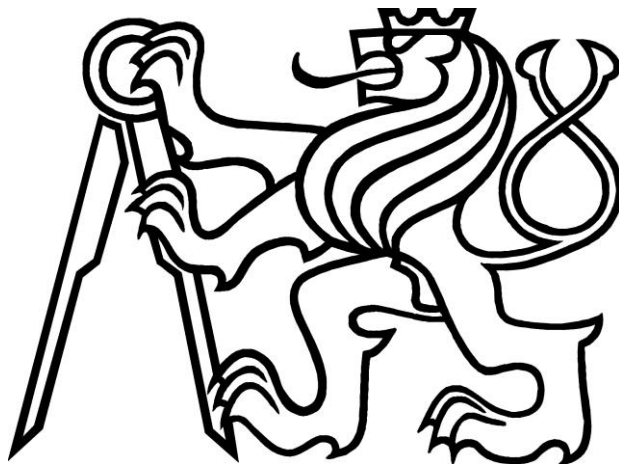


České vysoké učení technické
Fakulta elektrotechnická
Katedra počítačů



Bakalářská práce

System pro management času projektového týmu

Pavel Gruda

Vedoucí práce: doc. Ing. Zdeněk Kouba, CSc.

Studijní program: Softwarové technologie a management, Bakalářský

Obor: Softwarové inženýrství

Květen 2014

České vysoké učení technické v Praze
Fakulta elektrotechnická

katedra počítačů

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: **Pavel Gruda**

Studijní program: Softwarové technologie a management
Obor: Softwarové inženýrství

Název tématu: **Systém pro management času projektového týmu**

Pokyny pro vypracování:

Seznamte se s existujícími systémy pro podporu řízení práce týmu řešícího více projektů současně.

Vytvořte seznam požadavků na systém pro podporu řízení práce s důrazem na přehledné přiřazování úkolů pracovníkům, oddělené vykazování výkonů na jednotlivých projektech a tvorbu přehledných reportů.

Systém navrhnete jako webovou aplikaci a implementujete ve vhodném frameworku.

Navrhnete sadu testových případů (test cases) k ověření správné funkčnosti implementovaného systému.

Přílohou bakalářské práce bude protokol o provedených testech.

Seznam odborné literatury:


Dodá vedoucí práce

Vedoucí: Doc. Ing. Zdeněk Kouba, CSc.

Platnost zadání: do konce zimního semestru 2014/2015


doc. Ing. Filip Železný, Ph.D.
vedoucí katedry




prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 25. 9. 2013

Poděkování

Na tomto místě bych rád poděkoval vedoucímu mé bakalářské práce doc. Ing. Zdeňku Koubovi, CSc. za odbornou pomoc, cenné rady a čas strávený konzultacemi, kterými přispěl k vytvoření bakalářské práce.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne 23. 5. 2014.

Podpis

Abstrakt

Práce se zabývá systémem pro management času projektového týmu. V první části analyzuje problematiku, v dalších částech probíhá implementace danou technologií a nakonec je vytvořená aplikace testována.

Abstract

The thesis deals with the time management system of project team. First of all it analyzes the issues. In the next step it implements functionality by chosen technology and at the end the created application is tested.

Obsah

1	Analytická dokumentace.....	1
1.1	Představení problematiky.....	1
1.2	Cíl bakalářské práce	1
1.3	Funkční a obecné požadavky	1
	Funkční požadavky	1
	Obecné požadavky.....	2
1.4	Analýza rizik	2
1.4.1	Riziko R01	2
1.4.2	Riziko R02	3
1.4.3	Riziko R03	3
1.4.4	Riziko R04	4
1.4.5	Riziko R05	4
1.4.6	Riziko R06.....	4
1.5	Konceptuální datový model s datovými typy.....	5
1.5.1	Popis konceptuálního datového modelu	5
1.5.2	Rozepsání entit konceptuálního datového modelu	6
1.5.3	Vztahy vystupující v konceptuálním datového modelu.....	10
1.6	Aktéři systému	11
1.7	Definice uživatelských rolí	12
1.8	Případy užití	13
	Diagram případů užití nepřihlášeného uživatele	13
	Diagram případů užití přihlášeného uživatele	14
	Diagram případů užití člena projektu	15
	Diagram případů užití vlastníka projektu	16
1.8.1	Rozepsané případy užití.....	17
1.8.2	Další případy užití.....	25
1.8.3	Oprávnění uživatelů systému.....	26
2	Hlavní použité nástroje a technologie.....	28
2.1	Použité vývojové nástroje	28
	IntelliJ Idea	28
	Notepad++	28
	Astash	28

2.2	Použité technologie.....	28
	Grails	28
	Twitter bootsrap	28
	Apache Tomcat	29
	XAMPP	29
	phpMyAdmin	29
	Apache.....	29
3	Implementace	30
3.1	Popis použitého frameworku Grails	30
	Hlavní technologie používané frameworkem Grails.....	30
3.2	Popis struktury aplikace.....	31
3.2.1	Kořenový adresář aplikace	31
3.2.2	Konfigurační soubory aplikace	31
3.2.3	Domain	33
3.2.4	Controllers.....	33
3.2.5	Views.....	33
3.2.6	Services	33
3.2.7	i18n.....	33
3.3	Seznam kontrolerů aplikace.....	34
	Vypsání seznam kontrolerů aplikace.....	34
3.4	Seznam views (webových stránek aplikace)	35
	Vypsání seznam views aplikace a jejich popis.....	35
3.5	Uživatelské rozhraní aplikace.....	36
3.6	Ukázka uživatelského rozhraní aplikace	37
3.7	Diagram nasazení	39
4	Testování	40
4.1	Využití souboru bootsrap při testování.....	40
4.2	Testování uživatelského rozhraní v různých prohlížečích	40
	Seznam testovaných vlastností GUI.....	40
	Výstup testu uživatelského rozhraní	41
4.3	Testování programem Selenium.....	41
4.4	Testování aplikace podle případů užití.....	43
5	Závěr.....	45
6	Seznam použité literatury	46

Seznam obrázků

Obrázek 1 - Konceptuální datový model s datovými typy	5
Obrázek 2 - Uživatelské role systému	11
Obrázek 3 - Diagram případů užití nepřihlášeného uživatele.....	13
Obrázek 4 - Diagram případů užití přihlášeného uživatele	14
Obrázek 5 - Diagram případů užití člena projektu	15
Obrázek 6 - Diagram případů užití vlastníka projektu	16
Obrázek 7 - Orientační obrázek hlavních technologií ve frameworku Grails	30
Obrázek 8 - GUI - klik na tlačítko přidání zákazníka.....	37
Obrázek 9 - GUI - zobrazení formuláře "Přidání zákazníka"	37
Obrázek 10 - GUI - upozornění na nesprávně vyplněné údaje.....	38
Obrázek 11 - GUI - správně vyplněné údaje	38
Obrázek 12 - GUI - Zobrazení detailu vytvořeného zákazníka a vypsaná hláška o vytvoření	39
Obrázek 13 - Diagram nasazení.....	39
Obrázek 14 - Výstup Selenium testů	42

Seznam tabulek

Tabulka 1 - Riziko R01.....	2
Tabulka 2 - Riziko R02.....	3
Tabulka 3 - Riziko R03.....	3
Tabulka 4 - Riziko R04.....	4
Tabulka 5 - Riziko R05.....	4
Tabulka 6 - Riziko R06.....	4
Tabulka 7 - Atributy entity User (Uživatel).....	6
Tabulka 8 - Atributy entity Customer (Zákazník)	7
Tabulka 9 - Atributy entity Project Projekt (Projekt)	7
Tabulka 10 - Atribut entity WorkOnProject (Pracuje na projektu)	8
Tabulka 11 - Atributy entity Ticket (Úkol)	8
Tabulka 12 - Stav entity Ticket (Úkol).....	9
Tabulka 13 - Atributy entity WorkedTime (Odpracovaný čas).....	9
Tabulka 14 - Atributy entity Milestone	10
Tabulka 15 - Definice uživatelských rolí.....	12
Tabulka 16 - PU1 - Registrace nového uživatele a přihlášení	17
Tabulka 17 - PU2 - Přihlášení.....	18
Tabulka 18 - PU3 - Odhlášení	18
Tabulka 19 - PU4 - Prohlížení seznamu svých a zúčastněných projektů	18
Tabulka 20 - PU5 - Vytvoření svého zákazníka	19
Tabulka 21 - PU6 - Editace svého zákazníka	20
Tabulka 22 - PU7 - Smazání zákazníka.....	20
Tabulka 23 - PU8 - Založení svého projektu	21
Tabulka 24 - PU9 - Přidání registrovaného uživatele do projektu	22
Tabulka 25 - PU10 - Výpis odpracovaných hodin účastníka projektu v časovém období	23
Tabulka 26 - PU11 - Vykázání práce na svém ticketu	24
Tabulka 27 - Další případy užití	26
Tabulka 28 - Oprávnění uživatelů v systému	27
Tabulka 29 - Seznam kontrolerů aplikace	34
Tabulka 30 - Vypsání seznamu views aplikace a jejich popis.....	36
Tabulka 31 - Výstup testu uživatelského rozhraní vůči různým prohlížečům	41

Tabulka 32 - Výstup testování aplikace podle případů užití.....	44
---	----

1 Analytická dokumentace

1.1 Představení problematiky

Již třetím rokem pracuji v Centru informatiky na Fakultě tělesné výchovy a sportu Univerzity Karlovy jako brigádník. Hledali jsme řešení pro evidenci času stráveného řešením úkolů a přiřazování úkolů pracovníkům. Do této chvíle se úkoly rozdělují různými variantami, ať posíláním emaily, slovně nebo v papírové podobě. Řešení, která jsme našli, nevyhovovala z různých důvodů. Nevyhovující důvody byly například drahá licence, předávání osobních informací na neznámý server nebo nevhodná funkčnost programu.

1.2 Cíl bakalářské práce

Cílem bakalářské práce je vytvoření webové aplikace, která bude umožňovat evidenci pracovních hodin zaměstnanců a kolegů, zejména v oboru informačních technologií. Díky této aplikaci bude možné udržovat přehled o aktuálních úkolech, evidovat jak dlouho na úkolech pracovali a porovnávat odhadovaný čas při zadání úkolu s celkovým časem vykázaným účastníky projektu. Tím pádem bude možné lépe plánovat úlohy a odhadnout jejich časovou náročnost.

1.3 Funkční a obecné požadavky

Obecné i funkční požadavky jsou požadavky od zákazníka. Funkční požadavky jsou vlastnosti, které by měla aplikace splňovat. Nefunkční požadavky nepodporují business cíle, ale jsou nutné k správnému chodu aplikace.

Funkční požadavky

1. Systém bude umožňovat zaznamenávat odpracované hodiny na jednotlivých úkolech.
2. Systém bude umožňovat zakládat, evidovat a mazat tickety sloužící k přiřazování úkolů a také odhadu času práce stráveného řešením daného úkolu.
3. Systém bude umožňovat zakládat projekty.
4. Systém bude umožňovat přiřazovat registrované uživatele k vytvořeným projektům.
5. Systém bude umožňovat zadávat, editovat a mazat milníky projektu.
6. Systém bude umožňovat porovnat odhadovaný čas práce s vykázaným časem.

7. Systém bude umožňovat tisk odpracovaných hodin účastníka projektu v určitém časovém období.
8. Systém bude umožňovat tisk odpracovaných hodin na celém projektu.

Obecné požadavky

1. Data aplikace se budou ukládat v jakékoli JDBC kompatibilní databázi (takové databáze jsou např.: MySQL, SQL, Oracle, ...).
2. Rozšiřitelnost – v případě zájmu do programu možnost přidání další funkcionality.
3. Jednoduchost, přehlednost a intuitivnost – aplikaci by měl zvládnout ovládat každý pracovník po krátkém zaškolení.
4. Aplikace bude provozována na serveru s operačním systémem Linux.

1.4 Analýza rizik

V této části popíšete rizika, která se mohou nastat během vývoje nebo po nasazení aplikace, a navrhnete řešení, jak těmto rizikům předcházet.

1.4.1 Riziko R01

Název	Nedostatečná analýza
Druh	procesní
Popis	Nedostatečné pochopení dané problematiky. Špatně provedená analýza a sběr požadavků.
Prevence	Kvalitní příprava před konzultací se zadavatelem a průzkum dané problematiky a konkurenčních systémů.
Protiopatření	V případě nejasností konzultace se zadavatelem.
Odhad pravděpodobnosti	15%
Závažnost	střední

Tabulka 1 - Riziko R01

1.4.2 Riziko R02

Název	Odklon od požadavků
Druh	procesní
Popis	Při vývoji aplikace nedostatečný důraz na počáteční analýzu.
Prevence	Kontrolovat funkčnost podle zadání.
Protipatření	Na konci vývoje provést testování vůči všem požadavkům.
Odhad pravděpodobnosti	10%
Závažnost	malá

Tabulka 2 - Riziko R02

1.4.3 Riziko R03

Název	Nedostatečné znalosti technologií
Druh	implementační
Popis	Během vývoje zjistit, že daná technologie potřebuje na implementaci více času nebo s ostatními technologiemi nejde použít.
Prevence	Vybírat si pouze známé technologie s řádnou dokumentací.
Protipatření	Konzultovat s kolegy z oboru.
Odhad pravděpodobnosti	15%
Závažnost	střední

Tabulka 3 - Riziko R03

1.4.4 Riziko R04

Název	Nekompatibilní prostředí
Druh	technologické
Popis	Riziko, že u zadavatele není vhodný server pro správný chod aplikace.
Prevence	Zjistit od zadavatele jaké servery vlastní a jaké technologie na nich používá.
Protiopatření	Používat pouze vhodné technologie.
Odhad pravděpodobnosti	5%
Závažnost	střední

Tabulka 4 - Riziko R04

1.4.5 Riziko R05

Název	Problémy s používáním aplikace
Druh	procesní
Popis	U koncových uživatelů se může stát, že aplikaci nebudou umět obsluhovat.
Prevence	Navrhnout intuitivní uživatelské rozhraní.
Protiopatření	Poskytovat školení.
Odhad pravděpodobnosti	5%
Závažnost	malá

Tabulka 5 - Riziko R05

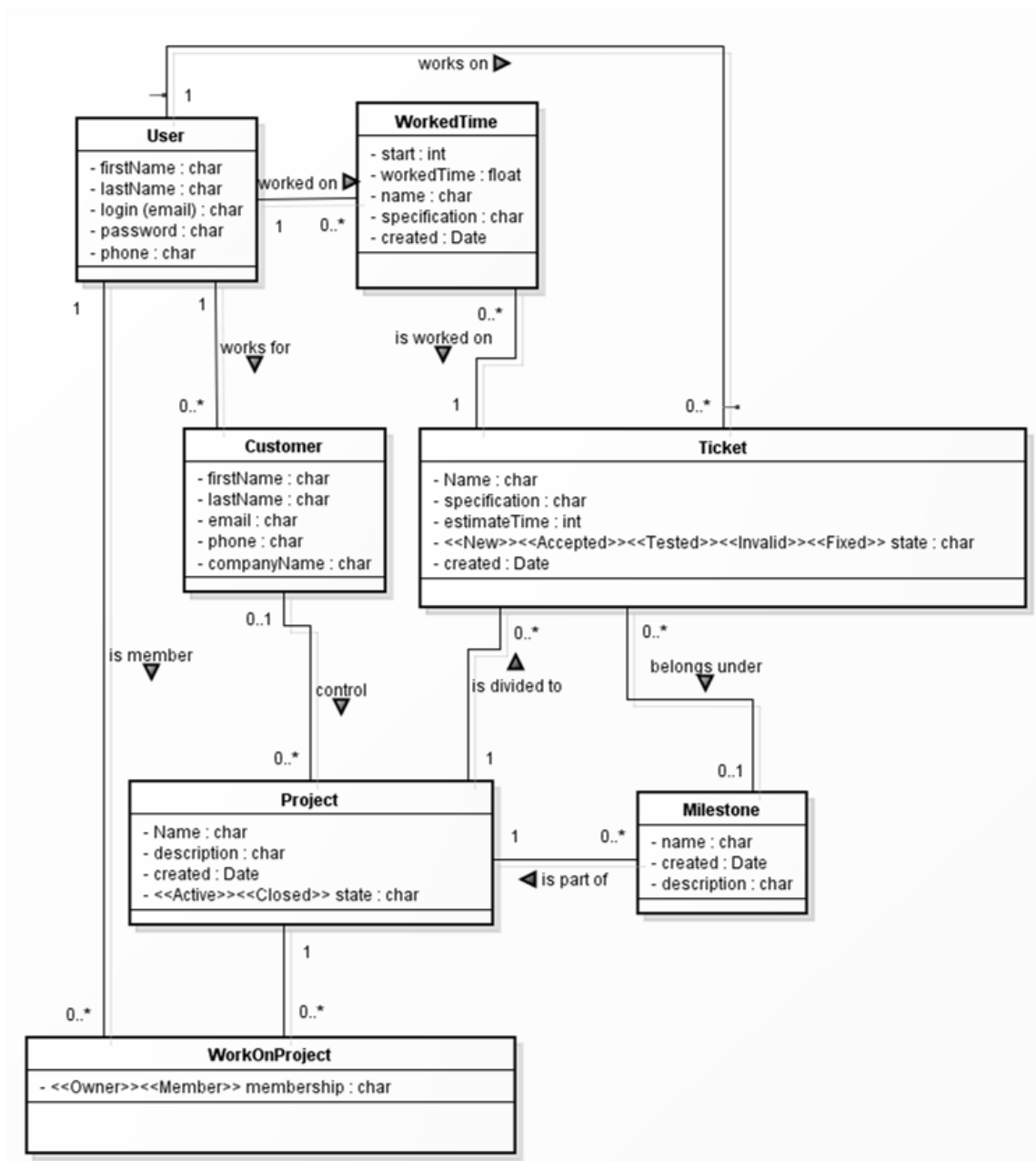
1.4.6 Riziko R06

Název	Ztráta dat
Druh	technologické
Popis	Ztráta dat z databáze, ať chybou serveru nebo nesprávným používáním.
Prevence	Pravidelné zálohování.
Protiopatření	Obnova ze zálohy
Odhad pravděpodobnosti	10%
Závažnost	střední

Tabulka 6 - Riziko R06

1.5 Konceptuální datový model s datovými typy

Konceptuální datový model je model popisující vztahy mezi jednotlivými entitami. Přidal jsem do modelu jednoduché datové typy, aby šlo poznat, jaký druh informace do nich přibližně budu ukládat.



Obrázek 1 - Konceptuální datový model s datovými typy

1.5.1 Popis konceptuálního datového modelu

System má za hlavní funkci evidenci hodin na jednotlivých úkolech projektu. Model je složen ze 7 entit: User, WorkedTime, Ticket, Project, Milestone, Customer, WorkOnProject. Uživatelé (entita User) mají možnost pracovat na projektech vedených

zákazníkem (entita Customer). Na projektech mohou vytvářet dílčí úkoly (Ticket). Dílčí úkoly mohou být přiřazeny pod dílčí cíl projektu (entita Milestone). Na jednotlivých ticketech mohou uživatelé projektu vykazovat odpracovaný čas (entita WorkedTime).

1.5.2 Rozepsání entit konceptuálního datového modelu

Entita User (Uživatel)

Entita Uživatel bude používána při přihlašování a registraci do systému. Obsahuje přihlašovací údaje uživatele. Jako přihlašovací jméno jsem zvolil emailovou adresu, protože ji má každý uživatel jedinečnou. Heslo bude hashované. Dále obsahuje jméno, příjmení a telefonní číslo uživatele pro možnost kontaktování v případě spolupráce na stejném projektu. Uživatelské rozhraní nebude obsahovat tabulku s uživateli, protože každý uživatel si bude své údaje moci editovat sám.

Atributy entity User (Uživatel)

Název atributu	Popis
firstName	Křestní jméno uživatele.
lastName	Příjmení uživatele.
login	Pracovní email uživatele, musí být jedinečný. Bude sloužit k přihlášení do systému.
password	Přihlašovací heslo. Z důvodu bezpečnosti musí být hesla hashovaná. K tomuto účelu použijte metodu z Spring Security.
phone	Telefonní číslo uživatele.

Tabulka 7 - Atributy entity User (Uživatel)

Entita Customer (Zákazník)

Entita Zákazník bude sloužit k ukládání zadavatelů projektů. Její hlavní význam je zachování kontaktních údajů zadavatelů projektů, aby každý účastník projektu mohl v případě nejasností kontaktovat zadavatele. Přihlášený uživatel si bude moci vytvořit své zákazníky a jim přiřazovat projekty. Tím také bude mít přehled o všech projektech, které pro zákazníka vypracoval.

Atributy entity Customer (Zákazník)

Název atributu	Popis
firstName	Křestní jméno zadavatele projektu nebo kontaktní osoby.
lastName	Příjmení zadavatele projektu nebo kontaktní osoby.
email	Pracovní email zadavatele projektu nebo kontaktní osoby.
phone	Pracovní telefon zadavatele projektu nebo kontaktní osoby.
companyName	V případě, že zadavatel bude společnost nebo oddělení, tak toto pole bude obsahovat jeho jméno.

Tabulka 8 - Atributy entity Customer (Zákazník)

Entita Project (Project)

Entita Projekt bude využívána v případě, že přihlášený uživatel bude opakovaně pracovat na nějakém tématu nebo činnosti a bude si chtít udržovat přehled o opracovaných hodinách. Každý přihlášený uživatel si může vytvořit projekt pod svým zákazníkem a může do něj přizvat registrované uživatele v systému.

Atributy entity Project Projekt (Projekt)

Název atributu	Popis
name	Jméno projektu.
description	Podrobný popis účelu projektu.
created	Datum a čas vytvoření projektu, zaznamená se automaticky v době jeho vytvoření.
state	Stav projektu. Stav projektu bude moci nabývat dvou hodnot. První hodnota je aktivní (Active) a druhá hodnota je uzavřen (Closed). Aktivní je projekt v době, kdy na projektu někdo pracuje nebo není dokončen. Vlastník projektu může jeho stav změnit na uzavřený, pokud už nechce, aby na něm účastníci projektu vykazovali další práci, nebo je projekt dokončen.

Tabulka 9 - Atributy entity Project Projekt (Projekt)

Entita WorkOnProject (Pracuje na projektu)

Entita WorkOnProject je mezi uživatelem a projektem a slouží k zaznamenání oprávnění účastníků projektu.

Atribut entity WorkOnProject (Pracuje na projektu)

Název atributu	Popis
membership	Položka určuje oprávnění uživatele na projektu. Může nabývat dvou hodnot. První hodnota je <i>Owner</i> (Vlastník). Této hodnoty uživatel nabude při založení svého projektu. Druhá hodnota je <i>Member</i> (člen projektu). Člen projektu má nižší oprávnění. Nemůže například editovat nebo mazat projekt a přidávat nebo mazat uživatele projektu. Vlastník projektu může do projektu přiřadit další uživatele aplikace. Přidávaným uživatelům bude moci zvolit, jaká práva bude mít vůči projektu.

Tabulka 10 - Atribut entity WorkOnProject (Pracuje na projektu)

Entita Ticket (Úkol)

Entita Ticket bude používána k zaznamenání dílčích úkolů týkajících se projektu. U dílčích úkolů budou uživatelé projektu odhadovat potřebný čas pro dokončení úkolu. Ten k úkolu také zaznamenají. Ticket půjde přiřadit k účastníkovi projektu, aby bylo jasné, kdo v současné době úkol řeší.

Atributy entity Ticket (Úkol)

Název atributu	Popis
name	Název úkolu.
specification	Podrobnější popis, čeho se úkol týká.
estimateTime	Odhadovaný čas pro vyřešení úkolu.
created	Datum a čas vytvoření ticketu. Vytvoří se automaticky při založení ticketu.
state	Stav úkolu. Stav může nabývat 5 hodnot: New, Accepted, Tested, Invalid, Fixed

Tabulka 11 - Atributy entity Ticket (Úkol)

Stavy entity Ticket (Úkol)

Stavy entity Ticket	Popis
New	Nový úkol, který ještě není přijatý řešitelem úkolu.
Accepted	Řešitel úkolu se na úkol podíval a změnil jeho stav z New na Accepted, což znamená, že úkol přijal a pracuje na něm.
Tested	Tento stav znamená, že řešitel úkol dokončil a nyní je potřeba otestovat správnou funkčnost.
Fixed	Úkol je dokončený i otestovaný. Tato hodnota znamená, že se úkolem dále nemusí řešitel zabývat a usuzuje ho jako dokončený.
Invalid	Tento stav znamená, že účastník projektu vyhodnotil úkol jako neopodstatněný nebo nepotřebný, a proto jej neřeší.

Tabulka 12 - Stavy entity Ticket (Úkol)

Entita WorkedTime (Odpracovaný čas)

Entitu Odpracovaný čas budou používat účastníci projektu ve vykazování času stráveného řešením ticketu. Vykázat práci na ticketu mohou účastníci projektu mnohokrát.

Atributy entity WorkedTime (Odpracovaný čas)

Název atributu	Popis
start	Čas a datum začátku práce na ticketu. Vyplní ho řešitel ticketu.
workedTime	Délka času stráveného řešením ticketu zaokrouhleného na čtvrt hodinu.
name	Název vykázané práce. Nejlépe krátký a výstižný text shrnující na čem účastník projektu pracoval.
specification	Podrobný popis vykazované práce.
created	System uloží v okamžiku vytvoření vykázané práce datum a čas jejího vytvoření.

Tabulka 13 - Atributy entity WorkedTime (Odpracovaný čas)

Entita Milestone

Entita Milestone bude používána v případě, že více ticketů řeší dílčí cíl v rámci projektu. Milestone jde přiřazovat k jednotlivým ticketům projektu.

Atributy entity Milestone

Název atributu	Popis
name	Název dílčího cíle projektu.
description	Popis dílčího cíle projektu.
created	Systém uloží v okamžiku vytvoření Milestonu datum a čas jeho vytvoření.

Tabulka 14 - Atributy entity Milestone

1.5.3 Vztahy vystupující v konceptuálním datového modelu

works on (vztah mezi entitami User a Ticket)

Každý ticket může mít v jednu chvíli jednoho účastníka projektu (uživatele), který na ticketu pracuje. Ticket může mít pouze jednoho řešitele.

worked on (vztah mezi entitami User a WorkedTime)

Účastník projektu (uživatel) může mít mnoho vykázanych hodin práce, na kterých pracoval, avšak vykázané hodiny musí být přiřazeny právě jednomu účastníkovi projektu (uživateli).

works for (vztah mezi entitami User a Customer)

Uživatel si může vytvořit více zákazníků, pro které pracuje, ale zákazník náleží právě jednomu uživateli.

is worked on (vztah mezi entitami WorkedTime a Ticket)

Odpracovaný čas je vykázán právě na jednom ticketu projektu, ale jeden ticket může mít mnoho odpracovaných časů.

is member (vztah mezi entitami User a WorkOnProject)

Uživatel může být členem více projektů a projekt může mít za členy více uživatelů.

control (vztah mezi entitami Customer a Project)

Každý projekt je kontrolován právě jedním zákazníkem, avšak zákazník může mít pod kontrolou více projektů.

is dividend to (vztah mezi entitami Project a Ticket)

Projekt je rozdělený na více ticketů, avšak ticket musí být právě pod jedním projektem.

is part of (vztah mezi entitami Milestone a Project)

Milestone je částí právě jednoho projektu, avšak projekt může mít více milestoneů.

belongs under (vztah mezi entitami Ticket a Milestone)

Milestone může mít více ticketů, ale ticket může patřit pod jeden Milestone.

1.6 Aktéři systému

Na obrázku můžeme vidět aktéry systému. Aktéři jsou podrobněji popsáni v tabulce 1. Všichni aktéři systému jsou fyzické osoby.



Obrázek 2 - Uživatelské role systému

V následující tabulce *Definice uživatelských rolí* jsou definovány i další role, které se dále používají pro přehlednější popis.

1.7 Definice uživatelských rolí

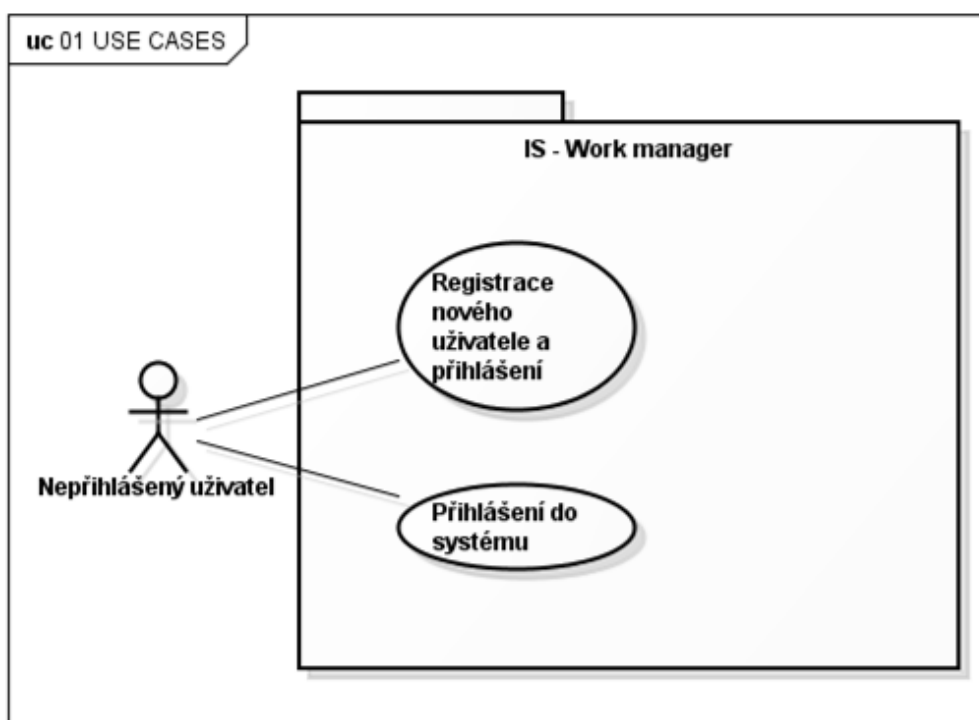
Role	Popis role
Nepřihlášený uživatel	Uživatel přistupující na stránky a není přihlášený. Stačí v prohlížeči zadat URL adresu.
Registrovaný uživatel	Uživatel, který úspěšně dokončil proces registrace (PU01).
Přihlášený uživatel	Uživatel autentizovaný přihlašovacími údaji validovanými oproti databázi.
Účastník projektu	Vlastník nebo člen projektu. Toto označení používám například při výpisech odpracovaných hodin, protože obě role mohou sledovat všechny vykázané hodiny na projektu.
Člen projektu	Uživatel přiřazený do projektu jako člen. Tato role má omezená práva a nemůže zasahovat do výkazů ostatních uživatelů projektu. Jedná se o spolupracovníka nebo podřízeného pracovníka, který se podílí na práci na projektu.
Vlastník projektu	Vlastníkem projektu se stane uživatel, který projekt založí nebo je do projektu přiřazen jako vlastník. Rozšiřuje práva uživatele. Umožňuje navíc spravovat vykázané hodiny a údaje o projektu. Také může projekt smazat a tím odstraní všechny záznamy projektu z databáze. Vlastník projektu je osoba, která má na starosti vedení projektu a mentorování projektu.

Tabulka 15 - Definice uživatelských rolí

1.8 Případy užití

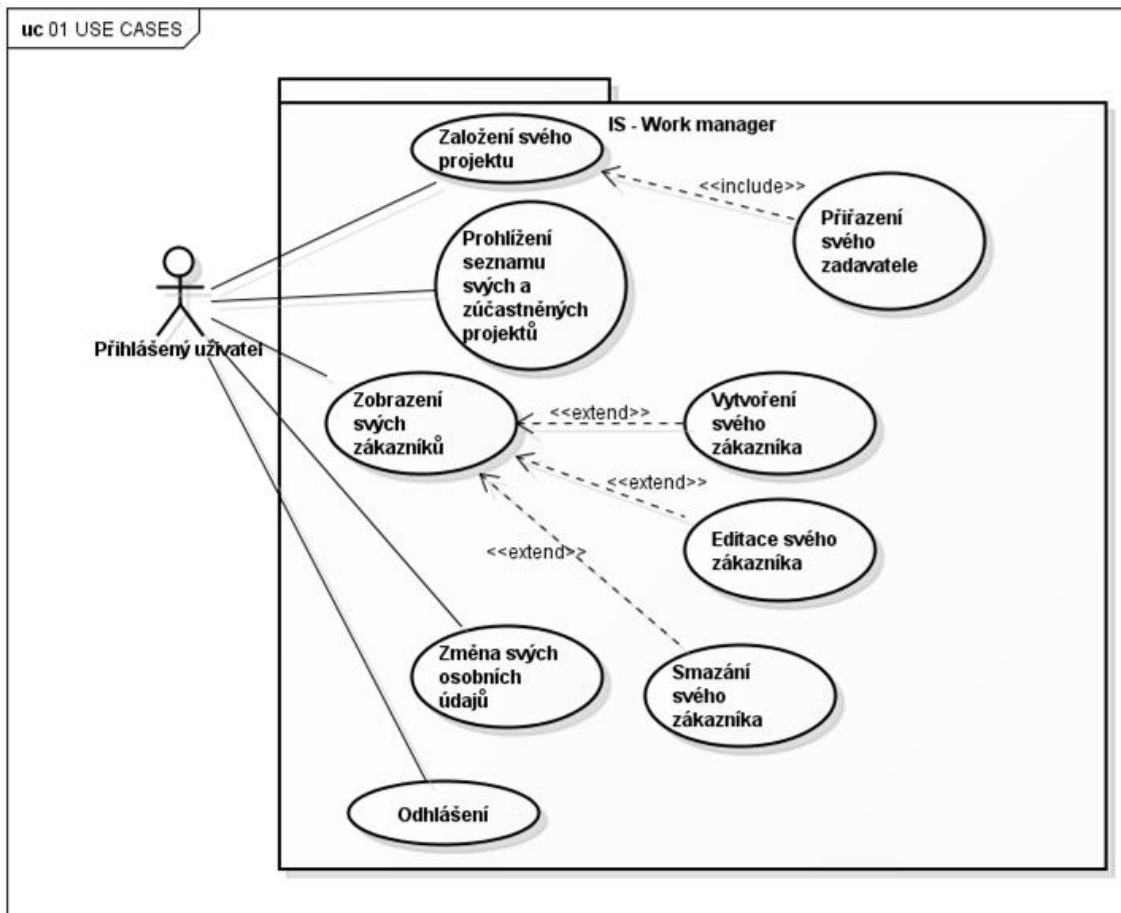
Diagramy případů užití (use case diagram) se používají k popisu chování systému a uživatele. Umožňují popisovat interakce mezi uživatelem a systémem. Jsou to jedny z prvních diagramů, které se tvoří při vývoji systému. Vyplynají z funkčních požadavků a rozšiřují jejich popis. Určují tedy, co by měl systém za jakých podmínek dělat.

Diagram případů užití nepřihlášeného uživatele



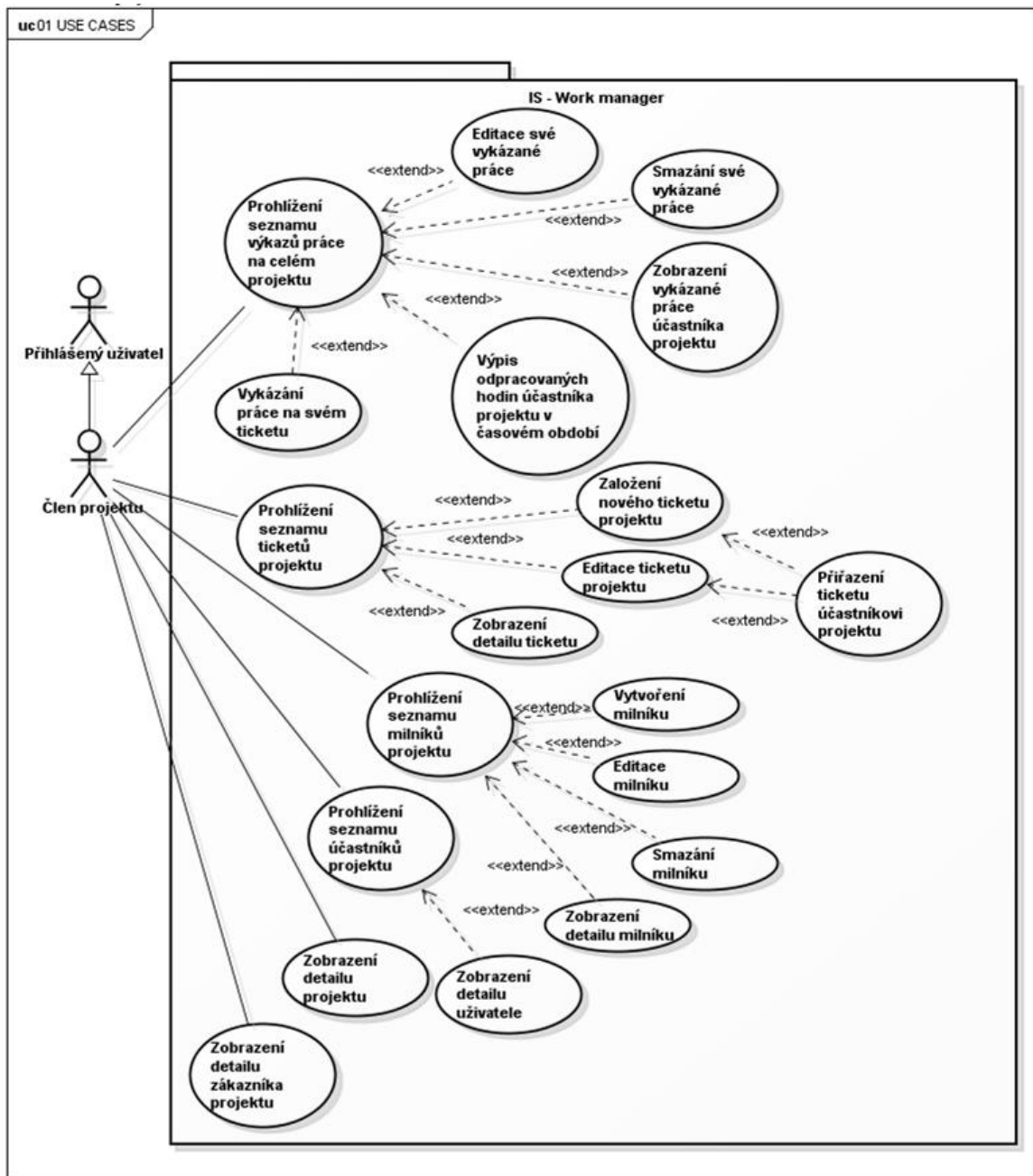
Obrázek 3 - Diagram případů užití nepřihlášeného uživatele

Diagram případů užití přihlášeného uživatele



Obrázek 4 - Diagram případů užití přihlášeného uživatele

Diagram případů užití člena projektu



Obrázek 5 - Diagram případů užití člena projektu

Diagram případů užití vlastníka projektu



Obrázek 6 - Diagram případů užití vlastníka projektu

1.8.1 Rozepsané případy užití

Podrobnější popis případů užití podle metodiky brief use case.

Registrace nového uživatele a přihlášení

ID: PU01	
Popis:	Registrace nového uživatele do systému.
Akteři:	Nepřihlášený uživatel
Vstupní podmínky:	Správě zadané registrační údaje.
Následné podmínky:	Přidání nového uživatele do systému a jeho následné přihlášení.
Základní tok událostí: <ol style="list-style-type: none">1. Uživatel klikne na tlačítko registrace.2. Systém zobrazí registrační formulář.3. Uživatel vyplní registrační údaje a stiskne tlačítko registrovat4. Systém ověří vyplněné údaje.5. Systém přidá nového uživatele do databáze a přihlásí ho do systému.	
Alternativní tok událostí: <ol style="list-style-type: none">4. V případě, že vyplněné údaje nejsou správné, systém uživatele upozorní.	

Tabulka 16 - PU1 - Registrace nového uživatele a přihlášení

Přihlášení

ID: PU02	
Popis:	Přihlášení uživatele do aplikace. Pro používání aplikace se musí uživatel přihlásit.
Akteři:	Nepřihlášený uživatel
Vstupní podmínky:	Správě zadané uživatelské jméno (email) a heslo.
Následné podmínky:	Přihlášení uživatele do systému.
Základní tok událostí: <ol style="list-style-type: none">1. Uživatel klikne na tlačítko přihlášení.2. Systém zobrazí přihlašovací formulář.3. Uživatel vyplní uživatelské jméno a heslo a stiskne tlačítko přihlásit.4. Systém ověří přihlašovací údaje.5. Systém přihlásí uživatele.	
Alternativní tok událostí:	

4. V případě, že vyplněné údaje nejsou správné, systém uživatele upozorní.

Tabulka 17 - PU2 - Přihlášení

Odhlášení

ID: PU03	
Popis:	Umožňuje přihlášené uživateli odhlášení ze systému.
Akteři:	Přihlášený uživatel
Vstupní podmínky:	Uživatel musí být přihlášen.
Následné podmínky:	Uživatel je odhlášený ze systému.
Základní tok událostí: <ol style="list-style-type: none">1. Uživatel klikne na tlačítko odhlášení.2. Systém uživatele odhlásí.	
Alternativní tok událostí: Není	

Tabulka 18 - PU3 - Odhlášení

Prohlížení seznamu svých a zúčastněných projektů

ID: PU04	
Popis:	Umožňuje přihlášené uživateli zobrazit seznam svých projektů a projektů, do kterých byl uživatel přidán vlastníky projektů.
Akteři:	Přihlášený uživatel
Vstupní podmínky:	Uživatel musí být přihlášen.
Následné podmínky:	Zobrazení seznamu projektů.
Základní tok událostí: <ol style="list-style-type: none">1. Uživatel klikne na tlačítko pro zobrazení projektů.2. Systém zobrazí seznam projektů.	
Alternativní tok událostí: Není	

Tabulka 19 - PU4 - Prohlížení seznamu svých a zúčastněných projektů

Vytvoření svého zákazníka

ID: PU05	
Popis:	Umožňuje přihlášené uživateli vytvořit nového zákazníka, kterému bude moci přiřazovat své nové projekty
Aktéři:	Přihlášený uživatel
Vstupní podmínky:	Uživatel musí být přihlášen.
Následné podmínky:	Přidání zákazníka do systému a jeho přiřazení k uživateli.
Základní tok událostí: <ol style="list-style-type: none">1. Přihlášený uživatel klikne na tlačítko přidat zákazníka.2. Systém zobrazí formulář pro přidání zákazníka.3. Uživatel vyplní formulář a stiskne tlačítko vytvořit.4. Systém zkontroluje, zda jsou vložené údaje ve správném formátu.5. Systém přidá zákazníka do databáze.6. Systém zobrazí detail vytvořeného zákazníka a vypíše uživateli hlášku o jeho vytvoření	
Alternativní tok událostí: <ol style="list-style-type: none">4. V případě, že vyplněné údaje nejsou správné, systém uživatele upozorní.	

Tabulka 20 - PU5 - Vytvoření svého zákazníka

Editace svého zákazníka

ID: PU06	
Popis:	Umožňuje přihlášené uživateli editovat svého, již vytvořeného zákazníka.
Aktéři:	Přihlášený uživatel
Vstupní podmínky:	Uživatel musí být přihlášen a musí mít vytvořeného svého zákazníka.
Následné podmínky:	Aktualizované údaje zákazníka v systému.
Základní tok událostí: <ol style="list-style-type: none">1. Uživatel klikne na tlačítko editace zákazníka.2. Systém zobrazí formulář s vyplněnými údaji uloženými v databázi.3. Uživatel změní potřebné informace o zákazníkovi.4. Systém zkontroluje nově zadané údaje	

5. Systém zobrazí detail aktualizovaného zákazníka a vypíše uživateli hlášku o jeho aktualizaci.
Alternativní tok událostí:
4. V případě, že vyplněné údaje nejsou správné, systém uživatele upozorní.

Tabulka 21 - PU6 - Editace svého zákazníka

Smazání zákazníka

ID: PU07	
Popis:	Umožňuje přihlášené uživateli smazat svého zákazníka.
Aktéři:	Přihlášený uživatel
Vstupní podmínky:	Uživatel musí být přihlášen a musí mít vytvořeného svého zákazníka.
Následné podmínky:	Smazání zákazníka ze systému.
Základní tok událostí:	
<ol style="list-style-type: none"> 1. Uživatel klikne na tlačítko smazání zákazníka. 2. Systém zobrazí potvrzovací okno s upozorněním smazání zákazníka. 3. Uživatel potvrdí tlačítkem OK. 4. Systém odstraní zákazníka z databáze. 	
Alternativní tok událostí:	
3. V případě, že uživatel klikne na tlačítko zrušit, tak smazání neproběhne.	

Tabulka 22 - PU7 - Smazání zákazníka

Založení svého projektu

ID: PU08	
Popis:	Umožňuje přihlášené uživateli založit svůj nový projekt.
Aktéři:	Přihlášený uživatel
Vstupní podmínky:	Uživatel musí být přihlášen a musí mít vytvořeného svého zákazníka.
Následné podmínky:	Založený nový projekt zapsaný v systému a přiřazený k přihlášenému uživateli. Stav uživatele vůči tomuto projektu je vlastník.
Základní tok událostí: <ol style="list-style-type: none">1. Uživatel klikne na tlačítko pro založení projektu.2. Systém zobrazí formulář s údaji pro založení projektu.3. Uživatel vyplní formulář.4. Uživatel zvolí zákazníka, ke kterému se má projekt přiřadit.5. Uživatel klikne na tlačítko vytvořit.6. Systém zkontroluje vložené údaje.7. Systém zapíše projekt do databáze.8. Systém zobrazí uživateli stránku s detaily projektu a vypíše uživateli hlášku o vytvoření projektu.	
Alternativní tok událostí: <ol style="list-style-type: none">4. V případě, že uživatel nemá vytvořeného zákazníka, nebude moci vytvořit projekt. Musí nejdříve projít PU5.6. V případě, že vyplněné údaje nejsou správné, systém uživatele upozorní.	

Tabulka 23 - PU8 - Založení svého projektu

Přidání registrovaného uživatele do projektu

ID: PU09	
Popis:	Umožňuje vlastníkovu projektu přidat uživatele, který je registrovaný v systému, do projektu a tím se stane uživatel účastníkem projektu.
Aktéři:	Vlastník projektu, registrovaný uživatel
Vstupní podmínky:	Uživatel musí být přihlášen, musí být vlastníkem projektu a přidání uživatele do projektu musí být registrován v systému.
Následné podmínky:	Přidání uživatele je v systému přiřazen k projektu. V systému je také zaznamenána role uživatele v projektu.
Základní tok událostí:	
<ol style="list-style-type: none"> 1. Vlastník projektu na stránce vyplní přihlašovací jméno (email) registrovaného uživatele v systému. 2. Vlastník projektu zvolí, zda má být přidán uživatel vlastníkem projektu nebo členem. 3. Vlastník projektu klikne na tlačítko přidat člena. 4. Systém zkontroluje, zda vyplněné přihlašovací jméno registrovaného uživatele odpovídá některému z uživatelů databáze. 5. Systém přidá vazbu registrovaného uživatele k projektu s příslušnou rolí – vlastník nebo člen. 6. Systém zobrazí hlášku uživateli, že registrovaný uživatel byl úspěšně přidán do projektu a obnoví stránku s účastníky projektu, tak aby zde byl zobrazen i nově přidání účastník. 	
Alternativní tok událostí:	
<ol style="list-style-type: none"> 4. V případě, že email není registrován v databázi uživatelů systému, systém zobrazí vlastníkovu projektu hlášku o chybně zadaném emailu. 	

Tabulka 24 - PU9 - Přidání registrovaného uživatele do projektu

Výpis odpracovaných hodin účastníka projektu v časovém období

ID: PU10	
Popis:	Umožňuje účastníkovi projektu vypsát odpracované hodiny účastníka projektu v určitém časovém období.
Akteři:	Účastník projektu
Vstupní podmínky:	Uživatel musí být přihlášen a musí být účastníkem projektu
Následné podmínky:	Žádné
Základní tok událostí: <ol style="list-style-type: none">1. Účastník projektu zvolí časové období, ve kterém chce provést výpis odpracovaných hodin.2. Účastník projektu zvolí některého z účastníků projektu, kterého se bude výpis týkat.3. Účastník projektu klikne na tlačítko zobrazení výběru odpracovaných hodin.4. Systém vypíše vykázané práce zvoleného účastníka projektu v daném období a vytiskne součet těchto vypsanych hodin.	
Alternativní tok událostí: Žádný	

Tabulka 25 - PU10 - Výpis odpracovaných hodin účastníka projektu v časovém období

Vykázání práce na svém ticketu

ID: PU11	
Popis:	Umožňuje účastníkovi projektu vykázat odpracované hodiny na svém ticketu.
Aktéři:	Účastník projektu
Vstupní podmínky:	Uživatel musí být přihlášen a musí být účastníkem projektu.
Následné podmínky:	Připsání vykázaných hodin do systému.
Základní tok událostí: <ol style="list-style-type: none">1. Účastník projektu klikne na tlačítko „Přidat můj čas“2. Systém zobrazí uživateli formulář pro vykázání času.3. Uživatel zvolí, ke kterému ticketu chce čas přiřadit.4. Uživatel vyplní zbytek formuláře včetně počtu odpracovaných hodin.5. Uživatel klikne na tlačítko pro vytvoření.6. Systém zkontroluje vyplněné údaje.7. Systém zapíše vykázané hodiny do systému, přiřadí je k ticketu a k uživateli.	
Alternativní tok událostí: Žádný	

Tabulka 26 - PU11 - Vykázání práce na svém ticketu

1.8.2 Další případy užití

Protože návrh obsahuje mnoho případů užití a jejich vypsání by zabralo příliš stránek, rozhodl jsem se, že další případy užití vypíši do následující tabulky. Případy užití v tabulce vypsány jsou buď triviální, nebo jsou podobné již vypsáným případům užití.

Název případu užití	Popis
Zobrazení svých zákazníků	Triviální
Změna svých osobních údajů	Podobné, jako PU6
Prohlížení seznamu výkazů práce na celém projektu	Podobné, jako PU4
Editace své vykázané práce	Podobné, jako PU6
Smazání své vykázané práce	Podobné, jako PU7
Zobrazení vykázané práce účastníka projektu	Triviální
Prohlížení seznamu ticketů projektu	Podobné, jako PU4
Založení nového ticketu projektu	Podobné, jako PU5
Editace ticketu projektu	Podobné, jako PU6
Přiřazení ticketu účastníkovi projektu	Podobné, jako PU6
Zobrazení detailu ticketu	Triviální
Prohlížení seznamu milníků projektu	Podobné, jako PU4
Vytvoření milníku	Podobné, jako PU5
Editace milníku	Podobné, jako PU6
Smazání milníku	Podobné, jako PU7
Zobrazení detailu milníku	Triviální
Prohlížení seznamu účastníků projektu	Podobné, jako PU4
Zobrazení detailu uživatele	Triviální
Zobrazení detailu projektu	Triviální
Zobrazení detailu zákazníka projektu	Triviální
Smazání uživatele z projektu	Podobné, jako PU7
Editace vykázaných hodin účastníka projektu	Podobné, jako PU6
Přeřazení vykázané práce k jinému ticketu	Podobné, jako PU6
Smazání projektu	Podobné, jako PU7
Editace projektu	Podobné, jako PU6

Změna stavu projektu	Podobné, jako PU6
----------------------	-------------------

Tabulka 27 - Další případy užití

1.8.3 Oprávnění uživatelů systému

Oprávnění v systému jsou zřejmí již z diagramů aktérů a případů užití. Pro přehlednost je navíc shrnu v následující tabulce. Položky označené „X“ znamenají, že příslušný uživatel má oprávnění příslušný případ užití provést. Označení „N/A“ znamená, že uživatel nemá oprávnění příslušný případ užití provést.

Název případu užití	Nepřihlášený uživatel	Přihlášený uživatel	Člen projektu	Vlastník projektu
Registrace nového uživatele a přihlášení	X	X	X	X
Přihlášení	X	X	X	X
Odhlášení	X	X	X	X
Prohlížení seznamu svých a zúčastněných projektů	N/A	X	X	X
Vytvoření svého zákazníka	N/A	X	X	X
Editace svého zákazníka	N/A	X	X	X
Smazání zákazníka	N/A	X	X	X
Založení svého projektu	N/A	X	X	X
Přidání registrovaného uživatele do projektu	N/A	N/A	N/A	X
Výpis odpracovaných hodin účastníka projektu v časovém období	N/A	N/A	X	X
Vykázání práce na svém ticketu	N/A	N/A	X	X
Zobrazení svých zákazníků	N/A	X	X	X
Změna svých osobních údajů	N/A	X	X	X
Prohlížení seznamu výkazů práce na celém projektu	N/A	N/A	X	X
Editace své vykázané práce	N/A	N/A	X	X
Smazání své vykázané práce	N/A	N/A	X	X
Zobrazení vykázané práce účastníka projektu	N/A	N/A	X	X
Prohlížení seznamu ticketů projektu	N/A	N/A	X	X
Založení nového ticketu projektu	N/A	N/A	X	X
Editace ticketu projektu	N/A	N/A	X	X
Přiřazení ticketu účastníkovi projektu	N/A	N/A	X	X
Zobrazení detailu ticketu	N/A	N/A	X	X
Prohlížení seznamu milníků projektu	N/A	N/A	X	X

Název případu užití	Nepřihlášený uživatel	Přihlášený uživatel	Člen projektu	Vlastník projektu
Vytvoření milníku	N/A	N/A	X	X
Editace milníku	N/A	N/A	X	X
Smazání milníku	N/A	N/A	X	X
Zobrazení detailu milníku	N/A	N/A	X	X
Prohlížení seznamu účastníků projektu	N/A	N/A	X	X
Zobrazení detailu uživatele	N/A	N/A	X	X
Zobrazení detailu projektu	N/A	N/A	X	X
Zobrazení detailu zákazníka projektu	N/A	N/A	X	X
Smazání uživatele z projektu	N/A	N/A	N/A	X
Editace vykázaných hodin účastníka projektu	N/A	N/A	N/A	X
Přeražení vykázané práce k jinému ticketu	N/A	N/A	N/A	X
Smazání projektu	N/A	N/A	N/A	X
Editace projektu	N/A	N/A	N/A	X
Změna stavu projektu	N/A	N/A	N/A	X

Tabulka 28 - Oprávnění uživatelů v systému

2 Hlavní použité nástroje a technologie

2.1 Použité vývojové nástroje

Zde jsou vypsané vývojové nástroje, které jsem využíval při tvorbě programu.

IntelliJ Idea

IntelliJ Idea[6] je vývojové prostředí. Podporuje všechny programovací jazyky, jež jsem využíval. A co je nejdůležitější, umožňuje vyvíjet ve webovém frameworku Grails, v kterém je aplikace psaná. To bylo také hlavním kritériem pro výběr. Umožňuje projekt napsaný ve frameworku Grails také zkompilovat, spustit, debugovat a využívat Grails z příkazové řádky.

Notepad++

Notepad++ je aplikace sloužící místo poznámkového bloku. Umožňuje jednoduchou editaci zdrojového kódu. Podporuje několik programovacích jazyků. Využíval jsem ho například při editaci XML, HTML, CSS a mnoho dalších. Program jsem si vybral pro jeho rychlost a jednoduchost.

Astah

Program Astah jsem používal při tvorbě obrázků a diagramů. Program podporuje UML (Unified Modeling Language) jazyk pro vizualizaci, navrhování a dokumentaci softwarových systémů. Vybral jsem si tento program, protože mi umožnil tvorbu většiny obrázků použitých v tomto dokumentu.

2.2 Použité technologie

V této části stručně popíši hlavní technologie, které jsem použil.

Grails

Grails je open source framework pro vývoj webových aplikací. Protože je to hlavní technologie vybraná k vytvoření projektu, tak je podrobněji popsán v oddílu 3.1.

Twitter bootstrap

Twitter bootstrap je front-end framework, který jsem používal k tvorbě uživatelského rozhraní. Vybral jsem si ho, protože obsahuje širokou základnu komponent pro tvorbu uživatelského rozhraní.

Apache Tomcat

Apache Tomcat je open source aplikační server založený na jazyku Java. Používám ho k spouštění vytvořené aplikace.

XAMPP

XAMPP (Apache + MySQL + PHP + Perl) je oblíbený program na vytvoření vývojového prostředí. Použil jsem z něho části obsluhující server Apache, databázi MySQL a program na správu databáze phpMyAdmin.

phpMyAdmin

Program phpMyAdmin jsem použil k administraci databáze MySQL. Je napsaný v php a pomocí webového rozhraní s ním můžete jednoduše zobrazit, editovat nebo vložit záznamy. Tímto programem jsem kontroloval správnost zapsaných záznamů v databázi.

Apache

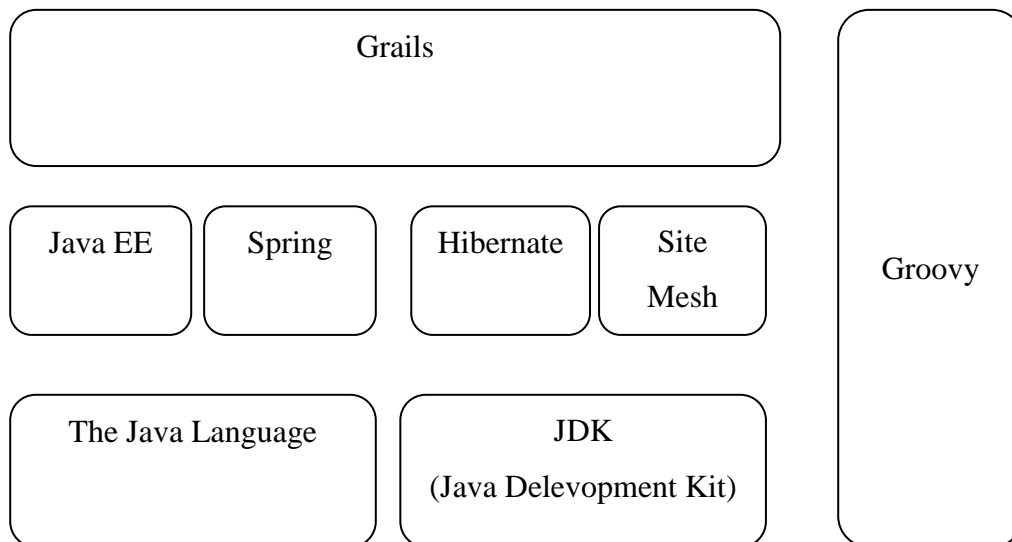
Apache je softwarový HTTP server. Je to program běžící na hardwarovém stroji zajišťující obsluhu prohlížečů jednotlivých návštěvníků. Je dostupný pro všechny hlavní platformy (Linux, Windows...). Na serveru Apache jsem spouštěl program phpMyAdmin.

3 Implementace

3.1 Popis použitého frameworku Grails

Pro lepší pochopení architektury aplikace je nutné popsat, jaké hlavní technologie framework Grails používá a jakým způsobem funguje. Framework Grails používá dnes již standart - návrhový vzor MVC (model-view-controller).

Hlavní technologie používané frameworkem Grails



Obrázek 7 - Orientační obrázek hlavních technologií ve frameworku Grails

Orientační obrázek hlavních technologií ve frameworku Grails, převzat z knihy The definitive guide to Grails 2[1],s.2

Java EE

Java EE je platforma založená na jazyku Java sloužící pro vývoj a provoz informačních systémů a podnikových aplikací.

Spring

Spring je populární framework pro vývoj JavaEE aplikací. Spring využívá návrhový vzor IoC (Inversion of control). Návrhový vzor zabezpečuje přesouvání zodpovědnosti za vytvoření a provázání objektů na Spring framework.

Hibernate

Hibernate je framework napsaný v jazyce Java, umožňující ORM (objektově-relační mapování).

Site mesh

Robustní a stabilní framework pro generování uživatelského rozhraní.

Groovy

Groovy je objektově orientovaný skriptovací programovací jazyk. Jeho účelem je zjednodušení psaní webových, databázových nebo desktopových aplikací a snížit redundanci kódu.

The Java Language

Java je objektově orientovaný programovací jazyk. Jeho hlavní výhodou je přenositelnost.

Java development kit

Java development kit obsahuje základní soubor nástrojů pro vývoj aplikací v programovacím jazyce Java.

3.2 Popis struktury aplikace

V této části popíši strukturu vytvořené aplikace. Vysvětlím hlavní části aplikace a části aplikace, které jsem použil.

3.2.1 Kořenový adresář aplikace

Cesta: /workManager

Tento adresář obsahuje celý vytvořený program – konfigurační soubory, controllery, doménové třídy, service, views, překlady a další...

3.2.2 Konfigurační soubory aplikace

Cesta: /workManager/grails-app/conf

Adresář obsahuje konfigurační soubory, které jsou rozepsány níže.

ApplicationResources.groovy

Cesta: /workManager/grails-app/conf/ApplicationResources.groovy

V tomto souboru jsou uloženy zdroje a závislosti, které jsou použity při generování stránek uživatelského rozhraní.

Bootstrap.groovy

Cesta: /workManager/grails-app/conf/Bootstrap.groovy

Soubor se spustí po každém startu aplikace ve vývojovém módu. Využívá se pro vložení testovacích dat do databáze. Vytvořil jsem posloupnost příkazů, která mi zajistí přidání dvou uživatelů do databáze a přiloží k nim projekty s odpracovanými hodinami. Více o tomto souboru napíši v kapitole o testování.

BuildConfig.groovy

Cesta: /workManager/grails-app/conf/BuildConfig.groovy

V tomto souboru se určuje konfigurace Java virtual machine¹ během běhu aplikace a závislost aplikace na pluginech. Soubor buildConfig.groovy se spouští před během programu.

Config.groovy

Cesta: /workManager/grails-app/conf/Config.groovy

V tomto souboru se určuje konfigurace aplikace. Používá se během běhu programu.

DataSource.groovy

Cesta: /workManager/grails-app/conf/DataSource.groovy

V tomto souboru se určuje konfigurace a chování databáze. Zvolil jsem databázi MySQL. Musel jsem zde nastavit driver, který se má použít, a přístupové údaje do databáze.

¹ JVM (Java virtual machine) je program, v němž se spouští aplikace psané v programovacím jazyce Java

UrlMapping.groovy

Cesta: /workManager/grails-app/conf/UrlMapping.groovy

Zde se nastavuje, jaké URL bude aplikace používat, jakým způsobem. V tomto souboru mám nastavené, co se zobrazí na domovské stránce, na stránce s chybovým hlášením a jakým způsobem se adresují konkrétní objekty databáze.

3.2.3 Domain

Cesta: /workManager/grails-app/domain/org/workmanager

Adresář obsahuje soubory určující práci s entitami v databázi, vztahy mezi nimi, atributy, validátory...

3.2.4 Controllers

Cesta: /workManager/grails-app/controllers/org/workmanager

Adresář obsahuje soubory řadičů, které reagují na události a zajišťují změny v databázi nebo v uživatelském rozhraní aplikace.

3.2.5 Views

Cesta: /workManager/grails-app/views/org/workmanager

Adresář obsahuje soubory, které převádí data reprezentovaná modelem do formy vhodné pro presentaci uživateli.

3.2.6 Services

Cesta: /workManager/grails-app/services/org/workmanager

Adresář obsahuje soubory, kam se umísťuje složitější logika aplikace nebo logika, která se používá napříč kontrolery.

3.2.7 i18n

Cesta: /workManager/grails-app/i18n

Adresář obsahuje soubory, které zajišťují překlady aplikace do cizích jazyků. Snažil jsem se aplikaci psát tak, aby v případě potřeby přeložení aplikace do cizího jazyka nebylo nutné provádět velké úpravy kódu.

3.3 Seznam kontrolerů aplikace

Vytvořená aplikace obsahuje 11 kontrolerů, z nichž 7 kontrolerů řídí stránky pro vytvořené entity, 2 kontrolery obsluhují statické stránky pro nepřihlášeného a přihlášeného uživatele a zbylé 2 kontrolery jsou součástí Spring Security² a spravují přihlašování uživatelů do aplikace.

Vypsany seznam kontrolerů aplikace

Název kontroleru	Popis využití
CustomerController.groovy	Kontroler se stará o logiku stránek souvisejících s entitou Customer.
HomeController.groovy	Kontroler nemá žádnou vnitřní logiku a stará se o přesměrování uživatele na úvodní stránku po přihlášení.
MilestoneController.groovy	Kontroler se stará o logiku stránek souvisejících s entitou Milestone.
ProjectController.groovy	Kontroler se stará o logiku stránek souvisejících s entitou Project a navíc spravuje logiku přidávání a odebrání členů projektu.
PublicController.groovy	Kontroler nemá žádnou vnitřní logiku a stará se o zobrazení úvodní stránky před přihlášením.
TicketController.groovy	Kontroler se stará o logiku stránek souvisejících s entitou Ticket.
UserController.groovy	Kontroler se stará o logiku stránek souvisejících s entitou User. Obsahuje logiku registrace uživatele a změny osobních údajů po přihlášení.
WorkedTimeController.groovy	Kontroler se stará o logiku stránek souvisejících s entitou WorkedTime.
WorkOnProjectController.groovy	Kontroler se stará o logiku stránek souvisejících s entitou WorkOnProject.
LoginController.groovy	Kontroler se stará o přihlašování uživatelů do aplikace a také mimo jiné kontroluje, zda uživatel má oprávnění pro návštěvu zabezpečených stránek. Využívá Spring security.
LogoutController.groovy	Stará se o odhlášení uživatele z aplikace. Využívá Spring security

Tabulka 29 - Seznam kontrolerů aplikace

² Spring Security je Java/Java EE framework, starající se o autentizaci, autorizaci a další bezpečnostní prvky enterprise aplikací.

3.4 Seznam views (webových stránek aplikace)

Views jsou soubory starající se o generování uživatelského rozhraní. Jsou ve složkách, jejichž jméno většinou reprezentuje jméno kontroleru, který je obsluhuje. Uvnitř složek jsou soubory pojmenované <nazev>.gsp. <nazev> je ve většině případů jméno metody v příslušném kontroleru. Pokud je před názvem podtržítka, tak je tento GSP soubor používán jako šablona (template) v jiné stránce. Šablony jsem využíval k vytváření formulářů na stránce pro zvýšení přehlednosti kódu. Navíc se můžou používat ve více stránkách, což je velká výhoda. Views obsahují minimální logiku a pouze přejímají proměnné posílané z kontrolerů a podle nich reagují při vytváření uživatelského rozhraní.

Vypsání seznamu views aplikace a jejich popis

Složka	Název GSP souboru	Popis
customer	_form	Template používaný pro vytváření a editaci zákazníka.
customer	create	Stránka, starající se o vytvoření zákazníka.
customer	edit	Stránka, starající se o editaci zákazníka.
customer	index	Stránka, starající se o vypsání zákazníků přihlášeného uživatele.
customer	show	Stránka, starající se o zobrazení detailu zákazníka.
home	index	Uvodní stránka po přihlášení.
layout	main	Stránka obsahuje prvky, které se zobrazují přihlášenému uživateli na všech stránkách aplikace (Přihlášení/odhlášení, horní menu, patička).
login	auth	Stránka zobrazená při stisku tlačítka přihlásit.
login	denied	Stránka zobrazená, když uživatel nemá práva na navštívení stránky.
milestone	_form	Template používaný pro vytváření a editaci milestonu.
milestone	create	Stránka, starající se o vytvoření milestonu.
milestone	edit	Stránka, starající se o editaci milestonu.
milestone	index	Stránka, zobrazující milestonu projektu.
milestone	show	Stránka, zobrazující detail milestonu
project	_form	Template používaný pro vytváření projektu.
project	_formEdit	Template používaný pro editaci projektu.
project	create	Stránka, starající se o vytvoření projektu.
project	edit	Stránka, starající se o editaci projektu.
project	index	Stránka, starající se o vypsání projektů.
project	listMembers	Stránka, starající se o vypsání členů projektu a jejich přidávání a odebírání.
project	show	Stránka, starající se o zobrazení detailu projektu.

Složka	Název GSP souboru	Popis
public	index	Stránka, starající se o zobrazení zúčastněných projektů přihlášeného uživatele.
ticket	_formCreate	Template používaný pro vytváření ticketu.
ticket	_formEdit	Template používaný pro editaci ticketu.
ticket	create	Stránka, starající se o vytvoření ticketu.
ticket	edit	Stránka, starající se o editaci ticketu.
ticket	index	Stránka, starající se vypsání ticketů projektu.
ticket	show	Stránka, starající se o zobrazení detailu ticketu.
user	_form	Template používaný pro registraci zákazníka.
user	_formEdit	Template používaný pro změny osobních údajů registrovaného uživatele.
user	create	Stránka, starající se o registraci uživatele.
user	edit	Stránka, starající se o editaci osobních údajů přihlášeného uživatele.
user	index	Stránka, používaná pro zobrazení detailu uživatele.
user	show	Stránka, používaná při vývoji k výpisu registrovaných uživatelů.
workedTime	_form	Template používaný pro vytvoření odpracovaného času pro vlastníka projektu.
workedTime	_formCreateAsMember	Template používaný pro vytvoření odpracovaného času pro častníka projektu.
workedTime	create	Stránka, starající se o vytvoření odpracovaného času jakéhokoli účastníka projektu.
workedTime	createAsMember	Stránka, starající se o vytvoření odpracovaného času přihlášeného účastníka projektu.
workedTime	edit	Stránka, starající se o editaci odpracovaného času.
workedTime	index	Stránka, zobrazující celkový vykázaný čas na projektu a umožňující filtrování podle času a účastníka projektu.
/	error	Stránka používaná pro vývoj aplikace, vypisující chybové hlášení

Tabulka 30 - Vypsání seznam views aplikace a jejich popis

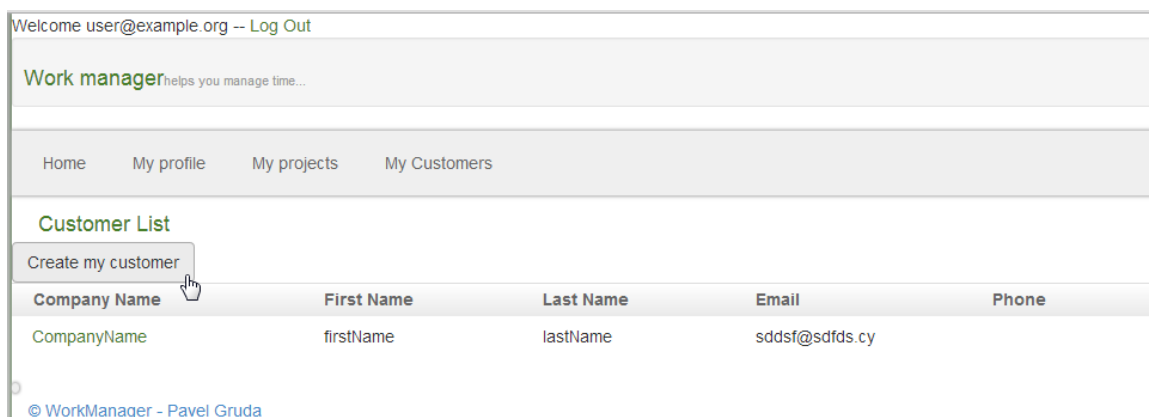
3.5 Uživatelské rozhraní aplikace

U uživatelského rozhraní jsem vycházel z předpokladu, že každý uživatel bude mít rozlišení displeje větší než 1024x728px. Stránky jsou složeny z horní části, těla a patičky. Horní část i patička zůstává pro každou stránku stejná. Dle mého názoru to zlepšuje orientaci a podporuje ucelenost vzhledu. Horní část obsahuje název aplikace a menu. Tělo stránky se mění podle akcí uživatele.

3.6 Ukázka uživatelského rozhraní aplikace

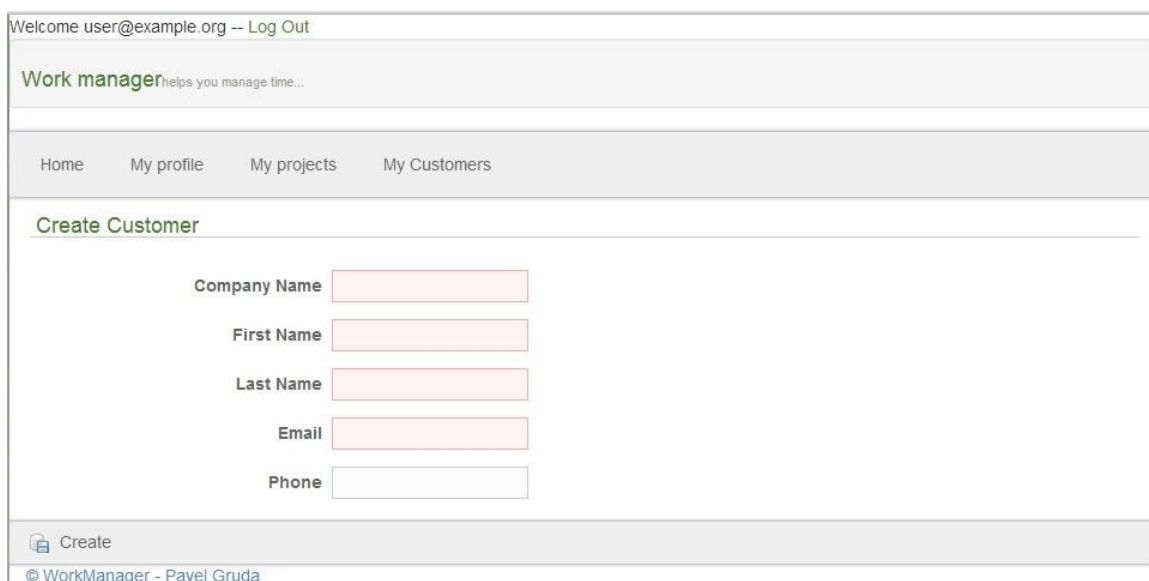
V této části předvedu ukázkou uživatelského rozhraní. Rozhodl jsem se pro průchod případu užití *PU05 - Vytvoření svého zákazníka*.

1) Přihlášený uživatel klikne na tlačítko přidat zákazníka.



Obrázek 8 - GUI - klik na tlačítko přidání zákazníka

2) Systém zobrazí formulář pro přidání zákazníka.

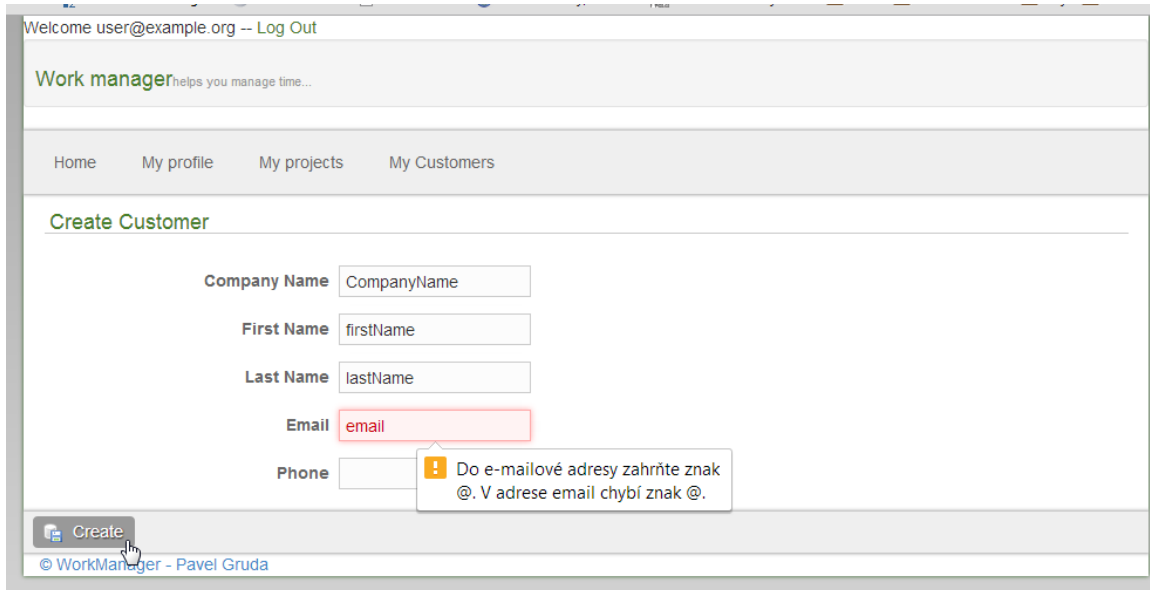


Obrázek 9 - GUI - zobrazení formuláře "Přidání zákazníka"

3) Uživatel vyplní formulář a stiskne tlačítko vytvořit.

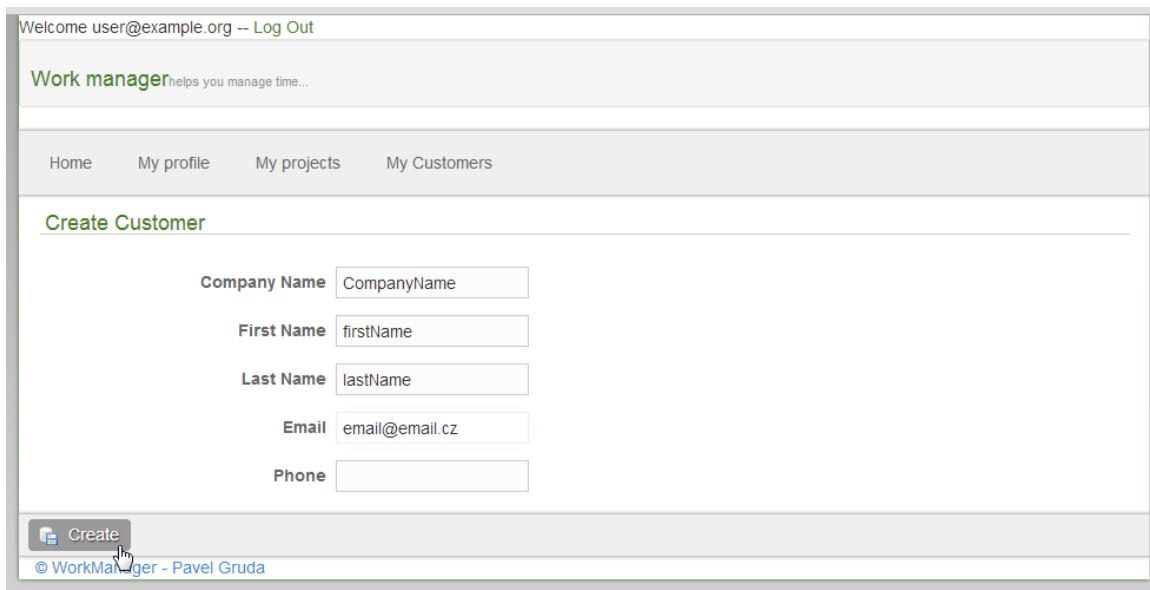
4) Systém zkontroluje, zda jsou vložené údaje ve správném formátu.

- 5) Alternativní tok událostí: V případě, že vyplněné údaje nejsou správné, systém uživatele upozorní.



Obrázek 10 - GUI - upozornění na nesprávně vyplněné údaje

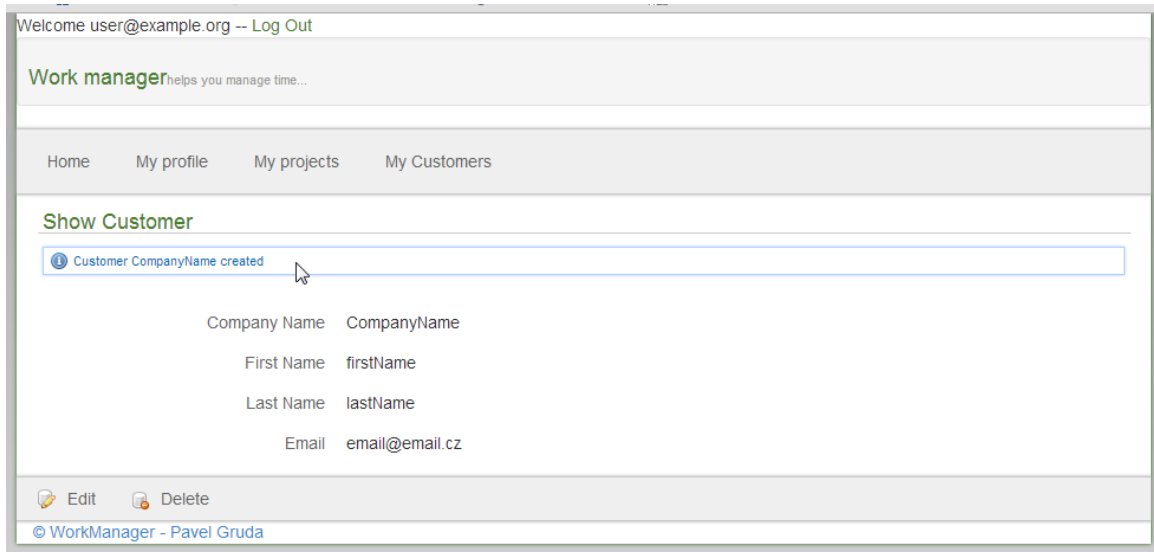
- 3) Uživatel vyplní správně formulář a stiskne tlačítko vytvořit.



Obrázek 11 - GUI - správně vyplněné údaje

- 4) Systém zkontroluje, zda jsou vložené údaje ve správném formátu.
5) Systém přidá zákazníka do databáze.

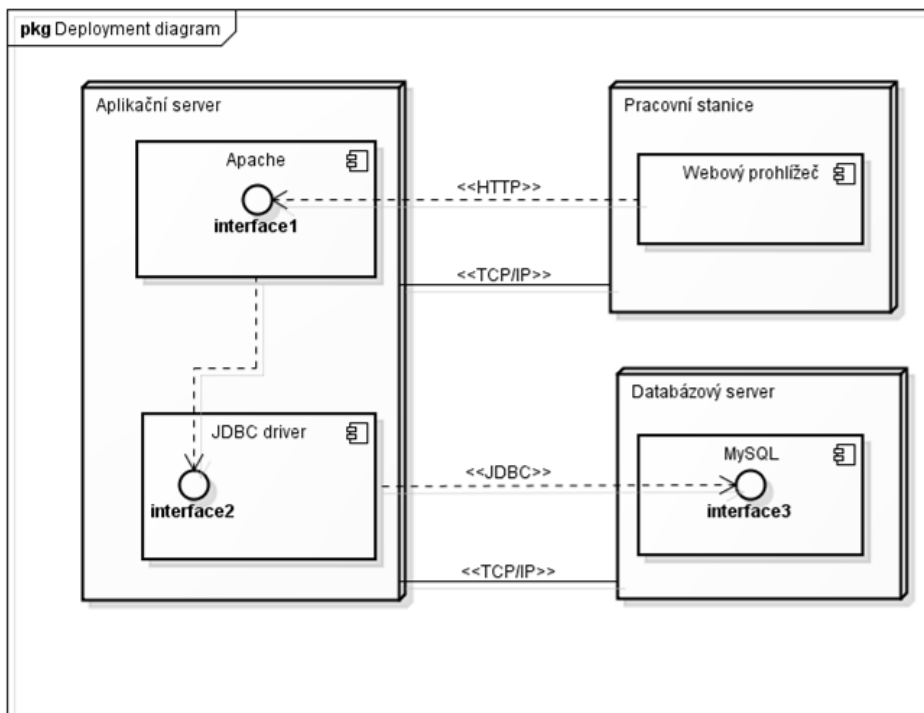
- 6) Systém zobrazí detail vytvořeného zákazníka a vypíše uživateli hlášku o jeho vytvoření.



Obrázek 12 - GUI - Zobrazení detailu vytvořeného zákazníka a vypsání hlášky o vytvoření

3.7 Diagram nasazení

Diagram nasazení zobrazuje rozložení jednotlivých softwarových komponent na hardwarových zařízeních a znázorňuje jejich komunikaci. Používá se pro popis fyzické architektury systému.



Obrázek 13 - Diagram nasazení

4 Testování

4.1 Využití souboru bootstrap při testování

Bootstrap.groovy je soubor umístěný v konfigurační části aplikace a používá se bezprostředně po spuštění aplikace. Při vývoji aplikace umožňuje nahrát testovací záznamy do databáze. Používal jsem ho před testování jednotlivých funkcí programu. Souborem jsem si ověřil správnou funkčnost komponent pracujících s databází (doménové třídy) a správně implementovaný konceptuální datový model do aplikace. V souboru testuji, zda se záznamy do databáze nahrály správně příkazy „assert“. Všechny testy proběhly úspěšně.

4.2 Testování uživatelského rozhraní v různých prohlížečích

Smyslem tohoto testu je ověření funkčnosti vyvinuté aplikace na různých prohlížečích. Zjišťoval jsem, zda jsou komponenty uživatelského rozhraní správně zobrazeny a mají správnou funkčnost.

Seznam testovaných vlastností GUI

- 1) V prohlížeči se nesmí zásadně změnit formátování aplikace tak, aby znemožňovalo přehlednost nebo zakrývalo informace aplikace. Testuji zobrazením domovské stránky po přihlášení.
- 2) V prohlížeči musí správně fungovat ovládací prvky aplikace (tlačítka, vyskakovací okna...). Testuji letným průchodem aplikací.
- 3) Při změně velikosti okna prohlížeče, minimalizaci a maximalizaci musí být okno správně obnoveno a informace v okně dostupné. Testuji změnou velikosti, minimalizací a maximalizací úvodní obrazovky po přihlášení a kontroluji formátování stránky a dostupnost všech informací.

Výstup testu uživatelského rozhraní

Zde můžeme vidět výstup testování různých prohlížečů. Na testování jsem zvolil tři známé prohlížeče: Internet Explorer, Google Chrome, Firefox. Výstup testu může nabývat dvou hodnot: prošel, neprošel.

Prohlížeč	Test 1	Test 2	Test 3
Internet Explorer 11	prošel	prošel	prošel
Google Chrome v.34.0.1847.131m	prošel	prošel	prošel
Firefox v29.0.1	prošel	prošel	prošel

Tabulka 31 - Výstup testu uživatelského rozhraní vůči různým prohlížečům

4.3 Testování programem Selenium

Abych ověřil správnou funkčnost, rozhodl jsem se testovat případy užití přímo na uživatelském rozhraní, protože je to jediné rozhraní, kterým se aplikace používá. K testování jsem zvolil nástroj Selenium[8]. Jedná se o rozšíření webového prohlížeče Firefox. V Selenium jsem vytvořil sadu testovacích případů. Celkem sada obsahuje 32 testovaných postupů průchodu uživatelským rozhraním. Od registrace, založení projektu, založení a editace ticketu, přidání výkazu práce po přidání člena projektu a otestování členských oprávnění. V následující tabulce *Výstup Selenium testů* je výpis všech testovacích případů.

Nevýhodou programu Selenium je ověřování správnosti výstupu – například jsem nemohl otestovat součty odpracovaných hodin. Proto jsem zvolil ještě další způsob testování aplikace, kde budu osobně postupovat podle scénářů případů užití jednotlivých aktérů. I když ověřování výstupu Seleniumem není dokonalé, myslím si, že testy se dají pokládat u tohoto programu za dostačující. Dokážu to následujícím příkladem. V testovacím balíku Selenium jsem vytvořil mimo jiné **test 08 Založení projektu**, **test 09 Editace projektu** a **test 10 Smazání projektu**. V případě, že by neprošel korektně **test 08 Založení projektu**, pak by nutně nemohly projít i následující testy **test 09** a **test 10**, protože k projektu přistupují.

Test Case	
01 Sign in - wrong data	
02 Register new users	
03 Log in	
04 Create customer	
05 Edit customer	
06 Delete customer	
07 Create Customer2	
08 Create project	
09 Edit project	
10 Delete project	
11 Create project2	
12 Create Milestone	
13 Edit milestone	
14 Delete milestone	
15 Create milestone 2	
16 Create ticket	
17 Edit ticket	
18 Delete ticket	
19 Create ticket 2	
20 Add time to ticket	
21 Edit worked time	
22 Remove Worked Time	
23 Add worked time as Owner	
24 report worked time	
25 Add member to project	
26 Delete member	
27 Add member to project 2	
28 Log in as member tested project	
29 Add ticket as member	
30 Add time to ticket as member	
31 Edit time as member	
32 Delete time as member	
Runs:	32
Failures:	0

Obrázek 14 - Výstup Selenium testů

4.4 Testování aplikace podle případů užití

Tento test bude probíhat následujícím způsobem. Budu postupně zkoušet scénáře případů užití jednotlivých aktérů a zjišťovat, zda fungují korektně. K testu použiji tabulku *Přehled oprávnění aktérů systému* a místo informace, zda uživatel má nebo nemá oprávnění, budu přiřazovat informaci, zda test prošel či neprošel. Hodnota N/A znamená, že příslušný případ užití není pro aktéra k dispozici.

Název případu užití	Nepřihlášený uživatel	Přihlášený uživatel	Člen projektu	Vlastník projektu
Registrace nového uživatele a přihlášení	prošel	prošel	prošel	prošel
Přihlášení	prošel	prošel	prošel	prošel
Odhlášení	prošel	prošel	prošel	prošel
Prohlížení seznamu svých a zúčastněných projektů	N/A	prošel	prošel	prošel
Vytvoření svého zákazníka	N/A	prošel	prošel	prošel
Editace svého zákazníka	N/A	prošel	prošel	prošel
Smazání zákazníka	N/A	prošel	prošel	prošel
Založení svého projektu	N/A	prošel	prošel	prošel
Přidání registrovaného uživatele do projektu	N/A	N/A	N/A	prošel
Výpis odpracovaných hodin účastníka projektu v časovém období	N/A	N/A	prošel	prošel
Vykázání práce na svém ticketu	N/A	N/A	prošel	prošel
Zobrazení svých zákazníků	N/A	prošel	prošel	prošel
Změna svých osobních údajů	N/A	prošel	prošel	prošel
Prohlížení seznamu výkazů práce na celém projektu	N/A	N/A	prošel	prošel
Editace své vykázané práce	N/A	N/A	prošel	prošel
Smazání své vykázané práce	N/A	N/A	prošel	prošel
Zobrazení vykázané práce účastníka projektu	N/A	N/A	prošel	prošel
Prohlížení seznamu ticketů projektu	N/A	N/A	prošel	prošel
Založení nového ticketu projektu	N/A	N/A	prošel	prošel
Editace ticketu projektu	N/A	N/A	prošel	prošel
Přiřazení ticketu účastníkovi projektu	N/A	N/A	prošel	prošel
Zobrazení detailu ticketu	N/A	N/A	prošel	prošel
Prohlížení seznamu milníků projektu	N/A	N/A	prošel	prošel
Vytvoření milníku	N/A	N/A	prošel	prošel
Editace milníku	N/A	N/A	prošel	prošel

Název případu užití	Nepřihlášený uživatel	Přihlášený uživatel	Člen projektu	Vlastník projektu
Smazání milníku	N/A	N/A	prošel	prošel
Zobrazení detailu milníku	N/A	N/A	prošel	prošel
Prohlížení seznamu účastníků projektu	N/A	N/A	prošel	prošel
Zobrazení detailu uživatele	N/A	N/A	prošel	prošel
Zobrazení detailu projektu	N/A	N/A	prošel	prošel
Zobrazení detailu zákazníka projektu	N/A	N/A	prošel	prošel
Smazání uživatele z projektu	N/A	N/A	N/A	prošel
Editace vykázaných hodin účastníka projektu	N/A	N/A	N/A	prošel
Přeřazení vykázané práce k jinému ticketu	N/A	N/A	N/A	prošel
Smazání projektu	N/A	N/A	N/A	prošel
Editace projektu	N/A	N/A	N/A	prošel
Změna stavu projektu	N/A	N/A	N/A	prošel

Tabulka 32 - Výstup testování aplikace podle případů užití

5 Závěr

V rámci bakalářské práce jsem provedl analýzu, návrh a vytvořil jsem webovou aplikaci pro management času projektového týmu.

Nejtěžší částí bylo dle mého názoru zjišťovat, jaké technologie budu používat pro pokrytí všech požadavků na aplikaci. Díky tomu jsem se naučil zkombinovat a používat nové technologie. Aplikaci jsem se rozhodl vytvořit ve frameworku Grails, se kterým jsem se již dříve setkal, avšak poprvé jsem prováděl celou implementaci kódu. Po překonání počátečních obtíží se vývoj značně urychlil a ukázala se efektivita frameworku. Framework využívá třívrstvou architekturu aplikace, proto je orientace v projektu přehledná. Přehlednost projektu zvyšuje i jednotnost pojmenování kontrolerů a GSP souborů. Další jeho výhodou je nezávislost na použité databázi.

Hlavním přínosem pro mě bylo absolvování celého procesu od získávání požadavků na aplikaci, analýzy, návrhu po implementaci a testování. Díky tomu jsem měl možnost pochopit, co ovlivní tyto jednotlivé části během vývoje.

S využitím navržených testů jsem ověřil správnou funkčnost programu. V testování jsem se zaměřil na uživatelské rozhraní, protože aplikace bude používána pouze tímto způsobem. Snažil jsem se uživatelské rozhraní vytvořit přehledné a intuitivní, ale zatím jsem neměl možnost realizovat usability testy.

Vytvořená aplikace obsahuje veškerou funkcionalitu popsanou v analýze a je připravená k nasazení na server.

6 Seznam použité literatury

- [1] JEFF SCOTT BROWN, Graeme Rocher a Graeme Keith ROCHER. The definitive guide to Grails 2. New York: Apress, 2013, xv, 343 p. ISBN 978-143-0243-786.

- [2] GRAILS by Pivotal [online]. Citováno dne 21.5.2014.
Dostupné z: <https://grails.org/>

- [3] XAMPP Apache + MySQL + PHP + Perl [online]. Citováno dne 21.5.2014.
Dostupné z: <https://www.apachefriends.org/index.html>

- [4] Astah.net: UML and Modeling Tools [online]. Citováno dne 21.5.2014.
Dostupné z: <http://astah.net/>

- [5] Bootstrap [online]. Citováno dne 21.5.2014.
Dostupné z: <http://getbootstrap.com/>

- [6] IntelliJ IDEA — The Best Java and Polyglot IDE - JetBrains [online].
Citováno dne 21.5.2014.
Dostupné z: www.jetbrains.com/idea/

- [7] Stack Overflow [online]. Citováno dne 21.5.2014.
Dostupné z: <http://stackoverflow.com/>

- [8] Selenium - Web Browser Automation [online]. Citováno dne 21.5.2014.
Dostupné z: <http://docs.seleniumhq.org/>

- [9] Wikipedie, otevřená encyklopedie [online]. Citováno dne 21.5.2014.
Dostupné z: <https://cs.wikipedia.org/>