování. Tato informace pomůže v analýze jeho chování při testu.

Po dokončení testování participant projde post-test dotazníkem, který má za účel se dozvědět dojem účastníka z aplikace, t.j. jak se cítil během používání aplikaci, co mu dělalo potíže a co se mu v aplikaci líbilo. Oba dotazníky jsou v přílohách [B.2](#) a [B.3](#)

6.1.5 Nastavení testu

Testování použitelnosti se provádí v usability laboratoři na Karlově náměstí. Laboratoř se skládá ze dvou oddělených místností. Sledovaná místnost je osazena dvěma kamerami, mikrofonem a jedním počítačem, na kterém participant testuje software. Kromě participantu sedí v této místnosti také moderátor, který má za úkol kontrolovat průběh testu a pomáhat participantovi v případě problému. V druhé místnosti sedí skupina testerů, kteří sledují účastníka testu a logují vše podstatné, co se během testu stane. Zároveň se zde pořizují audiovizuální záznamy z kamer a záznam ze vzdálené plochy participanta.

V daném případě testování budu v roli moderátora a můj asistent bude mi pomáhat sbírat záznamy z kamer.

Hardwarové prostředky

- Typ PC: Stolní
- Výkon: Intel Core 2 4300 1.8GHz, 2 GB RAM
- Displej: Monitor Dell, rozlišení 1280x960
- Příslušenství: klávesnice, myš, reproduktory, mikrofon
- Další vybavení místnosti: 2 kamery

Softwarové prostředky

- OS: Windows 7
- Morae - program určený ke sledování participantu a logování jejich činností během testu
- VirtualBox verze 4.3.*

6.1.6 Seznam úkolů

V následujících úkolech jsem se snažila zahrnout všechny důležitější funkce aplikace.

1. Vytvořit kapacitu uživatele (údaje žádejte libovolné)
2. Editovat tuto kapacitu (údaje žádejte libovolné)
3. Vytvořit výchozí kapacitu uživatele (údaje žádejte libovolné)
4. Změnit výchozí kapacitu na běžnou kapacitu
5. Vytvořit kapacitu projektu
6. Přidat alokaci pro tento projekt
7. Editovat dříve vytvořenou alokaci

8. Posunout časovou osu na začátek minulého měsíce
9. Vrátit začátek časové osy na aktuální týden

6.1.7 Průběh testu s uživateli

Popis skupiny

Naši testovací skupinu tvořilo 5 uživatelů s pokročilejší zkušeností s prací s počítačem. Všichni jsou muži v věkovém rozmezí 20 až 40 let, čtyři z nich mají zkušenost s systémem Redmine rok nebo více, tři se zabývají projekt managementem a dva k tomu využívají speciální software.

Pre-test dotazník ukázal, že všichni rozumějí pojmu „Řízení lidských zdrojů“ a že očekávají od aplikace pro řízení lidských zdrojů takové vlastnosti jako: „plánování dostupných kapacit a dovolené“, „seznam projektů a lidí, pracujících na nich“, „přehled pracovních kapacit“, „preferované dny práce“, „označení druhu práce, kterou dělá na daném projektu zaměstnanec“ atd. Tato informace je užitečná pro další vývoj aplikace.

Průběh testování

První úkol nebyl pro participanty těžký, ale téměř všechny zmátlo označení týdne od a do. Nebylo pochopitelné to, že pro označení týdne se používá pondělí a že týdny se sčítají včetně posledního. Jeden participant dlouho hledal tlačítko pro přidávání kapacity, chyběla mu ikonka „+“. Ve druhém úkolu se všichni snažili najít editaci přímo na pruhu zobrazujícím rozdíl mezi kapacitou a alokací. Pak nebyl velký problém najít seznam všech kapacit a příslušnou ikonku pro editaci.

Úkoly 3, 4, 5 a 8 proběhly bez potíží.

Všichni participanté zvládli 6 úkol, ale jiným způsobem, než se očekávalo. Pravděpodobně je tento problém spojen s tím, že tlačítko „Přidat uživatele“ je orientované na pokročilejšího uživatele aplikace.

Sedmý úkol předpokládal, že uživatel přijde na to jak se zobrazuje dodatečná informace a dále najde odkaz vedoucí na seznam všech alokací s možností editace. Zásadním problémem bylo, že participanté zaprvé nevěděli o existenci rozbalovací sekce a zadruhé ikonka pro otevření/zavření dodatečné informace nevypadala jako aktivní prvek, t.j. ten který vyvolává akci. Všichni potřebovali nápovědu, ale pak to moc dlouho netrvalo. Po zkušenosti s editací v předchozím úkolu, tento úkol byl zvládnut rychle.

Devátý úkol uživatelé zvládali, ale jen 2 ho provedli nejkratší a správnou cestou. Ostatní to vysvětlovali tím, že odkaz „Dnes“ nevypadá jako odkaz a nachází se daleko od navigace, která je ihned viditelná.

6.1.8 Vyhodnocení testování

Do výsledného vyhodnocení zahrnu odpovědi na post-test dotazník a průběh samotného testování. Dále následuje tabulka nálezů, seřazených podle priorit a také seznam doporučených vlastností pro rozšíření aplikace. Vysvětlení priorit:

- **Priorita 3** - Závažný problém, který silně ovlivňuje používání aplikace.

- **Priorita 2** - Nález, který znepříjemňuje používání aplikace.
- **Priorita 1** - Drobný problém, spíše kosmetický.

1. Výběr dat

Priorita: [3] Nalezeno u participantů číslo [1, 2, 4, 5]

Popis

Označení trvání kapacit a alokací není pro uživatele očividné. V kalendáři je možné vybrat jen pondělí jako „Týden od“ a také „Týden do“. Kvůli tomu uživatele nerozumí, kolik týdnů bude trvat kapacita nebo alokace.

Možné řešení

Jako „Týden do“ vybírat jen neděli nebo v kalendáři zvýrazňovat celý týden.

2. Ikonka pro zobrazení dodatečné informace

Priorita: [2] Nalezeno u participantů číslo [1, 2, 3, 4, 5]

Popis

Ikonka „>“ vpravo od jména Uživatele/názvu Projektu pro participanty nebyla viditelná. Všichni ji vnímali jako grafický prvek.

Možné řešení

Zvýraznit ikonku a změnit kurzor na ikonku pointer.

3. Grafické ikonky do tlačítek

Priorita: [1] Nalezeno u participantů číslo [3, 4]

Popis

Pro některé uživatele význam tlačítek nebyl zřejmý a tlačítka nevypadaly jako tlačítka. Proto participanté dlouho hledali tlačítka pro přidávání kapacity/alokace.

Možné řešení

Přidat k názvu tlačítka ikonku „+“ a měnit vzhled tlačítka na hover nebo focus.

4. Neviditelnost navigace na časové ose

Priorita: [1] Nalezeno u participantů číslo [1, 2, 4]

Popis

Většina participantů si nevšimla navigace vlevo pro časovou osu, která se skládá z dvou ikon pro krokový posun časové osy a odkaz „Dnes“ pro návrat na aktuální týden. Vysvětlovali to tím, že jiná navigace pro posun na aktuální/minulý měsíc a určitý datum se nachází na protější straně.

Možné řešení

Přesunout navigaci vlevo od časové osy k druhé navigaci v pravém horním rohu.

Participantí ohodnotili intuitivnost uživatelského rozhraní v průměru na 4 z 5 a grafické zpracování na 4,6 z 5. Dále následuje seznam návrhů od participantů pro rozšíření aplikace:

1. Editace a mazání kapacity/alokace po kliknutí na odpovídající pruh.
2. Zobrazení role uživatele na projektu
3. Zachovávat stav „rozbalené“ u dodatečně informace
4. Přidat odkaz na stránku s informacemi o uživateli
5. Přidat tlačítko pro vytváření nové kapacity/alokace vedle seznamu kapacit/alokací
6. V seznamu kapacit/alokací ukazovat jen aktuální, t.j. ode dnes a novější

Kapitola 7

Závěr

Cílem této práce bylo realizovat plnohodnotný nástroj pro pomoc projektovým manažerům. Větší část pozornosti byla zaměřena k návrhu aplikace, která by nebyla úplným opakováním již existujících, ale měla by užitečné funkce navíc a splnila by zásadní očekávání zadavatele. Z tohoto důvodů byla pečlivě provedena rešerše stávajících řešení, byly zdůrazněny jejich základní klady a nedostatky a na základě toho bylo řešením vytvořit plugin, který by realizoval již částečně existující funkčnost, ale jeho hlavní předností bude přímá komunikace s systémem Redmine a možnost zpracování jeho dat. Rešerši následoval prvotní návrh GUI pluginu, který byl vylepšen po připomínkách zákazníka. Dále vznikl přesný seznam požadavků rozdělený podle priorit. Pro vývoj pluginu byl zvolen iterativní přístup a proto se požadavky implementovaly postupně a na konci každé iterace byly otestované a zkontrolované zadavatelem, který je potenciálním uživatelem. Poté bylo provedeno testování s uživateli, které objevilo nedostatky spojené s používáním aplikace. Tyto problémy byly vzaty v úvahu a opraveny.

Plugin byl napsán v jazyce Ruby pomocí frameworku Ruby on Rails a s využitím webových technologií AJAX a JavaScript. Během implementace jsem se snažila řídit se konvencemi a standardy frameworku a vyhýbat se opakování kódu.

Výstupem je plugin, hotový k použití v praxi. Výsledná aplikace umožňuje sledovat vytíženost pracovníků a nedostatek nebo přebytek lidských zdrojů na projektech po týdnech v hodinách. Funguje to tak, že se žádá týdenní pracovní kapacita uživatele a také týdenní kapacita projektu, t.j. počet hodin které projekt potřebuje pro splnění, pak je možné uvést alokaci zaměstnance na projektu a také dobu, pro kterou tato alokace platí. Poté se zobrazí rozdíl mezi hodnotami kapacity a alokací. Toto je základní myšlenka aplikace, plugin umožňuje zobrazovat grafické kapacity, alokace a jejich rozdíl na časové ose, grafický upozorňovat na překročení kapacity a nedostatek lidských zdrojů, pohodlně se pohybovat po časové ose a má 2 pohledy: podle lidí a podle projektů.

Domnívám se, že vykonaná práce splňuje zadaný úkol a základní požadavky na aplikaci. Plugin se bude dále rozvíjet pro komfortní použití a také může být přeložen do dalších jazyků.

Literatura

- [1] Plugin Tutorial. *Redmine.org* [online]. [cit. 2015-05-06]. Dostupné z:
http://www.redmine.org/projects/redmine/wiki/Plugin_Tutorial
- [2] BUNDLER. *Oficiální stránky* [online]. [cit. 2015-05-06]. Dostupné z:
<http://bundler.io>
- [3] RUBYGEMS. *Oficiální stránky* [online]. [cit. 2015-05-06]. Dostupné z:
<https://rubygems.org>
- [4] Softwarové inženýrství. *Dudka.cz* [online]. [cit. 2015-05-06]. Dostupné z:
<http://dudka.cz/studyIUS>
- [5] Database Model. *Redmine.org* [online]. [cit. 2015-05-06]. Dostupné z:
<http://www.redmine.org/projects/redmine/wiki/DatabaseModel>
- [6] Relační databáze. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2015-05-06]. Dostupné z:
http://cs.wikipedia.org/wiki/Relační%C3%AD_databáze
- [7] RESOURCE GURU. *Oficiální stránky* [online]. [cit. 2015-05-06]. Dostupné z:
<http://resourceguruapp.com>
- [8] HARVEST FORECAST. *Oficiální stránky* [online]. [cit. 2015-05-06]. Dostupné z:
<https://www.getharvest.com/forecast>
- [9] RAILS GUIDES. *Tutorialy z Rails* [online]. [cit. 2015-01-17]. Dostupné z:
<http://guides.rubyonrails.org/>
- [10] REDMINE. *Oficiální stránky* [online]. [cit. 2015-01-12]. Dostupné z:
<http://www.redmine.org>
- [11] EASY REDMINE. *Oficiální stránky* [online]. [cit. 2015-01-12]. Dostupné z:
<http://www.easyredmine.com/en>
- [12] Ruby on Rails directory structure. TUTORIALS POINT. *Tutorials Point* [online]. [cit. 2015-01-17]. Dostupné z:
<http://www.tutorialspoint.com/ruby-on-rails/rails-directory-structure.htm>
- [13] Active Record Basics. *Rails Guides* [online]. 2010 [cit. 2015-05-06]. Dostupné z:
http://guides.rubyonrails.org/active_record_basics.html

- [14] Ruby on Rails. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2015-01-17]. Dostupné z: https://cs.wikipedia.org/wiki/Ruby_on_Rails

Příloha A

Seznam použitých zkratek

- GPL** General Public License
- REST** Representational State Transfer
- API** Application Programming Interface
- PHP** Personal Home Page, skriptovací programovací jazyk
- HTML** HyperText Markup Language
- CSS** Cascading Style Sheets
- RoR** framework Ruby on Rails
- XML** Extensible Markup Language
- JSON** JavaScript Object Notation
- GUI** Graphical User Interface
- MIT** Massachusetts Institute of Technology
- OS** Operating systém
- PC** Personal Computer
- IRB** Interactive Ruby Shell
- ERB** Embedded Ruby
- URL** Uniform Resource Locator
- ORM** Object-relational Mapping
- DOM** Document Object Model
- AJAX** Asynchronous JavaScript and XML

Příloha B

Dotazníky

B.1 Screener

Otázky screeneru (veřejná část) a počet požadovaných odpovědi (neveřejná část) :

1. Vaše pohlaví?
 - (a) Muž (4)
 - (b) Žena (1)

2. Váš vek?
 - (a) Do 20 (1)
 - (b) 20-30 (1)
 - (c) 30-40 (2)
 - (d) 40-50 (1)

3. Mate zkušenost s systémem Redmine?
 - (a) Vůbec (1)
 - (b) Méně než rok (1)
 - (c) Více než rok (3)

4. Zabýváte se projekt managementem?
 - (a) Ano (4)
 - (b) Ne (1)

5. Pokud ano, využíváte k tomu specializovaný software?
 - (a) Ano (3)
 - (b) Ne (2)

B.2 Pre-test dotazník

1. Pokud používáte specializovaný nástroj pro řízení projektů a lidských zdrojů, napište jaký.
2. Rozumíte pojmu „Řízení lidských zdrojů“?
3. Co by podle Vás měla aplikace pro řízení lidských zdrojů dělat?

B.3 Post-test dotazník

1. S jakým největším problémem jste se setkal při používání aplikace?
2. Bylo uživatelské rozhraní intuitivní? Označte 1 (vůbec ne) - 5 (rozhodně ano)
3. Jak se Vám líbilo grafické zpracování aplikace? Označte 1 (vůbec nelíbilo)- 5 (velmi líbilo)
4. Jaké vlastnosti podle Vás chybí aplikaci? (nápadů pro rozšíření)

Příloha C

Instalační a uživatelská příručka

C.1 Instalace

1. Zkopírujte adresář pluginu do složky `#{RAILS_ROOT}/plugins` v případě Redmine 2.x nebo `#{RAILS_ROOT}/vendor/plugins` v případě Redmine 1.x
2. Proveďte stazeni závislosti příkazem `bundler install`
3. Proveďte migrace databazi pomocí příkazu
Pro Redmine 1.x
`rake db:migrate_plugins RAILS_ENV=production`
pro Redmine 2.x
`rake redmine:plugins:migrate RAILS_ENV=production`
4. Restartujte Redmine

Poté by měl plugin být v seznamu pluginů v Administrace > Doplnky (`admin/plugins`)

C.2 Uživatelská příručka

V této uživatelské příručce krátce sepisujeme hlavní funkce pluginu a přivedeme screenshoty. Popisu akce na screenshotech shora dolů:

Screenshot [C.1](#) – stránka „Tým“

- Tlačítko „Nova kapacita“ – slouží pro přidání týdenní kapacity uživatele
- Tlačítko „Nova alokace“ – pro přidání alokace uživatele na projektu
- Rozbalovací menu „Posunout se na“ – začátek časové osy se posune do vybraného období
- konka „«“ – posouvá časovou osu o 2 týdny zpět
- Odkaz „Dnes“ – vrátí začátek časové osy na aktuální týden

« Dneska »	04 kvě - 10 kvě 2015	11 kvě - 17 kvě 2015	18 kvě - 24 kvě 2015	25 kvě - 31 kvě 2015	01 čvn - 07 čvn 2015	08 čvn - 14 čvn 2015	15 čvn - 21 čvn 2015
Redmine Admin >	volné 20 hod.	přes 58 hod.					
Antonina Lebedeva >	volné 55 hod.						
Tomáš Novák ▾		volné 20 hod.			přes 5 hod.		volné 25 hod.
Celková kapacita	40 hod.				10 hod.		40 hod.
Přidat do projektu							
Test_project_1	20 hod.						
Second_project					15 hod.		
Pavel Paraska >					volné 40 hod.		
Stanislav Kocanda >	volné 90 hod.	Volno			volné 90 hod.		

Obrázek C.1: Screenshot aplikace - stránka Tým

- Ikonka „»“ - posouvá časovou osu o 2 týdny vpřed
- Jmeno uzivatele – odkaz, vede na seznam všech jeho kapacit [C.4](#)
- Ikonka „>“ - slouží pro otevření/zavření dodatečných informací o alokaci
- Tlačítko „Přidat do projektu“ – pro vytvoření alokaci pro daného uživatele
- Název projektu - odkaz, vede na seznam všech alokace dotyčného uživatele na tomto projektu

Screenshot [C.2](#) – stránka „Projekty“

- Tlačítko „Nova kapacita“ – slouží pro přidání týdenní kapacity projektu
- Tlačítko „Nova alokace“ – pro přidání alokace uživatele na projektu
- Rozbalovací menu „Posunut se na“ – začátek časové osy se posune do vybraného období
- Ikonka „«“ – posouvá časovou osu o 2 týdny zpět
- Odkaz „Dnes“ – vrátí začátek časové osy na aktuální týden
- Ikonka „»“ – posouvá časovou osu o 2 týdny vpřed
- Název projektu – odkaz, vede na seznam všech kapacit daného projektu
- Ikonka „>“ – slouží pro otevření/zavření dodatečných informací o alokaci
- Tlačítko „Přidat do projektu“ – pro vytvoření alokaci pro daného uživatele
- Jmeno uzivatele - odkaz, vede na seznam všech alokace dotyčného uživatele na tomto projektu

Tým		Projekty						
Nová kapacita		Nová alokace		Posunut se na Vyberte období				
« Dneska »		04 kvě - 10 kvě 2015	11 kvě - 17 kvě 2015	18 kvě - 24 kvě 2015	25 kvě - 31 kvě 2015	01 čvn - 07 čvn 2015	08 čvn - 14 čvn 2015	15 čvn - 21 čvn 2015
Test_project_1		přes 20 hod.	chybí 20 hod.	přes 10 hod.	chybí 25 hod.	chybí 100 hod.		
Celková kapacita		60 hod.		100 hod.				
Přidat uživatele								
Antonina Lebedeva		35 hod.						
Tomáš Novák		20 hod.						
Redmine Admin		25 hod.		55 hod.				
Second_project		chybí 108 hod.	chybí 10 hod.	chybí 22 hod.	chybí 125 hod.	chybí 145 hod.		
Celková kapacita		120 hod.			160 hod.			
Přidat uživatele								
Antonina Lebedeva		12 hod.						
Pavel Paraska		20 hod.						
Redmine Admin		78 hod.						
Stanislav Kocanda								

Obrázek C.2: Screenshot aplikace - stránka Tým

Screenshot C.3 – stránka „Tým“, formulář pro přidávání kapacity

Obrázek C.3: Screenshot aplikace - stránka Tým, formulář pro přidávání kapacity

Screenshot C.4 – stránka „Tým“, seznam kapacit

The screenshot shows the Redmine interface. At the top, there is a navigation bar with 'Uvodni', 'Moje stranka', 'Projekty', 'Resource Manager', 'Administrace', and 'Napoveda'. On the right, it says 'Přihlášen jako admin', 'Můj účet', and 'Odhlášení'. Below the navigation bar, there is a search bar and a dropdown menu for 'Vyberte projekt...'. The main content area is titled 'Tým' and 'Projekty'. There are buttons for 'Nová kapacita' and 'Nová alokace'. A modal window titled 'Seznam kapacit uživatele Tomáš Novák' is open, showing a table with columns: 'Kapacita', 'Týden Od', 'Týden Do', 'Smazat', and 'Editovat'. The table contains three rows of capacity data for Tomáš Novák. Below the modal, there is a 'Ukázat další' button. The main table below the modal shows capacity for various team members across different weeks.

Kapacita	Týden Od	Týden Do	Smazat	Editovat
40 hodin	2015-06-15	(nekonečno)		
10 hodin	2015-06-01	2015-06-08		
40 hodin	2015-05-04	2015-05-25		

Uživatel	04 kvě - 10 kvě 2015	11 kvě - 17 kvě 2015	18 kvě - 24 kvě 2015	25 kvě - 31 kvě 2015	01 čvn - 07 čvn 2015	08 čvn - 14 čvn 2015	15 čvn - 21 čvn 2015
Redmine Admin	volné 20 hod.						
Antonina Lebedeva	volné 43 hod.						
Tomáš Novák	volné 20 hod.	volné 20 hod.	volné 20 hod.	přes 5 hod.	volné 25 hod.		
Pavel Paraska				volné 20 hod.	volné 40 hod.		
Stanislav Kocanda	volné 90 hod.	Volno		volné 90 hod.			

Obrázek C.4: Screenshot aplikace - stránka Tým, seznam kapacit

- Ikonka „smazat“ – smaže daný záznam
- Ikonka „editovat“ – vede do formuláře pro editace záznamu

Příloha D

Obsah přiloženého CD

Přiložene CD obsahuje všechny složky a soubory, které jsou součástí této práce.

- `\plugin` - složka se zdrojovými kódy aplikace.
- `\text` - složka obsahující dokumentaci ve formátu PDF a LATEX.
- `readme.txt` - soubor obsahující pokyny k instalaci.