

České vysoké učení technické

Fakulta elektrotechnická

Katedra počítačů



Bakalářská práce

Online rezervační systém pro laboratoř očních pohybů

Tomáš Kolařík

Vedoucí práce: Ing. Dobiáš Martin, Ph.D.

Studijní program: Softwarové technologie a management

Obor: Weby a multimédia

Květen 2015

Poděkování

Tímto bych chtěl poděkovat svému vedoucímu práce k věcným připomínkám, radám k řešení mé bakalářské práce a velké trpělivosti při tvorbě aplikace. Také bych chtěl poděkovat svým studijním kolegům a mému blízkému okolí za podporu.

Prohlášení

Prohlašuji, že jsem semestrální projekt vypracoval samostatně a použil jsem pouze podklady zde uvedené. Nemám závažný důvod proti užití tohoto díla ve smyslu §60 Zákona c.121/2000 Sb., o právu autorském a o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Praze, dne 13. 12. 2014

.....

Abstract

This work deals with analysis of requirements for reservation system for Eye movement laboratory. After the analysis, all the completed and available systems where checked and none of them covered all the requirements. That was the reason for implementing the system by myself. All the chosen technologies where compared and the Play framework was chosen for implementation. The report elaborates the implementation approach and the final part describes the testing procedures. The documentation for project is written in UML.

Abstrakt

Tato práce se zabývá analýzou požadavků na rezervační systém pro Laboratoř očních pohybů. Po analýze požadavků byla provedena kontrola všech již existujících řešení a bylo zjištěno, že neexistuje systém, který by plně vyhovoval všem požadavkům. Proto byla zvolena vlastní implementace. Byly porovnány všechny vybrané technologie a byla vybrán framework Play. V něm pak byla aplikace implementována. V práci je popis implementace její testování. K popisu aplikace je využíván jazyk UML.

České vysoké učení technické v Praze
Fakulta elektrotechnická

katedra počítačové grafiky a interakce

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: **Tomáš Kolařík**

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

Název tématu: **Online rezervační systém pro laboratoř očních pohybů**

Pokyny pro vypracování:

- Proveďte analýzu funkčních požadavků na online rezervační systém pro laboratoř očních pohybů. Řešení musí obsahovat nejen rezervaci laboratoře, ale také administraci experimentů, které mohou trvat různou dobu, i možnost přihlásit se na měření jako účastník.
- Analyzujte technologie vhodné pro online rezervační systémy.
- Implementujte a nasadte SW řešení na server katedry ekonomiky.
- Proveďte usability testy SW aplikace.

Seznam odborné literatury:

Rubin, J. - Chrisnell, D. - et al.: Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests. Wiley, 2008. ISBN 978-0470185483.
CDonnolly, T. - Begg, C.: Database Systems: A Practical Approach to Design, Implementation, and Management (6th Edition). Addison-Wesley, 2014. ISBN 978-0132943260.

Vedoucí: Ing. Martin Dobiáš, Ph.D.

Platnost zadání: do konce letního semestru 2015/2016

prof. Ing. Jiří Žára, CSc.
vedoucí katedry

prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 4. 11. 2014

Obsah

1	Úvod	1
1.1	Cíl práce	1
1.2	Zainteresované osoby	1
1.3	Zadání	1
1.4	Popis současného systému.....	1
1.5	Současný průběh experimentu.....	1
1.6	Budoucí průběh experimentu	2
2	Požadavky na systém.....	2
2.1	Funkční požadavky	2
2.2	Nefunkční požadavky	3
3	Uživatelské role v systému	3
3.1	Účastník.....	4
3.2	Pořadatel	4
3.3	Administrátor	6
4	Procesy v rezervačním systému	6
4.1	Procesy účastníka	6
4.2	Procesy pořadatele.....	8
4.3	Administrátor	11
5	Řešení třetích stran	12
5.1	Reservio.com	12
5.2	Orario.....	13
5.3	Doodle	15
5.4	Erezervuj.cz.....	16
5.5	Bookappo.....	18
5.6	Planyo	19
5.7	Srovnání řešení třetích stran	21
6	Vlastní řešení	22
6.1	PHP	22
6.2	Java	22
6.3	Frameworky.....	22
7	Srovnání frameworků.....	24
7.1	Výsledek srovnání.....	25
7.2	Framework Play	25
8	Návrh a implementace	27
8.1	Struktura projektu	27

9	Vytváření časových návrhů.....	32
9.1	Jeden pořadatel	32
9.2	Dva pořadatelé	34
9.3	Tři a více pořadatelů.....	35
10	Popis aplikace	36
11	Testování	38
11.1	Unit Testy	38
11.2	Selenium testy	38
12	Závěr	39
13	Seznam zkratk a pojmů	40
14	Použité zdroje	41

Obrázky

Obrázek 1: Uživatelské role, zdroj: vlastní práce	3
Obrázek 2: Případy užití pro roli účastník, zdroj: vlastní práce	4
Obrázek 3: Případy užití pro roli pořadatel, zdroj: vlastní práce.....	5
Obrázek 4: Případy užití pro roli administrátor, zdroj: vlastní práce	6
Obrázek 5: Diagram aktivity přihlášení se uživatele do systému, zdroj: vlastní práce.....	6
Obrázek 6: Diagram aktivity Registrace uživatele, zdroj: vlastní práce.....	7
Obrázek 7: diagram aktivity Přihlášení se na událost, zdroj: vlastní práce	7
Obrázek 8: Diagram aktivity Odhlášení se uživatele z události	8
Obrázek 9: Diagram aktivity Vytvoření nové události, zdroj: vlastní práce	8
Obrázek 10: Diagram aktivity Editace události, zdroj: vlastní práce	9
Obrázek 11: Diagram aktivity Odstranění události, zdroj: vlastní práce	9
Obrázek 12: Diagram aktivity Vyhledání účastníků, zdroj: vlastní práce	9
Obrázek 13: Diagram aktivity přidání pořadatele k událost, zdroj: vlastní práce	10
Obrázek 14: Odstranění časových návrhů, zdroj: vlastní práce	10
Obrázek 15: Diagram aktivity Vytvoření časového návrhu, zdroj: vlastní práce.....	11
Obrázek 16: Diagram aktivity Přiřazení pořadatelské role, zdroj: vlastní práce	11
Obrázek 17: Uživatelské prostředí služby Reservario, zdroj: [24]	12
Obrázek 18: Uživatelské prostředí služby Orario, zdroj: [3].....	14
Obrázek 19: Uživatelské prostředí služby Doodle, zdroj: [25].....	15
Obrázek 20: Uživatelské prostředí služby Erezervuj.cz, zdroj: [6]	17
Obrázek 21: Uživatelské prostředí služby Bookappo, zdroj: [26].....	18
Obrázek 22: Uživatelské prostředí služby Planyo, zdroj: [11]	20
Obrázek 23: ukázka prostředí nástroje Selenium, zdroj: [27]	26
Obrázek 24: ukázka programovacího jazyka Scala, zdroj: vlastní práce.....	27
Obrázek 25: seznam všech controllerů, zdroj: vlastní práce.....	28
Obrázek 26: Seznam všech modelů, zdroj: vlastní práce	29
Obrázek 27: doménový model aplikace, zdroj: vlastní práce.....	29
Obrázek 28: Seznam všech šablon, zdroj: vlastní práce.....	30
Obrázek 29: ukázka pohledu vytvoreniUdalosti, zdroj: vlastní práce	31
Obrázek 30: Ukázka pohledu detailUdalosti, zdroj: vlastní práce.....	31
Obrázek 31: Délka okna pro 1 pořadatele - měření, zdroj: vlastní práce.....	33
Obrázek 32: Délka okna pro 1 pořadatele – měření a dotazování před měřením, zdroj: vlastní práce	33
Obrázek 33: Délka okna pro 1 pořadatele – měření a dotazování po měření, zdroj: vlastní práce	33
Obrázek 34: Délka okna pro 1 pořadatele – měření a dotaz. před a po měření, zdroj: vlastní práce ..	33
Obrázek 35: Délka okna pro 2 pořadatele – měření a dotazování před měřením, zdroj: vlastní práce	34
Obrázek 36: Délka okna pro 2 pořadatele – měření a dotazování po měření, zdroj: vlastní práce.....	34
Obrázek 37: Délka okna pro 2 pořadatele – měření a dotaz. před a po měření, zdroj: vlastní práce ..	34
Obrázek 38: Délka okna pro 2 pořadatele – měření a dotaz. před a po měření, zdroj: vlastní práce ..	35
Obrázek 39: Délka okna pro 2 pořadatele – měření a dotaz. před a po měření, zdroj: vlastní práce ..	35
Obrázek 40: Délka okna pro 3 pořadatele – měření a dotaz. před a po měření, zdroj: vlastní práce ..	36
Obrázek 41: Hlavní menu aplikace, zdroj: vlastní práce.....	36
Obrázek 42: Vytvoření události a časového návrhu, zdroj: vlastní práce	37
Obrázek 43: Moje události a Editace události, zdroj: vlastní práce	37
Obrázek 44: Kalendář a detail událostí, zdroj: vlastní práce	38

1 Úvod

1.1 Cíl práce

Cílem projektu je vytvořit informační systém pro laboratoř očních pohybů, která dosud žádný informační systém nemá. Systém má ulehčit práci vytváření experimentů, má umožnit jejich evidenci a zamezit problémům s obsazeností jednotlivých laboratoří.

V práci se v kapitolách 1 a 2 zabývám analýzou požadavků na takový systém, v kapitolách 3 a 4 rolemi a procesy probíhajícími v takovém systému. V kapitolách 5-7 se zabývám analýzou již existujících řešení a možností vlastního řešení a následně tyto řešení porovnávám. V kapitolách 8-9 se zabývám popisem implantace a detailům kolem tohoto tématu. V kapitole 10 je pak popis již samotného existujícího systému. Kapitola 11 je pak věnovaná tématu testování.

1.2 Zainteresoované osoby

- Autor – Tomáš Kolařík
- Zadavatel – Ing. Martin Dobiáš, PhD.
- Konzultant/kontrolor – Ing. Martin Dobiáš, PhD.

1.3 Zadání

- Proveďte analýzu funkčních požadavků na online rezervační systém pro laboratoř očních pohybů. Řešení musí obsahovat nejen rezervaci laboratoře, ale také administraci experimentů, které mohou trvat různou dobu, i možnost přihlásit se na měření jako účastník.
- Analyzujte technologie vhodné pro online rezervační systémy.

1.4 Popis současného systému

V současnosti se stav experimentů udržuje pouze ve službách třetích stran (např. doodle.com a další). Každý experiment se zde manuálně vytváří, účastníkům experimentu je pak rozesílán odkaz na experiment, kde vyplní své jméno a mohou se zde přihlásit na konkrétní čas. Neeviduje se zde ale obsazenost místností, popisy experimentů a další věci. Obsazenost laboratoří a stavy uživatelů tedy nejsou k dispozici a musí se řešit osobně, pomocí telefonátů, emailů a podobně. Také nikde není k dispozici soupis všech probíhajících, minulých a budoucích experimentů.

1.5 Současný průběh experimentu

V současnosti vyřizuje zadávání experimentů M. Dobiáš. Pokud chce pořadatel uspořádat experiment, kontaktuje emailem, telefonem apod. pana M. Dobiáše.

Ten musí manuálně ověřit, zda je místo, kde se má experiment konat volné, a to pak zarezervovat. V systému doodle.com poté vytvoří událost, na kterou se mohou účastníci přes systém doodle hlásit.

V události, která je účastníkům manuálně rozesílána emaily, vyplní své jméno a vyberou časové okno, na které se chtějí přihlásit.

Největším problémem je právě nutnost zásahu třetí osoby, pokud je potřeba uspořádat nějaký experiment. Není evidence obsazenosti laboratoří, přehled o konaných experimentech, jejich údajích a přihlášených lidech.

1.6 Budoucí průběh experimentu

Pořadatel experimentu v systému vytvoří novou událost - vyplní detaily o události (název, popis, počet pořadatelů atd.). Vybere volné místo a čas konání. Systém následně vygeneruje možnosti konání události v určený čas. Poté se na tuto událost budou moci hlásit účastníci, kteří si vyberou časové okno, na které se chtějí přihlásit a přihlásí se na událost.

Nebude tak potřeba řešit obsazenost místností s třetí osobu, stejně tak nebude nutno rozesílat odkazy na události poštou - budou všechny pohromadě na jednom místě, kde si je může účastník snadno nalézt sám.

2 Požadavky na systém

2.1 Funkční požadavky

Systém bude umožňovat přihlášení a registraci uživatelů. Bude obsahovat několik uživatelských rolí. Přihlášený uživatel s rolí pořadatel bude moci vytvářet nové události. Jím bude moci nastavit popis, místo konání, čas konání atd. Systém pak pořadateli vygeneruje časový návrh pořádání experimentu (délka jednoho okna, na které se bude hlásit účastník). Ten si pak vybere buď navrhované časy nebo zvolí vlastní časový návrh.

Vlastní události může pořadatel editovat (měnit čas, místnost apod.) nebo smazat. Účastníci se mohou přihlásit na libovolný experiment, kde si každý může vybrat požadovaný čas, kdy se bude experimentu účastnit. Z události se může účastník také odhlásit. Jednotlivé požadavky budou v další kapitole podrobně popsány.

- F1. Systém bude podporovat několik uživatelských rolí
- F2. Systém bude umožňovat vytváření událostí a jejich editaci a mazání
- F3. Systém bude umožňovat vygenerování automatického časového návrh experimentu.
- F4. Účastník se bude moci přihlašovat na vytvořené události
- F5. Pořadatel bude moci upravovat a mazat pouze své události
- F6. Uživatel bude mít přidělenou jednu uživatelskou roli
- F7. Bude existovat administrátorský účet mající neomezená správcovská práva

F8. Systém bude dostupný zdarma

F9. Systém bude schopen obsloužit minimálně 500 rezervací denně

F10. Neomezený počet uživatelských účtů a událostí k rezervaci

F11. Systém umožňuje ovládání pomocí klávesnice a myši. Nebude speciálně uzpůsoben na ovládání dotykem.

2.2 Nefunkční požadavky

N1. Služba bude řešena jako webová aplikace.

N2. V případě vlastní implementace bude veškerá použitá technologie zdarma pro komerční použití.

N3. Aplikace bude podporovat nejpoužívanější moderní webové prohlížeče – Firefox, Internet Explorer a Opera

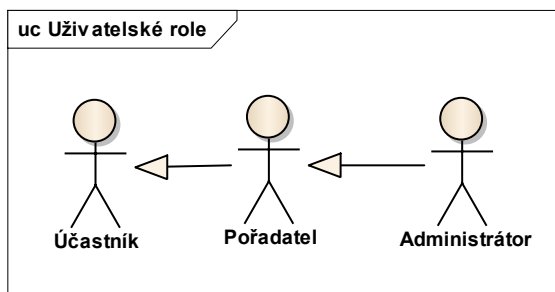
N4. Bude funkční 24 hodin denně

N5. Systém bude možné provozovat na běžném osobním počítači s minimální specifikací:

- 1) CPU – 1 GHz
- 2) RAM – 1 GB
- 3) Připojení k internetu
- 4) Verze Javy 1.8 a vyšší
- 5) Databáze podporující přístup pomocí JDBC

3 Uživatelské role v systému

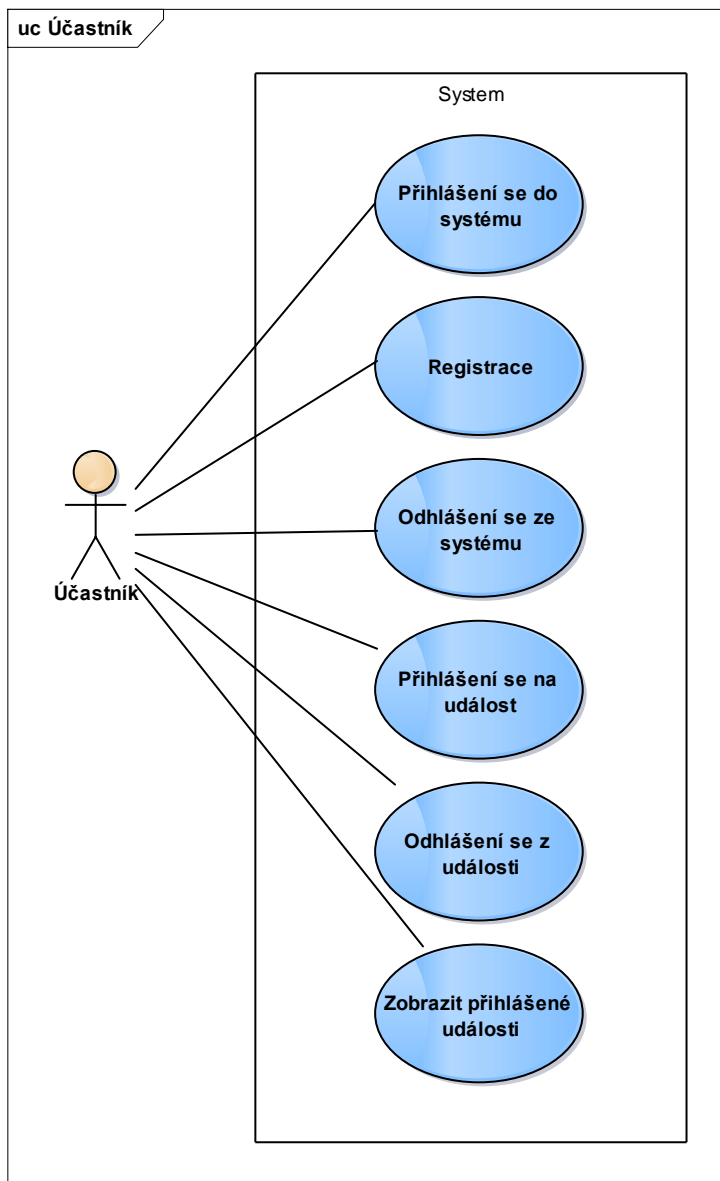
V systému jsou 3 uživatelské role – účastník, pořadatel a administrátor. Účastník má nejnižší pravomoci, administrátor nejvyšší. Pořadatel dědí pravomoci účastníka a administrátor dědí pravomoci pořadatele.



Obrázek 1: Uživatelské role, zdroj: vlastní práce

3.1 Účastník

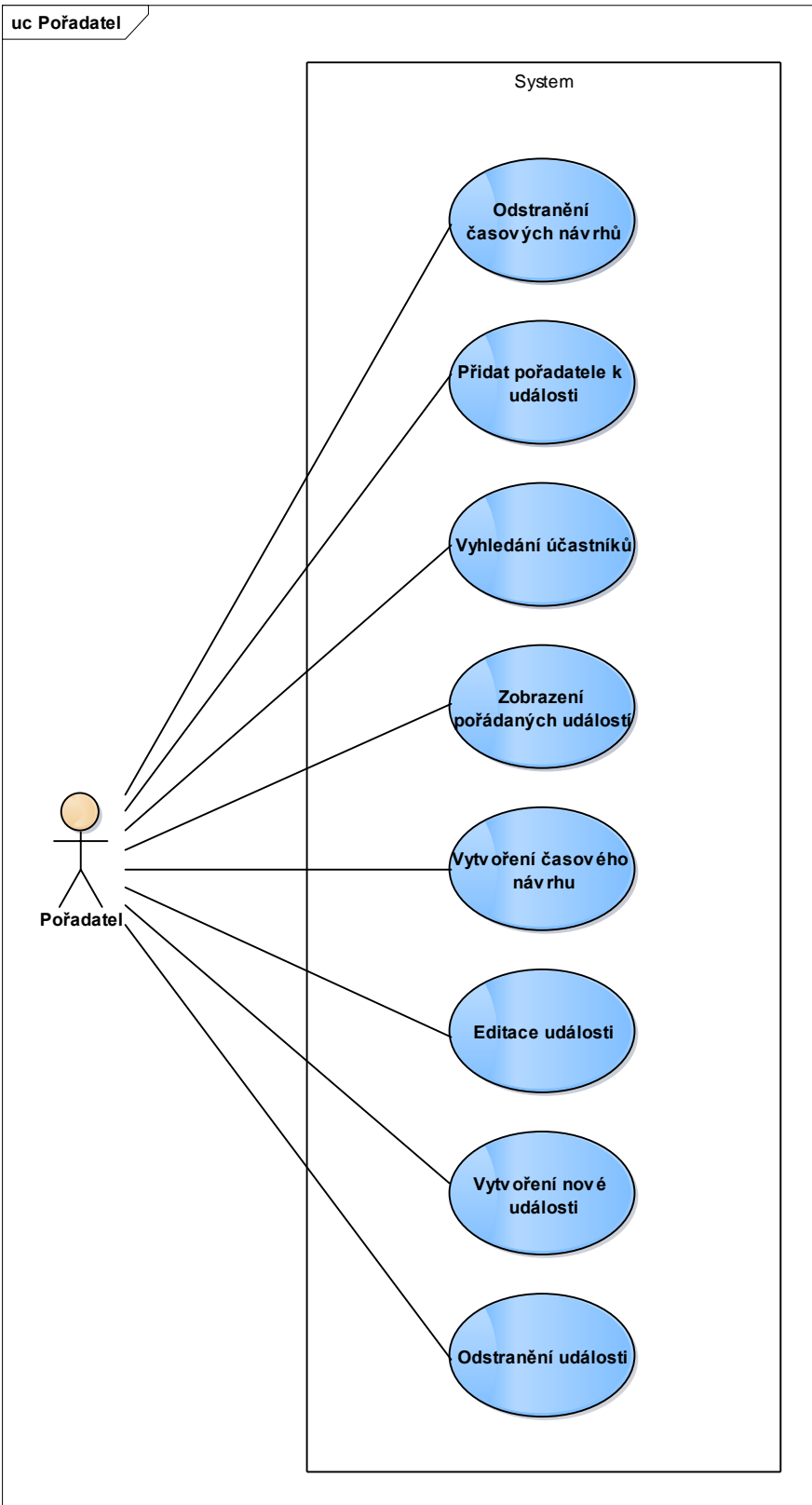
Účastník má v systému nejmenší pravomoci. Má právo hlásit se na již vytvořené události a rušit svou účast na událostech. Události může také vyhledávat.



Obrázek 2: Případy užití pro roli účastník, zdroj: vlastní práce

3.2 Pořadatel

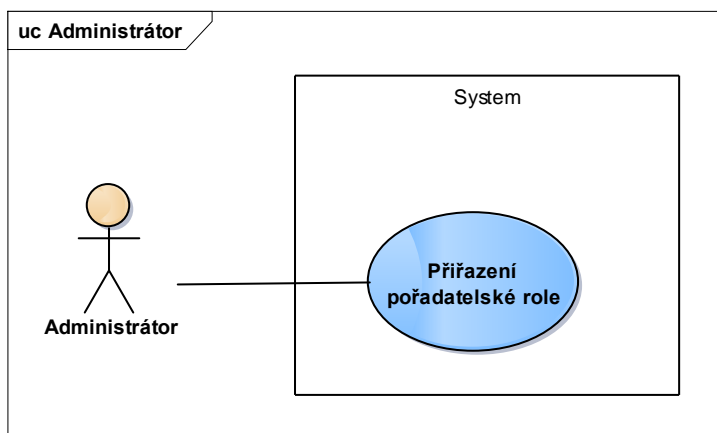
Pořadatel může vytvářet nové události. Svě vytvořené události může také mazat a editovat. K vytvořeným událostem může také přidávat časové návrhy, které může také mazat a upravovat a přidávat k událostem pořadatele. Může také zobrazit pořádané události. K tomu má navíc ty samé pravomoci, jako účastník.



Obrázek 3: Případy užití pro roli pořadatel, zdroj: vlastní práce

3.3 Administrátor

Administrátor má právo přiřazovat jednotlivým účtům pořadatelská práva. Administrátor má v systému nejvyšší pravomoci.



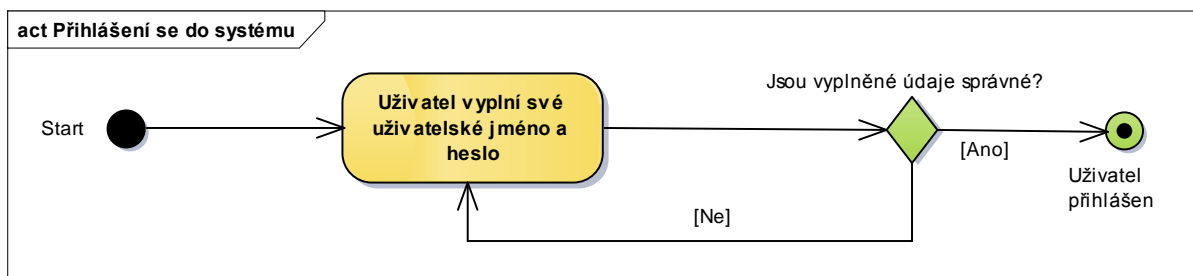
Obrázek 4: Případy užití pro roli administrátor, zdroj: vlastní práce

4 Procesy v rezervačním systému

Níže jsou popsány jednotlivé procesy, které se odehrávají v systému. Jsou zde popsány pouze procesy, které jsou pokryty systémem.

4.1 Procesy účastníka

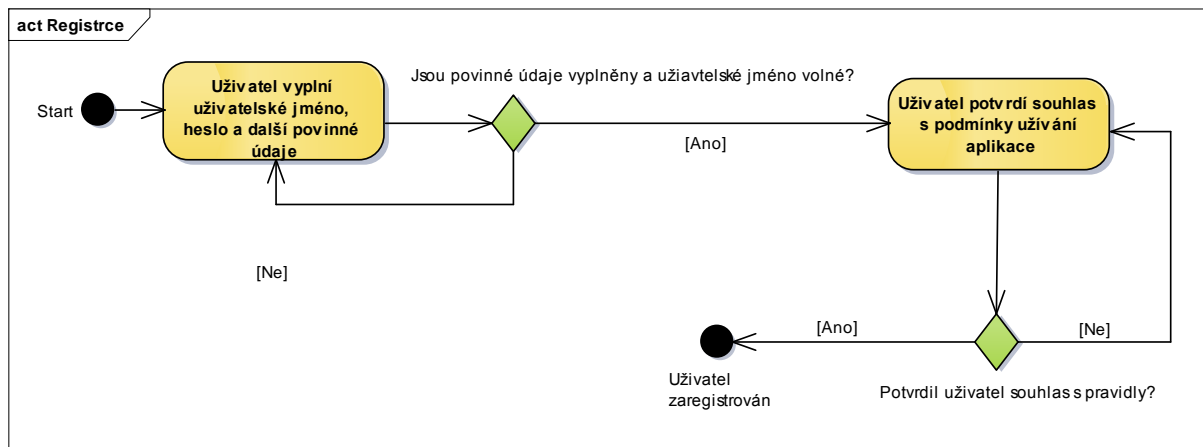
4.1.1 Přihlášení se do systému



Obrázek 5: Diagram aktivity přihlášení se uživatele do systému, zdroj: vlastní práce

4.1.2 Registrace

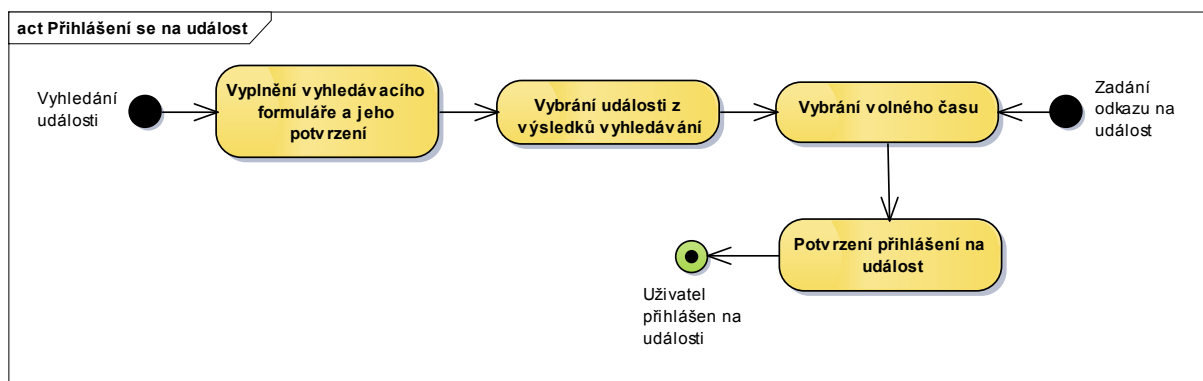
Uživatel vyplní unikátní uživatelské jméno, heslo a další povinné, případně nepovinné údaje a potvrdí registraci.



Obrázek 6: Diagram aktivity Registrace uživatele, zdroj: vlastní práce

4.1.3 Přihlášení se na událost

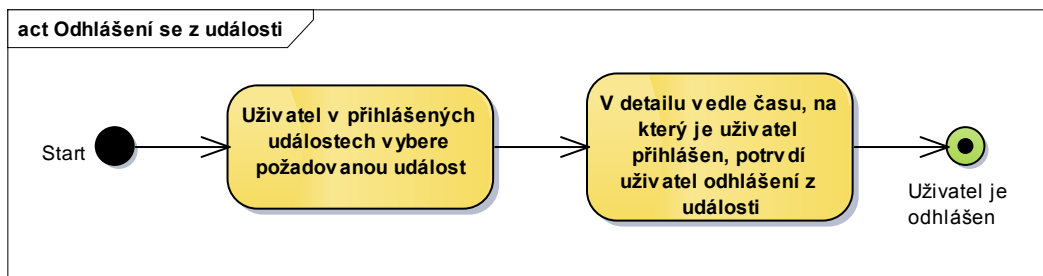
Uživatel buď vyhledá událost pomocí vyhledávacího formuláře, nebo pomocí již přijatého odkazu na událost. Na detailu události vybere příslušný čas, na který se chce přihlásit a akci potvrdí.



Obrázek 7: diagram aktivity Přihlášení se na událost, zdroj: vlastní práce

4.1.4 Odhlášení se z události

Uživatel v seznamu svých přihlášených událostí vybere požadovanou událost a vedle času na který je přihlášen potvrdí odhlášení.

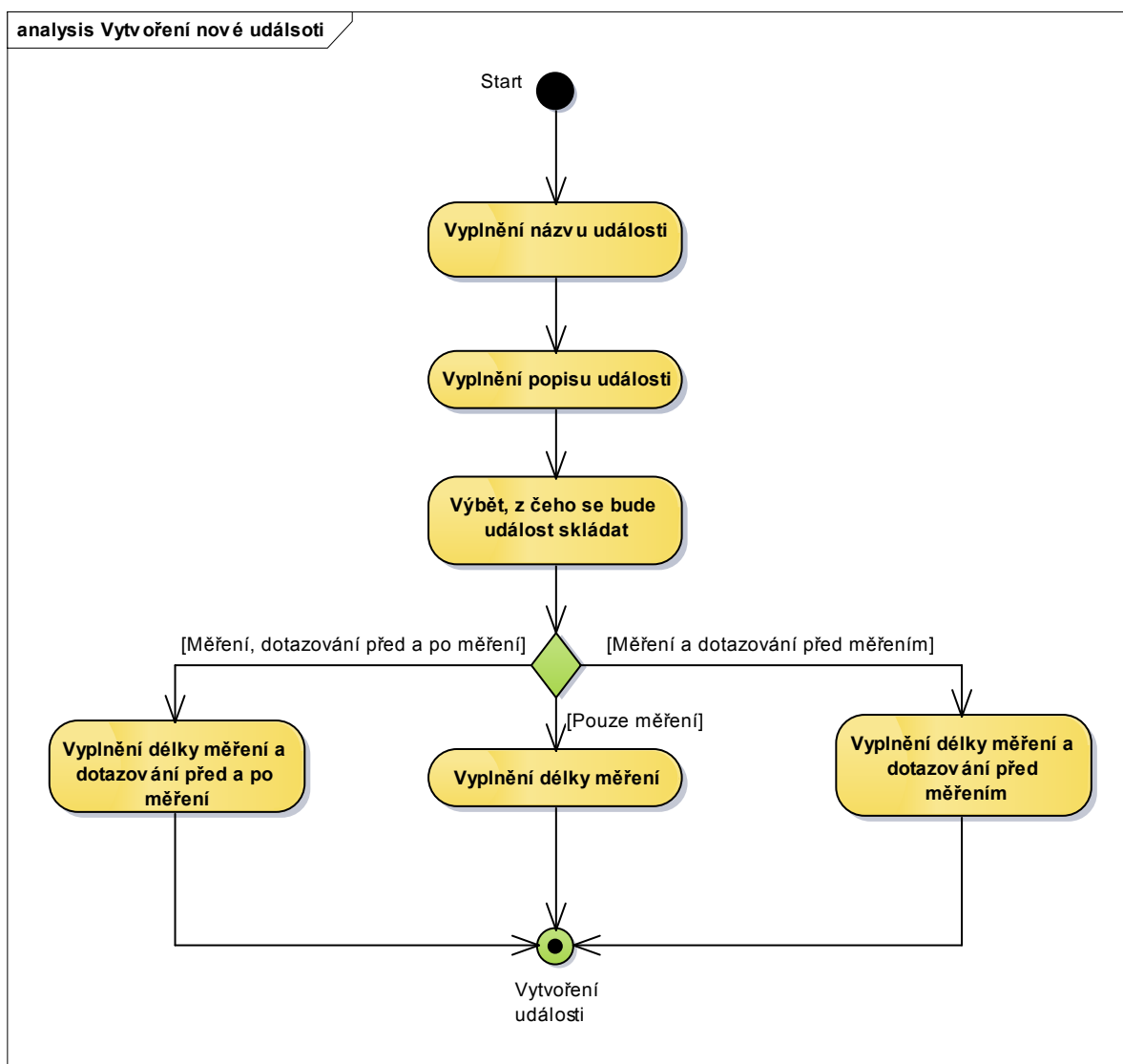


Obrázek 8: Diagram aktivity Odhlášení se uživatele z události

4.2 Procesy pořadatele

4.2.1 Vytvoření nové události

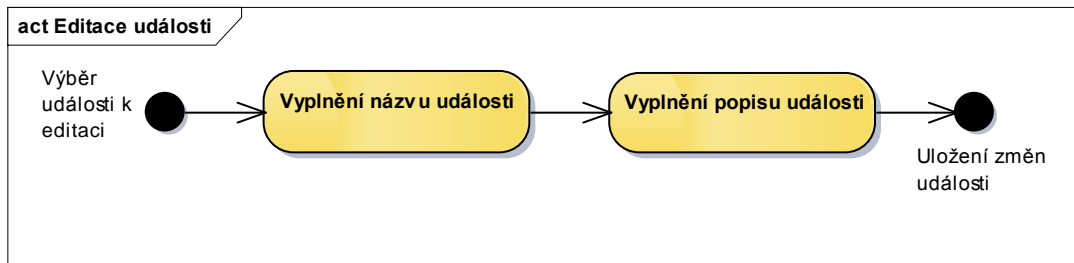
Pořadatel nebo administrátor vyplní název a popis události. Vyplní, zda se událost bude skládat z měření, dotazování před experimentem a po něm nebo jenom některé z nich a vytvoří událost.



Obrázek 9: Diagram aktivity Vytvoření nové události, zdroj: vlastní práce

4.2.2 Editace události

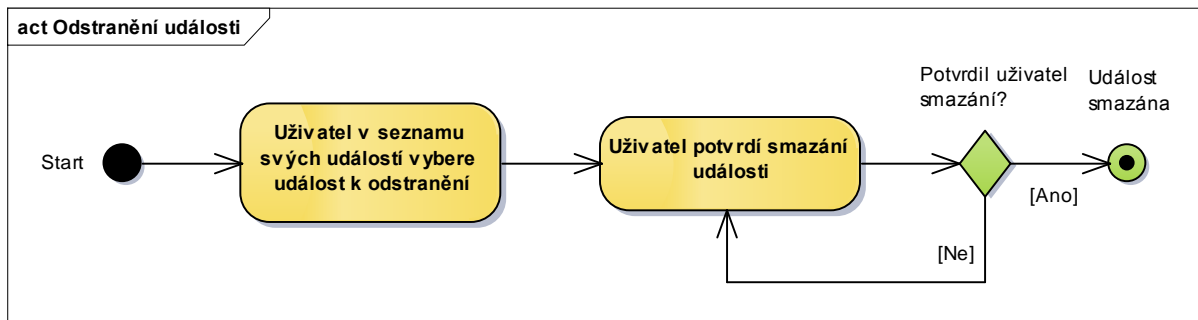
Uživatel, co událost vytvořil, vybere událost, kterou chce editovat. Vyplní upravený název a popis a událost uloží.



Obrázek 10: Diagram aktivity Editace události, zdroj: vlastní práce

4.2.3 Odstranění události

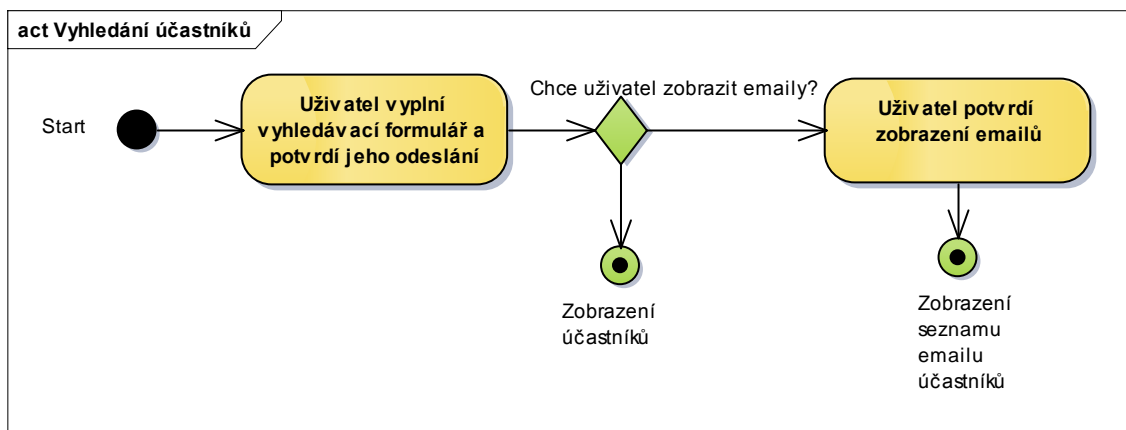
Uživatel, který událost vytvořil, vybere událost, kterou chce odstranit. Po potvrzení se událost smaže.



Obrázek 11: Diagram aktivity Odstranění události, zdroj: vlastní práce

4.2.4 Vyhledání účastníků

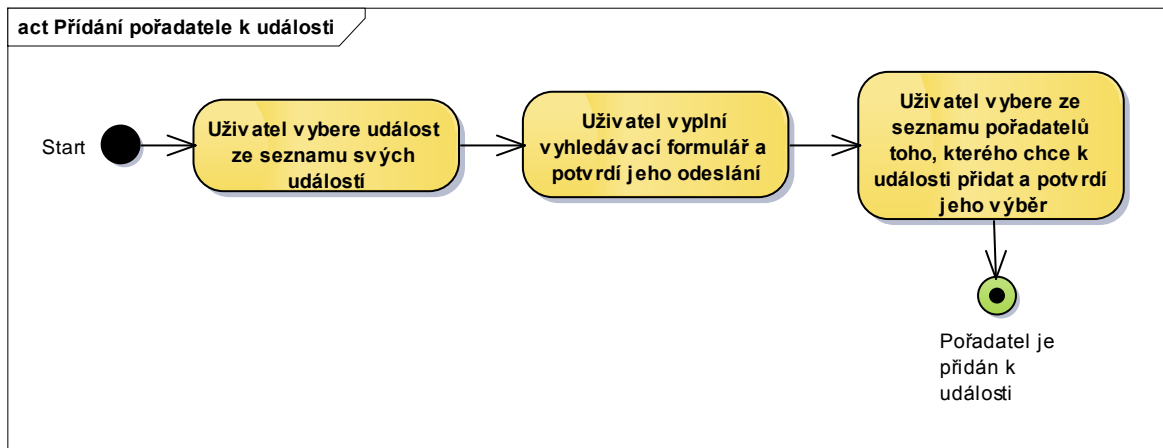
Pořadatel vyplní vyhledávací formulář a potvrdí vyhledávání. Systém vyhledá uživatele podle filtru vytvořeného z parametrů ve formuláři. Pořadatel poté může zobrazit jejich emailové adresy.



Obrázek 12: Diagram aktivity Vyhledání účastníků, zdroj: vlastní práce

4.2.5 Přidání pořadatele k události

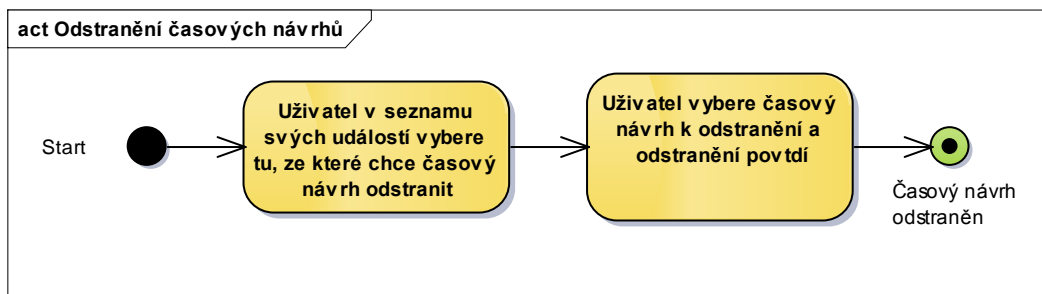
Pořadatel vybere v seznamu svých událostí tu, ke které chce pořadatele přidat. Vyplní vyhledávací formulář a po jeho potvrzení systém zobrazí seznam pořadatelů, které lze přidat k události. Uživatel vybere pořadatele, kterého chce přidat a potvrdí akci.



Obrázek 13: Diagram aktivity přidání pořadatele k události, zdroj: vlastní práce

4.2.6 Odstranění časových návrhů

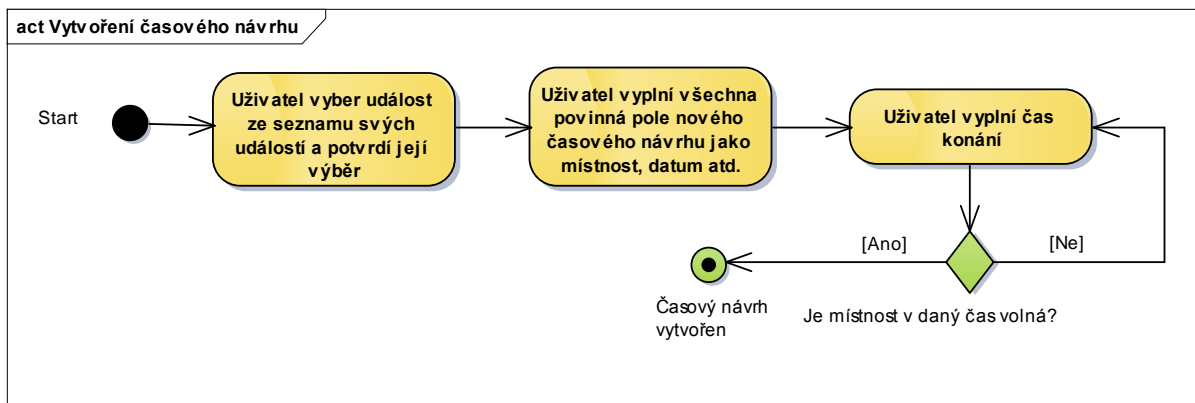
Uživatel vybere událost ze seznamu svých událostí, ze které chce časový návrh odstranit. Uživatel vybere časový návrh, který chce odstranit, potvrdí jeho odstranění a časový návrh je smazán.



Obrázek 14: Odstranění časových návrhů, zdroj: vlastní práce

4.2.7 Vytvoření časového návrhu

Uživatel vybere událost v seznamu svých událostí, ke které chce časový návrh přidat. Vyplní Povinná pole formuláře jako datum konání, čas, místnost a podobně. Pokud je místnost na tento čas volná, tak se po potvrzení vytvoří časový návrh.

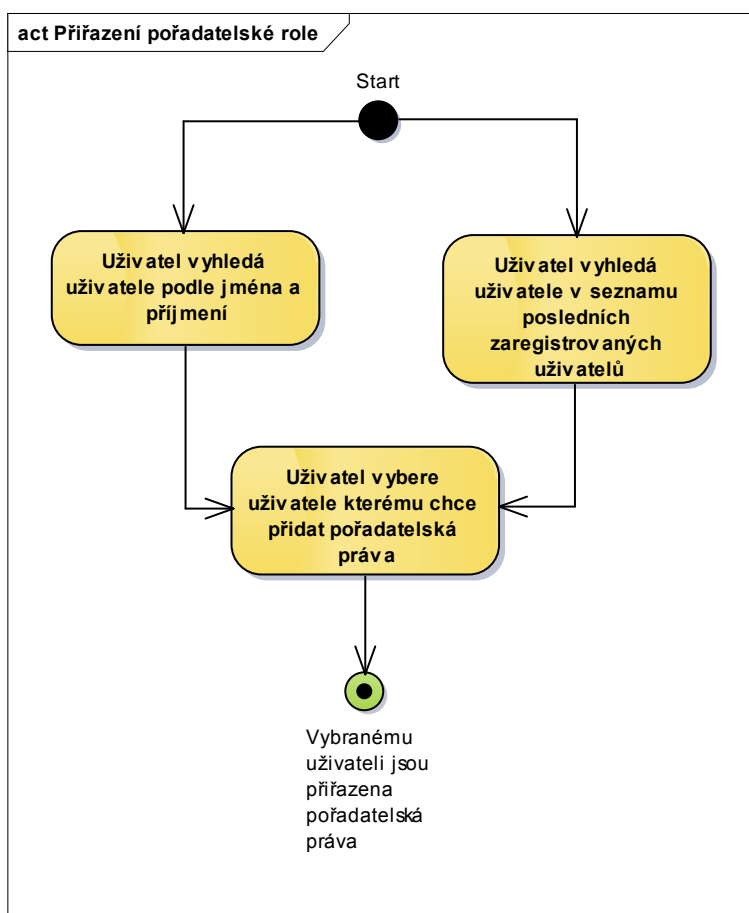


Obrázek 15: Diagram aktivity Vytvoření časového návrhu, zdroj: vlastní práce

4.3 Administrátor

4.3.1 Změna uživatelské role

Administrátor vyhledá uživatelský účet. U toho pak bude moci nastavit jednu ze tří uživatelských rolí v systému.



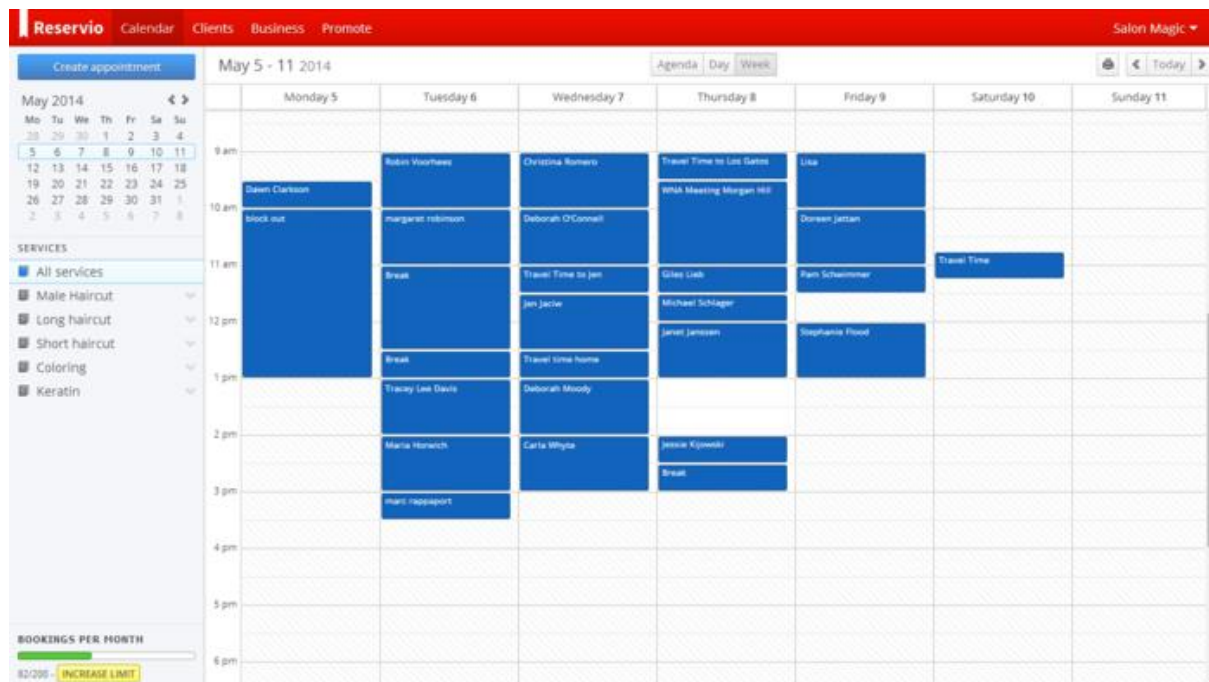
Obrázek 16: Diagram aktivity Přiřazení pořadatelské role, zdroj: vlastní práce

5 Řešení třetích stran

5.1 Reservio.com



www.reservio.com



Obrázek 17: Uživatelské prostředí služby Reservario, zdroj: [24]

Reservio.com je mladá služba, provozující univerzální rezervační systém, zaměřená především na drobné živnostníky.

Funguje pouze jako webová služba, není proto potřeba ji nasazovat na vlastní infrastrukturu. Má specificky navržené prostředí pro konkrétní podporované profese jako lékař nebo právník. Podporováno je přibližně 30 druhů profesí. Pokud profese chybí, je zde i univerzální prostředí, které by mělo vyhovět většině zákazníků. Nenabízí však možnost úpravy těchto prostředí a nelze si tedy upravit funkčnost systému pro svou potřebu. [1]

5.1.1 Cena

Základní verze služby je zdarma, je však limitována nízkým počtem uživatelů (100), pouze 40 registracemi za měsíc a pouze základní funkcí.

Za 200 Kč/měsíc dostane klient navíc neomezený počet klientů, 200 rezervací měsíčně, možnost neomezeného prohlížení historie, statistiky o rezervacích, sms upozornění a další funkce jako prémiovou podporu.

Za 800 Kč/měsíc je jak počet klientů, tak počet rezervací neomezen a uživatel má přístup ke všem funkcím této služby. [2]

5.1.2 Výhody

- Nulový čas potřebný k nasazení
- Již hotové a funkční řešení
- V omezené funkčnosti zdarma

5.1.3 Nevýhody

- Pouze základní a omezené funkce ve verzi zdarma
- Prostředí a funkčnost nelze upravit dle specifikace zákazníka
- Pouze 30 podporovaných profesí
- Chybí podpora funkcí, nutných pro laboratoř, jako definice experimentů apod.
- Funkce nad rámec služby nelze doplňovat
- Nelze měnit funkčnost služby
- Verze zdarma obsahují reklamy

5.1.4 Splnění funkčních požadavků

Systém nepodporuje více uživatelských rolí. Nabízí pouze jeden pořadatelský účet. Ve verzi zdarma je omezen počet rezervací pouze na 40/měsíc a také počet uživatelských účtů s tím, že systém nabízí pouze jeden pořadatelský účet. Systém také neumí vytvořit automaticky časový návrh konání události.

5.2 Orario



www.stafle.cz/orario.php

	Út 28.06.2011												St 29.						
	10	11	12	13	14	15	16	17	18	19	20	21	22	10	11	12	13	14	15
B1		Kovar			Vodrážka				Ma		Jirásek								
B2	Vonásek				Kovar													Kovar	
S1											Jirásek							Malý	
S2	Kovanda							Vlasák											
S3				Novák			Zajíče											Vonásek	
V1																			
V2																		Kadeřábek	

Obrázek 18: Uživatelské prostředí služby Orario, zdroj: [3]

Orario je další univerzální rezervační systém. V základní verzi má pouze omezenou funkčnost. Lze vytvářet a editovat jednotlivé rezervace, nastavovat jejich čas a podobně, ale nelze vytvořit událost, na kterou by se poté bylo možné hlásit. Systém je provozován jako webová služba. Není proto třeba ho nijak nasazovat a lze jej ihned začít využívat. Funkčnost se dá doplnit, ale zákazník nemá možnost si funkce vytvářet sám. Funkce lze po domluvě s majitelem služby doplnit za dohodnutý poplatek. [3]

5.2.1 Cena

Systém je pro malé podniky a instituce zdarma, pouze ale se základní funkčností. Za každou funkci požadovanou navíc se platí dohodnutý poplatek. [3]

5.2.2 Výhody

- Nulový čas potřebný k nasazení
- Již hotový základ aplikace
- Možnost doplnit potřebné funkce
- Možnost upravit funkcionalitu služby
- Fungování služby bez poplatků

5.2.3 Nevýhody

- Nutnost platit za funkce navíc
- Velice omezená funkčnost základní aplikace
- Nemožnost vlastního rozšíření aplikace

5.2.4 Splnění funkčních požadavků

Systém neumožňuje generovat automatický časový návrh a ani v základní verzi není zdarma.

5.3 Doodle

www.doodle.com

August 2010									
	Wed 25			Thu 26			Fri 27		
	2:00 PM	3:00 PM	4:00 PM	2:00 PM	3:00 PM	4:00 PM	2:00 PM	3:00 PM	4:00 PM
Bill Gates	(OK)	(OK)	OK		OK			(OK)	
Linus Torvalds			(OK)		(OK)			(OK)	
Tech Advice 4 SMB	OK	OK	OK	OK	OK	OK	OK	OK	OK
Fred Jones	○	○	○	○	○	○	○	○	○
	○	○	○	○	○	○	○	○	○
	○	○	○	○	○	○	○	○	○
Count	1	1	2	1	2	1	1	1	1
	1	1	1	0	1	0	0	2	0
	1	1	0	2	0	2	2	0	2

Obrázek 19: Uživatelské prostředí služby Doodle, zdroj: [25]

[<http://www.techadvice4smb.com/wp-content/uploads/2010/08/blog-doodle-002.jpg>]

Doodle je webová služba, která se k zajišťování rezervací používá nyní. Používá se však jen v malé míře. V případě potřeby byla ve službě vytvořena událost, do které se pak účastníci experimentů mohli hlásit. Služba však nepodporuje více uživatelů s právem vytvářet a spravovat rezervace. Pro tuto možnost je nutné mít zakoupenou verzi Business.

V základní verzi služba obsahuje reklamy a chybí jí především funkce týkající se více účtů a nedostatku uživatelských rolí. Služba v základní verzi také obsahuje reklamy. [4]

5.3.1 Cena

V základní verzi je služba zdarma, má však velmi limitovanou funkčnost. Chybí možnost více uživatelských rolí, nastavení uživatelských práv atd.

Private účet nabízí navíc statistiky o rezervacích, integraci kalendáře, automatické posílání upomínek a odstranění reklam. Účet private stojí 29 Euro/rok (780 Kč/rok), ale je zde k dispozici stále jen 1 účet s právy editovat rezervace.

Účet business umožňuje navíc upravit vzhled služby, vlastní subdoménu a více uživatelských účtu na editaci rezervací. V závislosti na počtu účtů pak stojí 119 Euro/rok (3213 Kč/Rok) za 5 uživatelských účtů až po 1899 Euro/Rok (51300 Kč/rok) za 1000 uživatelských účtů. [5]

5.3.2 Výhody

- Nulový čas nutný k nasazení
- Již hotová a fungující služba
- Základní účet zdarma
- Snadné používání
- Uživatelé laboratoře již mají zkušenost se službou

5.3.3 Nevýhody

- Velice omezená funkčnost ve verzi zdarma a private
- Omezený počet uživatelských účtů a rolí i v placené verzi
- Vysoká cena za uživatelské účty

5.3.4 Splnění funkčních požadavků

Systém nenabízí ve verzi zdarma neomezený počet účtů a také systém nedokáže automaticky generovat časový návrh události. Ve verzi zdarma existuje pouze jedna uživatelská role.

5.4 Erezervuj.cz



www.erezervuj.cz

Železný Brod - malý sál

	07 h	08 h	09 h	10 h	11 h	12 h	13 h	14 h	15 h	16 h	17 h	18 h	19 h
Kurt 1	250	250	250	250	250	350	350	350	350	350	350	350	350
Kurt 2	100	100	100	100	100	100	100	100	100	50	50	50	50

Semily - sál přízemí (větší)

	14 h	15 h	16 h	17 h	18 h	19 h	20 h	21 h
Bowling	500	500	500	500	500	500	500	500

Legenda - hodiny v přehledu

- proběhlá hodina
- plná hodina
- volná hodina
- již jste registrován
- označená hodina

Červen 2011 >>

Po	Út	St	Čt	Pá	So	Ne
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

Vybráno: 07. 06. 2011

ID	Datum	Čas od	Čas do	Cena
Kurt 1	07. 06. 2011	17:00	18:00	350 Kč x

rezervuj

©TFnet s.r.o. 2011

Obrázek 20: Uživatelské prostředí služby Erezervuj.cz, zdroj: [6]

Tento rezervační systém se zaměřuje na jednotlivé profese. Má 4 moduly – cvičení, sportoviště, chalupy a soutěže. Jedná se o čistě webovou službu, kde je třeba pouze rezervace. Není nutné službu nasazovat. Služba neumožňuje kromě volby modulu nijak upravovat funkce nebo prostředí služby.

Tyto 4 moduly ale v základní verzi neobsahují zásadní funkce jako více uživatelských rolí a je omezen počet rezervací. V placené verzi je více uživatelských rolí a neomezený počet rezervací. Jsou zde navíc pokročilejší možnosti notifikací a statistik.

I v placené verzi ale chybí možnost doplnit vlastní funkce jako vlastní autorizace uživatelů. [6]

5.4.1 Cena

Základní verze služby je zdarma. Je však limitován počet rezervací a počet a počet účtů, které mohou rezervace editovat.

Placená verze stojí od 100 Kč/měsíc do 250 Kč/měsíc v závislosti na zvoleném modulu. Počet účtů i rezervací je neomezen. [7]

5.4.2 Výhody

- V základní verzi zdarma
- Službu není potřeba nasazovat
- Již hotová a fungující služba

5.4.3 Nevýhody

- Nemožnost upravovat funkčnost služby
- Žádný z modulů neobsahuje všechny potřebné funkce
- Základní verze je funkčně velmi omezena

5.4.4 Splnění funkčních požadavků

Systém ve verzi zdarma podporuje pouze jeden uživatelský účet. Má také omezený počet rezervací na 100/měsíc. Funkce také nepodporuje možnost generovat časový návrh událostí.

5.5 Bookappo



www.bookappo.cz

The screenshot shows the Bookappo web interface. At the top, there are navigation tabs: OVERVIEW, CALENDAR, CLIENTS, MARKETING, and SETTINGS. The main area displays a calendar for Paula Winton, showing a grid of days from Monday, June 18th to Sunday, June 24th. The calendar is color-coded by service type: green for Massage, blue for Therapy, and yellow for Luxury Manicure. Specific appointments are listed with names and times, such as '8:00 - 9:00 Matej Sulc Luxury Manicure' on Monday and '9:00 - 11:00 Frank Wellens Therapy' on Tuesday. A sidebar on the left lists staff members under the heading 'STAFF', including Paula Winton, Jennifer Thomas, and others. The interface also shows a 'Today' dropdown and a 'Work Week' view option.

Obrázek 21: Uživatelské prostředí služby Bookappo, zdroj: [26]

Tato služba je univerzální rezervační systém, určený především menším živnostníkům a menším firmám. Jde o webovou službu, není tedy nutné její nasazování. Služba je dostupná ve dvou verzích, Lite a Pro. V Lite verzi je omezen počet událostí, které může uživatel rezervovat. Systém nabízí možnost generovat statistiky z již uskutečněných rezervací, posílat upozornění během uskutečnění rezervací, úpravu vzhledu uživatelského prostředí. Kromě přístupu přes web služby jsou k dispozici také aplikace pro platformy iOS a android. Služba také nabízí integraci se sociální sítí facebook.

Služba však neumožňuje přidávání vlastních funkcí, jako například vlastní způsob přihlašování. Uživatel je tedy omezen pouze na funkce již implementované ve službě. Je zde také omezen počet uživatelských rolí a účtů. [8]

5.5.1 Cena

Cena této služba je v základní verzi Lite 19 dolarů/měsíc (400 Kč/měsíc). V této verzi je omezen počet událostí k rezervaci.

Verze Pro stojí 39 dolarů/měsíc (819 Kč/měsíc) a oproti verzi Lite obsahuje neomezený počet událostí k rezervaci. [9]

5.5.2 Výhody

- Již hotová a fungující služba
- Není potřeba nasazovat
- Multiplatformní klient
- Integrace se sociálními sítěmi

5.5.3 Nevýhody

- Neexistuje verze zdarma
- Nemožnost přidávat vlastní funkce
- Omezený počet uživatelských rolí

5.5.4 Splnění funkčních požadavků

Služba nemá verzi zdarma. V levnější variantě nepodporuje neomezený počet uživatelů. Tak neumožňuje generovat automatický časový návrh události.

5.6 Planyo



www.planyo.com

Calendar overview Color filter: --- Resource filter: All apartments

«previous next» today

October 2014															November 2014																														
W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S														
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Casa Ala 1															x x x x x x x x x x																														
Casa Scim 1																																													

Available
 Reserved
 Arrivals
 Unavailable

Pending reservations - Your action is required

None found

Upcoming reservations

Reservation ID	Apartment	User	Rental time	Status	Created
R785130	Casa Ala 1	Tomas	2014-10-16 - 2014-10-25	Email address verified + Confirmed	2014-10-06 17:10

Latest reservations entered

Reservation ID	Apartment	User	Rental time	Status	Created
R785130	Casa Ala 1	Tomas	2014-10-16 - 2014-10-25	Email address verified + Confirmed	2014-10-06 17:10
R623906	Casa Ala 1	Anderson	2014-05-22 - 2014-06-06	Email address verified + Confirmed	2014-05-22 12:35
R445320	Casa Ala 1	Anderson	2013-12-21 - 2014-01-02	Email address verified	2013-11-16 13:01
R256994	Casa Ala 1	Joel Alegret	2013-03-23 - 2013-03-29	Fraud	2013-03-08 11:30
R204546	Casa Ala 1	Nicolas Meichtry	2012-11-23 - 2012-11-24	Email address verified + Confirmed	2012-11-08 16:10
R162867	Casa Ala 1	Karolina- cliente (karolina)	2012-07-24 - 2012-07-31	Email address verified	2012-07-24 20:28
R102378	Casa Ala 1	Joel Alegret	2012-02-01 - 2012-02-15	Email address verified	2012-01-25 14:48

Obrázek 22: Uživatelské prostředí služby Planyo, zdroj: [11]

[<http://www.planyo.com/learn-more.php>]

Tato služba umožňuje rezervování uživatelů na předem vytvořené události. Ty může vytvářet jak administrátor, tak zaměstnanci. Služba tedy podporuje více uživatelských rolí, stejně tak více uživatelských účtů. Umožňuje automatické vytváření událostí, statistiky rezervací a také integraci se sociálními sítěmi, google kalendářem a webovými stránkami třetích stran pomocí widgetů.

V základní verzi Free je však zpoplatněna rezervace uživatele a počet rezervovatelných událostí. [10]

5.6.1 Cena

Ve verzi Free je služba zdarma, každá rezervace stojí uživatele 1 Euro (27 Kč), také je omezen počet událostí k rezervaci a je omezena funkčnost jako notifikace, vlastní ověřování uživatelů a podobně.

Verze Pro stojí 522 Kč/měsíc. Každá událost pro rezervaci stojí 45-100 Kč/měsíc v závislosti na počtu událostí. Uživatel zde neplatí za jednotlivé rezervace. [11]

5.6.2 Výhody

- Poměrně široká funkčnost
- Není třeba nasazovat, již funkční služba
- Více uživatelských rolí a funkcí
- Integrace se sociálními sítěmi

5.6.3 Nevýhody

- Ve verzi zdarma musí platit uživatel za každou rezervaci
- Ve verzi zdarma omezena funkčnost a počet rezervovatelných událostí na 10
- V placené verzi nutno platit za každou událost (vysoká cena)

5.6.4 Splnění funkčních požadavků

System nemá verzi bez poplatků. Ve verzi zdarma s poplatky omezený počet událostí. System taky nepodporuje vytváření automatického časového návrhu.

5.7 Srovnání řešení třetích stran

Žádné z řešení třetích stran nespĺnilo všechny funkční požadavky uvedené v kapitole 4 na straně 2. Konkrétně funkční požadavek F3, „System bude umožňovat vygenerování automatického časového návrh experimentu“, nespĺnilo žádné z řešení třetích stran.

Nejlépe ze srovnání vychází služby Orario (strana 13) a Planyo (strana 19) s 8 splněnými funkčními požadavky z 10. Bohužel však ani tyto služby nejsou bezplatně dostupné a neumožňují generování automatického časového návrhu. Ostatní služby splňovaly pouze 6-7 funkčních požadavků (bližší popis splnění/nespĺnění funkčních požadavků se nachází u popisu každé služby v předchozí kapitole)

Details o jednotlivých řešeních třetích stran jsou uvedeny v předchozí kapitole v úvodu o samotných funkcích. Níže je uvedena tabulka (tabulka 1), kde je shrnuto splnění/nespĺnění jednotlivých funkčních požadavků jednotlivými službami.

Tabulka 1: Srovnání splnění funkčních požadavků, zdroj: vlastní práce

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
Reservario.com	Ne	Ano	Ne	Ano	Ano	Ano	Ano	Ano	Ne	Ne
Orario	Ano	Ano	Ne	Ano	Ano	Ano	Ano	Ne	Ano	Ano
Doodle	Ne	Ano	Ne	Ano	Ano	Ano	Ano	Ano	Ne	Ne
Erezervuj.cz	Ne	Ano	Ne	Ano	Ano	Ano	Ano	Ano	Ne	Ne
Bookappo	Ano	Ano	Ne	Ano	Ano	Ano	Ano	Ne	Ano	Ne
Planyo	Ano	Ano	Ne	Ano	Ano	Ano	Ano	Ne	Ano	Ano

Jelikož ani jedna služba nedokázala splnit všechny funkční požadavky, bude k realizaci použito vlastní řešení systému, kde lze vlastním návrhem docílit splnění všech funkčních požadavků.

6 Vlastní řešení

Jako technologie, na nichž bude kromě dalších služba postavena, jsem omezil na jazyky PHP a Javu, jelikož s těmito dvěma jazyky mám jako autor služby nejvíce zkušeností.

6.1 PHP

PHP je nejpoblárnější a nejpoužívanější jazyk sloužící k tvorbě webových systémů a služeb. [12] Jsou na něm vystaveny také četné knihovny a frameworky. V čisté podobě jde o základní jazyk, který ve svých novějších verzích podporuje objektově orientovaný přístup ke psaní kódu. Na jazyku PHP je postaveno mnoho světových webových služeb jako Yahoo.com nebo Wikipedia.com. [13] Pro větší projekty je ale používanějším jazykem Java, neboť se považuje za robustnější a bezpečnější s tím, že je vhodnější pro weby s vyšším provozem. Výhodou PHP je hlavně snadnost implementace, široká podpora a velká komunita. Nevýhodou je, že je nutné implementovat veškerou funkčnost vlastními silami, tedy dlouhá doba implementace.

6.2 Java

Java se používá především v korporátní sféře a u větších projektů. Javu využívá většina současných nejnavštěvovanějších webových služeb jako Google, Facebook nebo Youtube. Webové aplikace se vyvíjejí především ve verzi Java EE. Ta obsahuje již hotové knihovny např. na přístup k databázím. [14] Základna uživatelů není tak široká jako u PHP. Její podpora také není tak rozšířená napříč poskytovateli jako je tomu u PHP.

Její výhodou je především její robustnost a bezpečnost. Také má lepší škálovatelnost a rozšiřitelnost. V základní verzi už také, na rozdíl od PHP, obsahuje poměrně široké množství knihoven ulehčujících implementaci.

6.3 Frameworky

Níže jsou uvedeny frameworky které jsou založené na dvou dříve zmíněných jazycích Java a PHP.

6.3.1 Nette



www.nette.org

Framework Nette je především v České republice nejpoužívanějším frameworkem využívajícím PHP. Důvodem je pravděpodobně to, že pochází původně z České republiky. Má velmi širokou základnu uživatelů a poměrně jednoduché nasazení. Díky původu v ČR je také veškerá dokumentace kompletně v českém jazyce.

Jedná se o framework využívající MVC strukturu a objektový přístup. Podporuje práci se všemi nejpoužívanějšími druhy databází jako MySQL, Oracle a další.

Mezi dalšími věcmi obsahuje poměrně snadné vytváření formulářů, jejich automatickou kontrolu, ochranu proti útokům atd.

Požadavky na server jsou fungující http server a PHP verze 5.3 a vyšší. [15][16]

6.3.2 Zend



www.framework.zend.com

Zend je dalším frameworkem pro snadnou a rychlou tvorbu webových aplikací. Využívá objektivě orientovaného přístupu psaní kódu a návrhového vzoru MVC (Model, View, Controller) k oddělení ukládání, manipulaci a prezentace dat.

Je to celosvětově známý framework s poměrně širokou základnou uživatelů. Mezi další přednosti patří podpora multi-databázových systémů, rozdělení funkčnosti do samostatně použitelných modulů atd. Pro svůj běh potřebuje tento framework webový server Apache ve verzi 2.2 a vyšší a databázi MySQL verze 5.5 a výše pro běh vestavěné databáze frameworku. [15][17][18]

6.3.3 Symfony



www.symfony.com/

Symfony je opět PHP framework, který je postaven na objektivě přístup. Jeho hlavní výhodou je snadné vytváření aplikací a mnoho již hotových dostupných funkcí a nástrojů.

Světově se jedná o jeden z nejpoužívanějších frameworků na tvorbu webů, tudíž je k dispozici velké množství již hotových příkladů a široká základna vývojářů a podpory.

Jedná se také o jeden z nejstabilnějších PHP frameworků.

Framework podporuje spojení se všemi nejpoužívanějšími databázemi jako MySQL nebo PostgreSQL. Pro svůj běh vyžaduje běžící http server, PHP verze 5.3.3 a výše a jeden z podporovaných databázových systémů. [15][19][20]

6.3.4 Play



www.playframework.com

Play patří mezi jeden z nejméně komplikovaných způsobů, jak v jazyce java psát weby. Tento framework opět používá přístup MVC, což zaručuje oddělení prezentace, manipulace a ukládání dat. Tento přístup podporuje vytváření rozšiřitelných aplikací s čitelným kódem. Je to také jeden z nejpoužívanějších javovských frameworků na tvorbu webových aplikací. To zaručuje silnou uživatelskou základnu a velké množství příkladů a již hotové funkcionality.

Dalšími výhodami jsou například robustní systém testování aplikace.

Pro běh toho frameworku je vyžadován běh javy. [20][21]

6.3.5 Spring



www.spring.io

Spring je nejpoužívanějším frameworkem k vytváření webů v jazyce java. Je založen na jazyce Java EE. K tomuto jazyku přidává mnoho funkcí pomocí modulů. Obsahuje moduly jako např. „Data access“ modul pro přístup k databázím nebo „Authentication and authorization“ modul pro zabezpečené postupy přihlašování a komunikace. Nemá danou strukturu aplikace na rozdíl např. od frameworku Play, kde je uživatel veden ke struktuře aplikace MVC. To je zároveň výhodou toho řešení, neboť dává autorovi volnost při tvorbě aplikace, ale také nevýhodou, kdy kód není tak čistý a čitelný. Pro běh aplikace je vyžadována java verze 5 a výše. [20][23]

7 Srovnání frameworků

Všechny zde uvedené frameworky mají jak své výhody, tak nevýhody. Z PHP frameworků jsem vybral frameworky Nette a Zend a z Java frameworků Play. Důvodem bylo, že s těmito frameworky mám již jako autor budoucí aplikace zkušenosti a vývoj bude díky rychlejšímu seznámení rychlejší a méně problémový.

Framework Spring jsem vyřadil, z důvodu složitějšího vývoje a podobnosti s Javou ve verzi EE, se kterou nemám jako autor zkušenosti. Framework Symfony jsem vyřadil především kvůli slabé místní podpoře a chybějící dokumentaci a příkladům v českém jazyce.

Z těchto 3 frameworků se rozhodnu pro jeden konkrétní až na základě serveru, na kterém se bude aplikace nasazovat. Bude rozhodovat především podpora požadavků jednotlivých frameworků.

7.1 Výsledek srovnání

Z řešení třetích stran nespĺnilo ani jedno ze zmíněných řešení všechny funkční požadavky, jak je uvedeno v kapitole 5.7. Tudíž jsem se rozhodl pro vlastní implementaci rezervačního systému. Jako dva základní jazyky jsem si zvolil jazyk Java a PHP. Na těchto jazycích je postaveno mnoho frameworků, které usnadňují a zrychlují implementaci, a ty jsem se rozhodl použít. Ze srovnání nakonec vyšly tři frameworky – Zend, Nette a Play. Důvodem byla předchozí zkušenost s těmito frameworky, rychlý vývoj aplikace a dobrá uživatelská podpora. Podrobně jsou frameworky a jejich porovnání rozebrány v kapitole 6 a 7.

Rozhodnutí nakonec padlo na framework Play, protože jsem s ním jako autor aplikace měl největší zkušenosti a umožňoval nejrychlejší vývoj ze všech výše zmíněných.

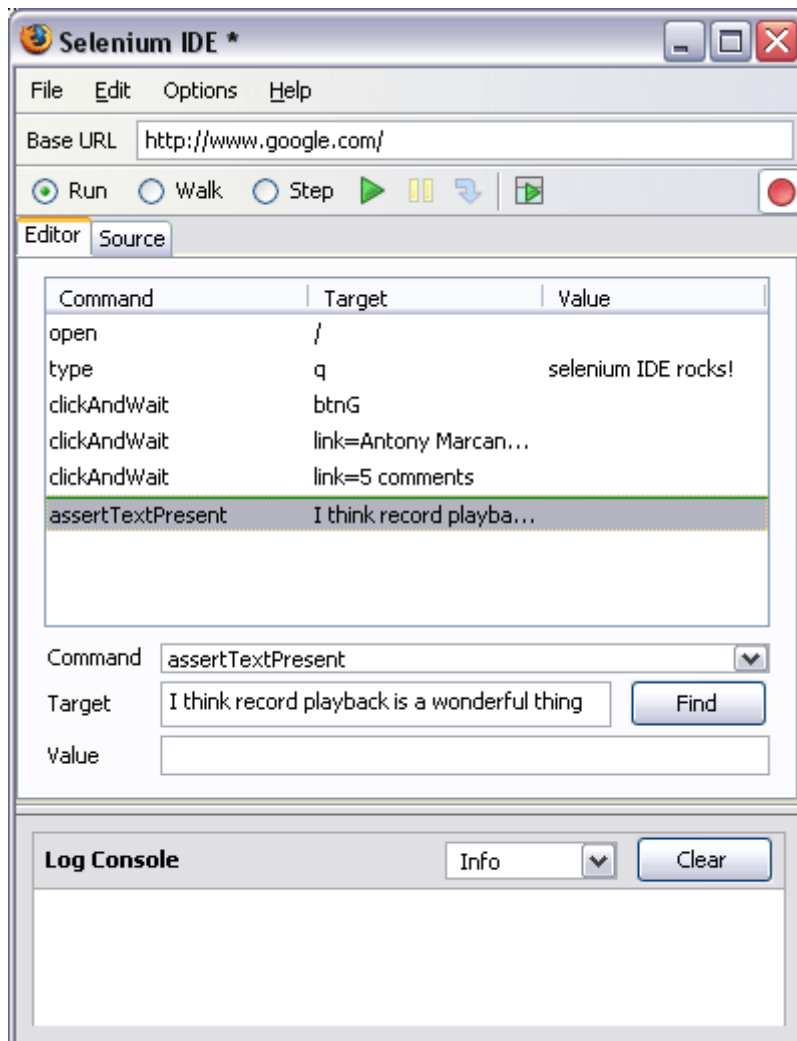
7.2 Framework Play

Framework Play plně podporuje architektonický styl REST. REST je architektura rozhraní, sloužící ke komunikaci mezi jednotlivými systémy. Na rozdíl od alternativ jako SOAP je orientován datově. Webové služby definují procedury a protokol, REST definuje přístup k samotným datům. Proto se REST používá především pro jednotlivý přístup k datům. REST implementuje čtyři základní metody pro práci s daty pod označením CRUD – vytvoření dat Create, získání dat Retrieve, změnu a úpravu dat Update a smazání dat Delete tyto metody jsou implementovány pomocí odpovídajících HTTP metod. POST a PUT odpovídající Create slouží k získání dat, DELETE odpovídá Delete (protože bývají webové nástroje často omezeny na metody POST a GET, bývá tato metoda nahrazována metodou POST s parametrem, který sděluje, že se má ve skutečnosti použít metoda Delete a co se má smazat) a metoda PUT odpovídá metodě Update (je podobná metodě create s tím rozdílem, že voláme konkrétní URL existujícího zdroje a předáváme novou změněnou hodnotu).

Již v základu framework podporuje základní validaci dat, vytváření formulářů a dalších funkcí potřebných ke tvorbě webových aplikací. Ty jsou ve většině ostatních frameworků jako Spring nutné importovat zvlášť nebo je ručně vytvářet. [28]

K testování je v tomto frameworku připravena podpora testů JUnit a Selenium. JUnit jsou testy k otestování menší funkčnosti, testují jednotlivé metody a třídy, nikoli funkčnost celé aplikace. Většinou se přidávají již v průběhu tvorby aplikace. Při každé implementaci další části funkčnosti se znovu spustí již vytvořené JUnit testy, čímž se zajistí, že nová funkčnost neporušila funkčnost té staré. Selenium je testovací nástroj určený k testování celé funkčnosti aplikace ze strany front-endu, tedy z uživatelského

prostředí. To dělá tento nástroj široce univerzální a použitelný pro všechny typy webových aplikací. Původně byl tento nástroj doplňkem pro prohlížeč Mozilla, dnes jsou již i jiné způsoby jeho použití. [29]



Obrázek 23: ukázka prostředí nástroje Selenium, zdroj: [27]

Jak již bylo zmíněno dříve, framework upřednostňuje architekturu MVC (model, view, controller), která podporuje čitelnost kódu, menší množství chyb a rychlejší úpravu aplikace. Pokud chce například autor změnit uživatelské rozhraní své aplikace, není nutné upravovat zpracování dat ani jejich ukládání a persistenci, ale pouze upraví jeden nebo více pohledů aplikace. V případě, kdy tato architektura není dodržena, je nutné předělávat větší funkčnost aplikace.

Už v základu Play podporuje kompletní funkčnost nutnou ke tvorbě aplikace, v případě že nějaká funkčnost není podporována (například šifrování), dá se přidat pomocí tzv. modulů.

Ke tvorbě se dají použít v základu dva jazyky – java nebo jazyk scala, který z javy vychází. Má jinou syntaxi, ale překládá se na java class soubory. Je kompatibilní s javou co se týká imporů knihoven a balíčků. Ve frameworku Play lze také použít kombinace scaly a javy. V mém případě jsem se rozhodl

použít scalu k tvorbě templatů (vrstva View) a javu ke tvorbě controllerů a modelů. Scala je čistě objektový jazyk (na rozdíl od javy tu neexistují žádná primitiva).

```
import java.util.{Date, Locale}
import java.text.DateFormat
import java.text.DateFormat._

object FrenchDate {
  def main(args: Array[String]) {
    val now = new Date
    val df = getDateInstance(LONG, Locale.FRANCE)
    println(df format now)
  }
}
```

Obrázek 24: ukázka programovacího jazyka Scala, zdroj: vlastní práce

K perzistenci dat je ve frameworku připravena podpora nástrojů Ebean, případně Scala alternativy Anorm. Ty umožňují objektový přístup k datům uloženým v databázi. Starají se o mapování hodnot uložených v databázi na objekty značené ve frameworku jako entity.

8 Návrh a implementace

8.1 Struktura projektu

Všechna konkrétní funkčnost a implementace aplikace je popsána u samotného kódu v příloženém projektu.

8.1.1 Controllery:

Controllery v architektuře MVC slouží ke zpracování dat a jich odeslání případně do prezenční vrstvy k zobrazení nebo naopak do modelové vrstvy k uložení a persistenci. V controllerech se odehrává hlavní logika aplikace. Níže jsou popsány jednotlivé controllery a jejich role v aplikaci.

Controller Application obsahuje metody, které se týkají obecného fungování aplikace a jsou používány v ostatních controllerech. Jsou zde metody jako například kontrola správnosti dat, převod Stringu na objekt Date, unikátní slučování kolekcí a podobně.

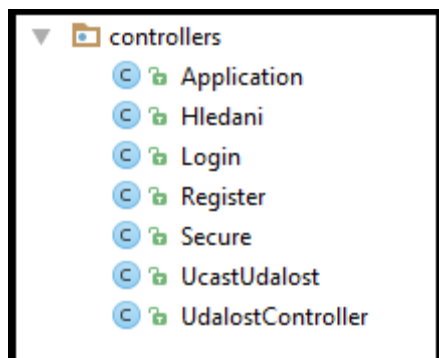
Controller Hledani obsahuje metody potřebné k vyhledávání informací v připojené databázi. Jsou zde metody například na vyhledávání osob podle jména, vyhledávání událostí a podobně.

Controllery Login a Register obsahují metody řešící vytvoření nové osoby a autentizaci již zaregistrované osoby.

Controlelr Secure se stará o omezení přístupu podle uživatelských rolí (autorizaci).

Controller UcastUdalost obsahuje metody řešící interakci přihlašování/odhlašování se na události a to jak v roli pořadatele, tak v roli účastníka a výpisu účastněných se událostí.

Controller UdalostController je nejobsáhlejším ze všech controllerů. Obsahuje všechny metody starající se o vytváření, úpravu a mazání událostí. V tomto balíčku je také řešeno vytváření a mazání časových návrhů. Kromě toho jsou zde obsaženy metody potřebné k vytváření výše zmíněných věcí – kontrola obsazenosti místností, vytvoření detailu události a podobně.



Obrázek 25: seznam všech controllerů, zdroj: vlastní práce

8.1.2 Modely:

Modely slouží k persistenci dat, ukládání dat a jejich zpětnému získání z databáze. Všechny třídy v tomto balíčku kromě třídy BCrypt jsou zároveň tzv. Entity. To znamená, že v databázi mají každý svou tabulku s odpovídajícími vztahy a parametry.

Model Cas a CasPomocna obe slouží k uložení času. CasPomocna slouží k tomu samému co čas, v aplikaci je přítomen pouze kvůli omezení počtu vazby mezi jednotlivými modely.

CasovyNavrh slouží k uložení informací o tom, kdy se událost, ke které je navázán, koná. V jakém datu, v jaké časy se mohou účastníci na událost hlásit a podobně. Vytvořením časového návrhu se budu zabývat podrobněji v dalších kapitolách.

Model Hash slouží k „zahashování“ hesla zadaného jako plaintext do zašifrované podoby při registraci nové osoby. Slouží také pak ke kontrole hesla při loginu uživatele.

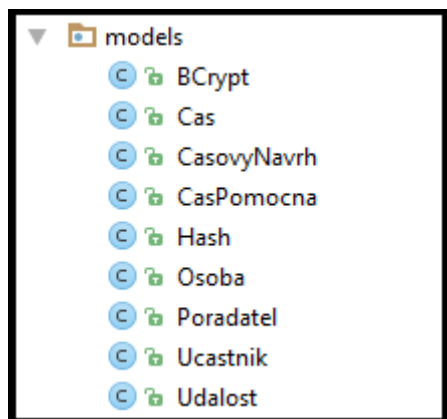
Model Osoba uchovává všechny osobní informace týkající se konkrétní osoby jako jméno, heslo, věk, pohlaví a podobně.

Modely Poradatel a Ucastnik jsou v současné době pouze prázdné objekty sloužící jako příznak, zda je uživatel pořadatelem nebo pouze účastníkem. Účastník je přiřazen každému novému uživateli, pořadatel pouze lidem majícím tuto roli. To znamená, že i pořadatel se může účastnit událostí.

Model Událost obsahuje informace o samotné události – z jakých částí se událost skládá, její název, popis, vlastník události a také seznam všech účastníků události a seznam všech časových návrhů.

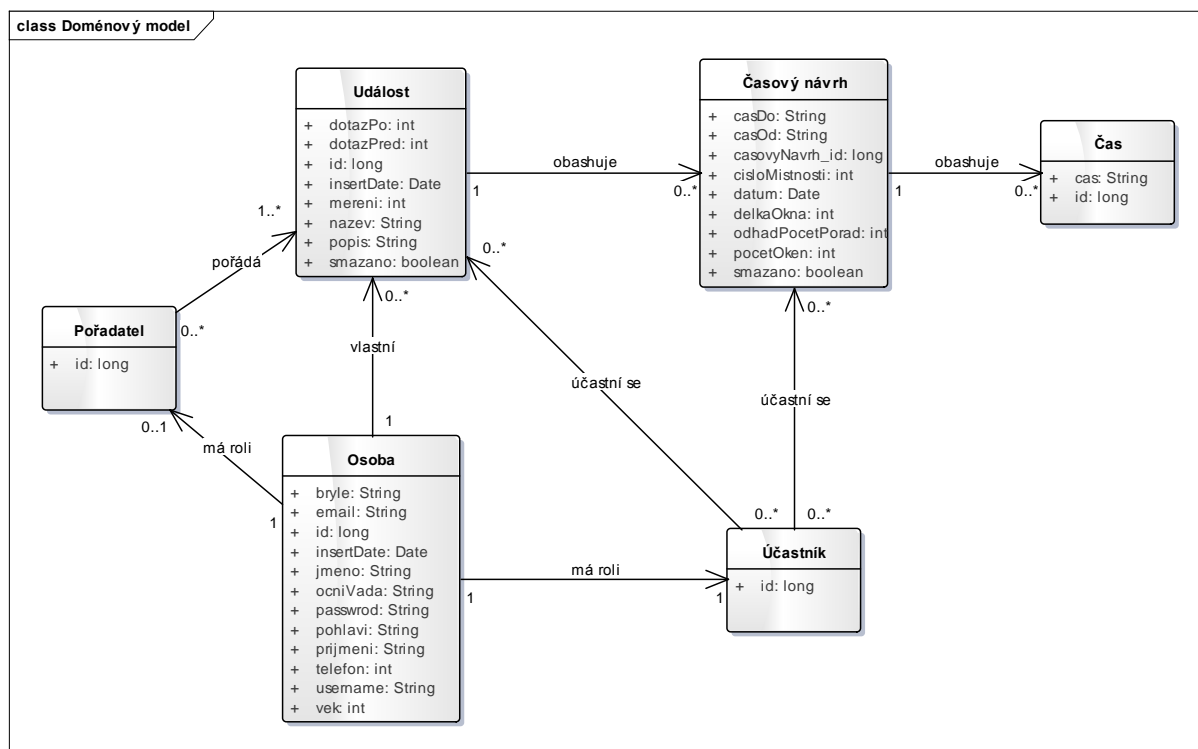
BCrypt je externí knihovna, kterou využívám v aplikaci k hashování hesel. Je založena na šifře Blowfish.

[30]



Obrázek 26: Seznam všech modelů, zdroj: vlastní práce

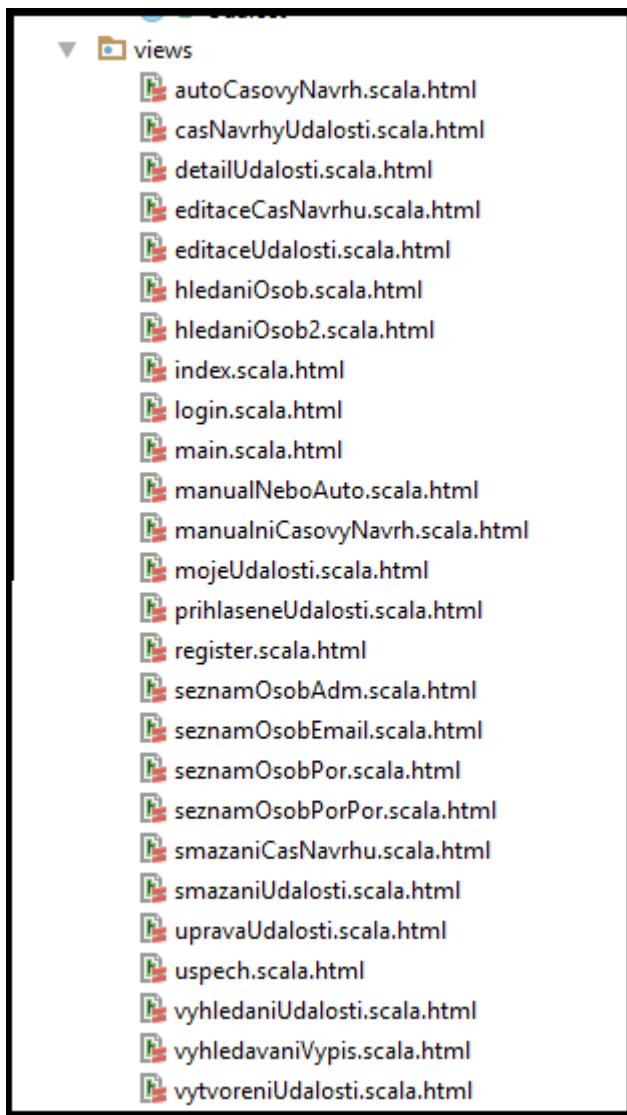
Níže je uveden doménový model, který znázorňuje vztahy jednotlivých modelů a jejich kardinality.



Obrázek 27: doménový model aplikace, zdroj: vlastní práce

8.1.3 Pohledy:

Views je prezenční vrstva aplikace. Měla by obsahovat co nejmenší část logiky a sloužit především k výpisu a zadávání informací. Prezenční vrstva v této konkrétní aplikaci je napsána v jazyce Scala.



Obrázek 28: Seznam všech templatů, zdroj: vlastní práce

Níže rozeberu pouze dva ukázkové pohledy, ostatní jsou pak již velmi podobné. První je ukáзка je z templaty vytvoreniUdalosti. V hlavičce je nutno specifikovat parametry, které do view posíláme z controllerů, ty pak používáme buď k výpisu, nebo získání informací. V tomto případě

je vidět, že do view posílám prázdný formulář `udalostForm`, ten pak vykresluji pomocí již hotové funkčnosti `helper`, které z konstruktu níže sama vygeneruje html formuláře.

```
@(udalostForm: play.data.Form[Udalost], navíc: Integer, chyba: Boolean)
@import ...
<!DOCTYPE html>

@main("Vytvoření události") {

  <div class="container">
    @helper.form(action=routes.UdalostController.save()) {

      <h1>Vytvoření události</h1>

      @if(udalostForm.hasGlobalErrors) {
        @udalostForm.globalError().message()
      }
      @inputText(
        udalostForm("nazev"),
        '_label -> "Název*',
        //'_help -> "Prosím zadejte název události.",
        '_error -> udalostForm.error("nazev"),
        '_showConstraints -> false
      )
    }
  }
}
```

Obrázek 29: ukázka pohledu `vytvoreniUdalosti`, zdroj: vlastní práce

Zde je ukázka pohledu `detailUdalosti`, který vypisuje všechny souhrnné informace o události jako časy konání, název, popis, všechny účastníky a podobně. Do pohledu posílám událost a její časové návrhy, ty pak pouze pomocí jednoduchých cyklů vypíši.

```
<h4>Časové návrhy</h4>
@if(casNavrhy.isEmpty) {
  Žádné časové návrhy.
}
<ul>
@for(c <- casNavrhy) {
  <li>
    Mistnost: @c.cisloMistnosti,
    Datum: @Application.dateToString(c.datum) od @c.casOd do @c.casDo
  </li>
  <ul>
    Účastníci:
    @for(a <- (0 until (c.pocetOken)).reverse) {
      <li...>
    }
  </ul>
}
</ul>
}
```

Obrázek 30: Ukázka pohledu `detailUdalosti`, zdroj: vlastní práce

9 Vytváření časových návrhů

Systém umožňuje přihlašovat se účastníkům na události pomocí tzv. časových návrhů. Událost obsahuje pouze obecné informace. Obsahuje také informaci, zda se bude událost skládat pouze z měření, měření a dotazování před měřením, měření a dotazování po měření, měření a dotazování před měřením a dotazování po něm.

Její navázaný časový návrh obsahuje informace ke konkrétnímu konání události jako čas, místnost, datum a podobně. Mimo jiné také obsahuje tzv. časová okna, do kterých se mohou účastníci hlásit. Ty určují, v jaký přesný čas se má konkrétní účastník dostavit na událost. Ty lze vytvářet buď ručně, kde pořadatel zadá od kdy a do kdy chce v konkrétní den událost pořádat, zadá délku časového okna, a systém vygeneruje časová okna dle zadaných parametrů.

Systém ale také podporuje tzv. automatická vytvoření časových oken. V tomto případě pořadatel zadá čas od a čas do kdy chce ten den událost pořádat a počet pořadatelů a systém automaticky vygeneruje co nejefektivněji okna tak, že žádný účastník nebude muset v průběhu události čekat na další část. Další z podmínek je také to, že lze provádět měření pouze jednoho účastníka najednou.

V následující kapitole vysvětluji výpočet délky oken, který jsem rozdělil podle počtu pořadatelů.

Vysvětlení následujících diagramů:

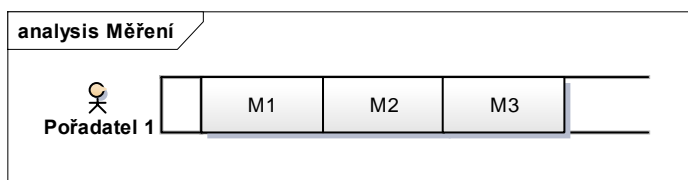
- M znamená měření
- PŘ znamená dotazování před měřením
- PO znamená dotazování po měření
- Svislá osa Y označuje pořadatele
- Vodorovná osa X označuje průběh času

9.1 Jeden pořadatel

V případě jednoho pořadatele je výpočet jednoduchý, je jím součet všech částí konkrétní události.

Skládá-li se událost pouze z měření, je délka časového okna rovna délce měření. Délka časového okna bude v tomto případě stejná pro libovolný počet pořadatelů, protože nelze provádět více měření v jeden čas.

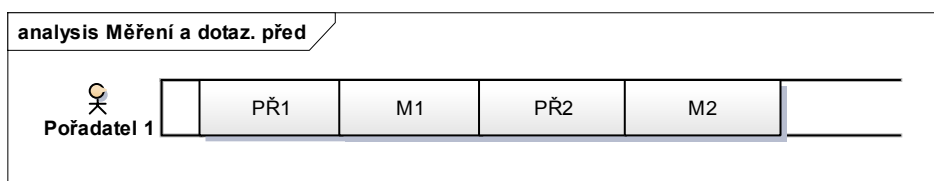
Délka okna = M



Obrázek 31: Délka okna pro 1 pořadatele - měření, zdroj: vlastní práce

Skládá-li se událost z měření a dotazování před měřením, je délka časového okna rovna délka měření + délka dotazování před měřením.

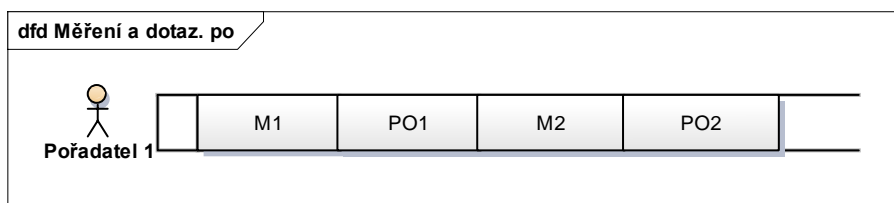
$$\text{Délka okna} = \text{PŘ} + \text{M}$$



Obrázek 32: Délka okna pro 1 pořadatele – měření a dotazování před měřením, zdroj: vlastní práce

Skládá-li se událost z měření a dotazování po měření, je délka časového okna rovna délka měření + délka dotazování po měření.

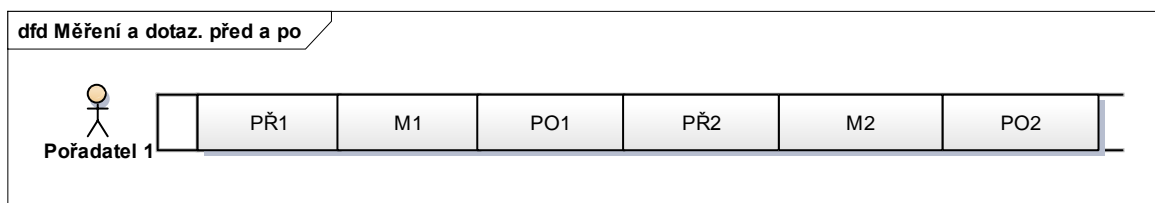
$$\text{Délka okna} = \text{M} + \text{PO}$$



Obrázek 33: Délka okna pro 1 pořadatele – měření a dotazování po měření, zdroj: vlastní práce

Skládá-li se událost, z měření a dotazování před a po měření, je délka časového okna rovna délka měření + délka dotazování před měřením + délka dotazování po měření.

$$\text{Délka okna} = \text{PŘ} + \text{M} + \text{PO}$$



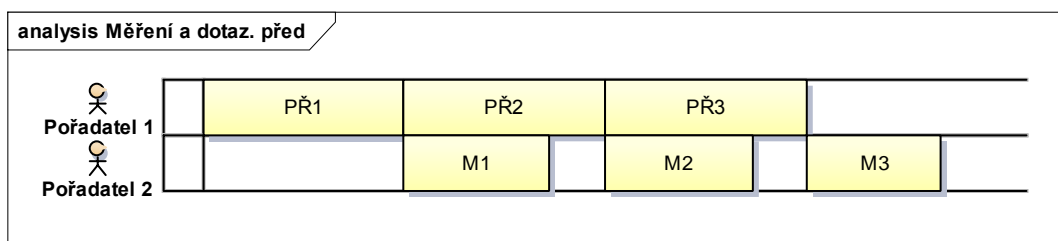
Obrázek 34: Délka okna pro 1 pořadatele – měření a dotaz. před a po měření, zdroj: vlastní práce

9.2 Dva pořadatelé

Pro dva pořadatele je výpočet nejsložitější. Je zde kromě počtu pořadatelů třeba také zohlednit délku jednotlivých částí události.

Skládá-li se událost z měření a dotazování před měřením, je délka časového okna rovna délce většímu z měření a dotazování před měřením

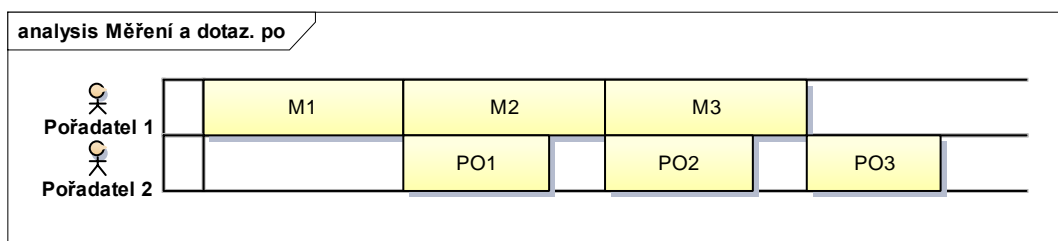
Délka okna = větší z [M, PŘ]



Obrázek 35: Délka okna pro 2 pořadatele – měření a dotazování před měřením, zdroj: vlastní práce

Skládá-li se událost z měření a dotazování před měřením, je délka časového okna rovna délce většímu z měření a dotazování před měřením

Délka okna = větší z [M, PŘ]

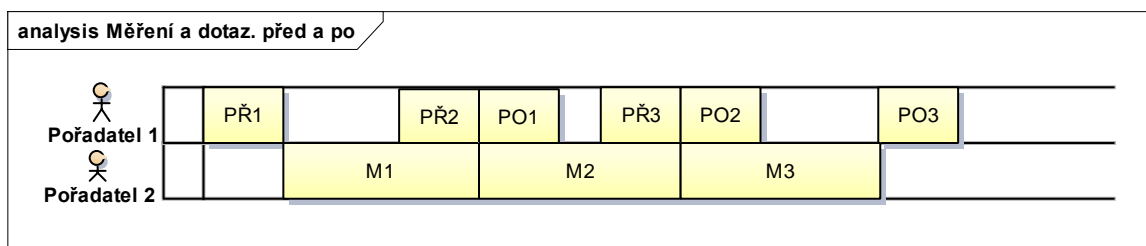


Obrázek 36: Délka okna pro 2 pořadatele – měření a dotazování po měření, zdroj: vlastní práce

V případě, že se skládá událost ze všech tří částí, je třeba výpočet rozdělit ještě na tři další části.

V případě, že je měření delší než součet dotazování před a po měření je délka okna rovna délce měření.

Délka okna = M

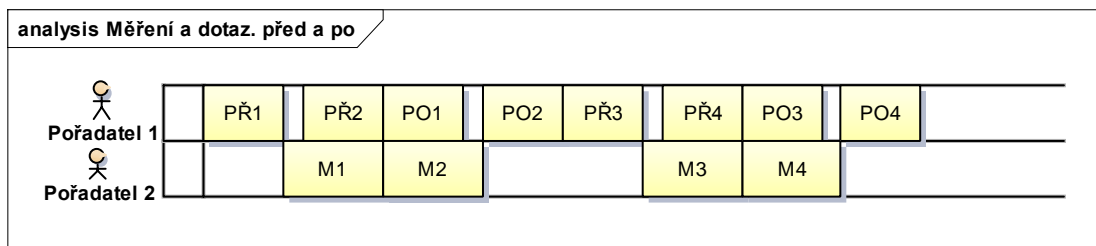


Obrázek 37: Délka okna pro 2 pořadatele – měření a dotaz. před a po měření, zdroj: vlastní práce

V případě, že je délka okna menší než součet dotazování před a po měření a zároveň je měření delší než dotazování po měření a dotazování před měřením, je délka pro lichá okna rovna délce měření a pro sudá okna součtu měření, dotazování před a po měření.

Délka lichého okna = M

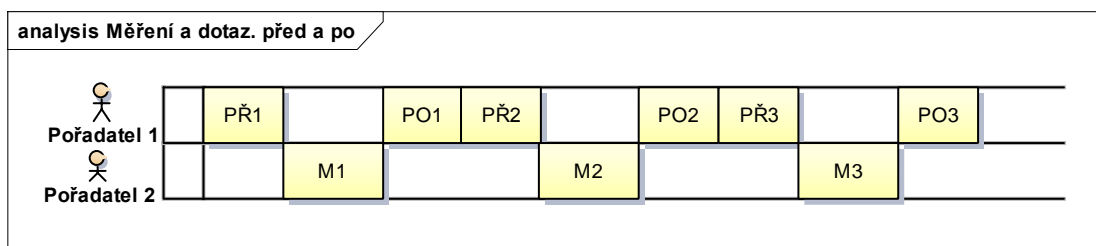
Délka sudého okna = PŘ + M + PO



Obrázek 38: Délka okna pro 2 pořadatele – měření a dotaz. před a po měření, zdroj: vlastní práce

V případě, že je délka okna menší než součet dotazování před a po měření a neplatí, že je měření delší než dotazování po měření a dotazování před měřením, je délka časového okna rovna součtu měření, dotazování před a po měření.

Délka okna = PŘ + M + PO

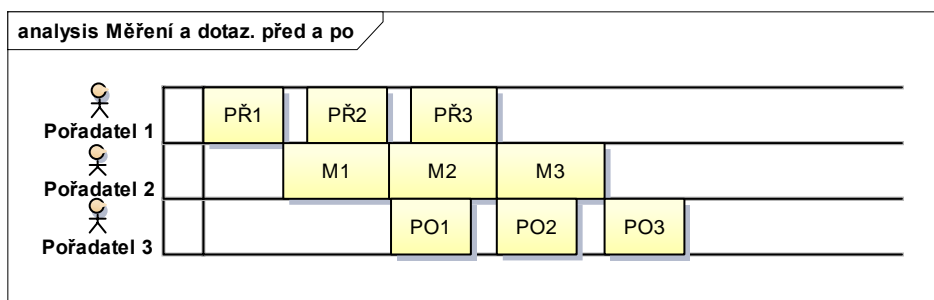


Obrázek 39: Délka okna pro 2 pořadatele – měření a dotaz. před a po měření, zdroj: vlastní práce

9.3 Tři a více pořadatelů

V případě tří a více pořadatelů se situace oproti méně pořadatelům změní pouze v případě, že se událost skládá ze všech tří částí. V této situaci je délka okna rovna délce nejdelší části události.

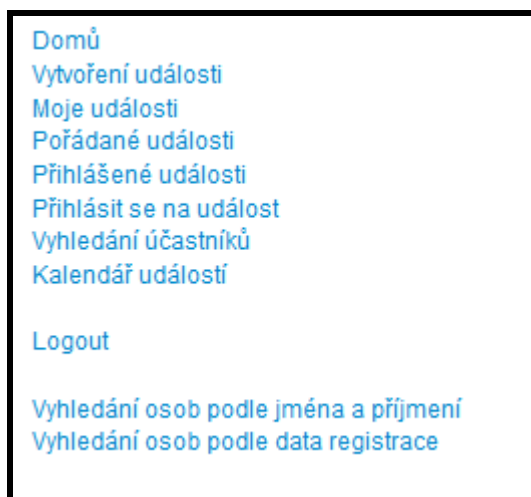
Délka okna = největší z [PŘ, M, PO]



Obrázek 40: Délka okna pro 3 pořadatele – měření a dotaz. před a po měření, zdroj: vlastní práce

10 Popis aplikace

Na obrázku níže je vidět hlavní menu z pohledu administrátora. V něm je rozcestník na všechny funkce systému. Pořadatelský účet by neviděl položky „Vyhledávání osob podle jména a příjmení“ a „Vyhledávání osob podle data registrace“, které slouží k přidělování pořadatelských práv administrátorovi. Účastník by poté neviděl navíc „Moje události“, „Pořádané události“ a „Vytvoření události“.



Obrázek 41: Hlavní menu aplikace, zdroj: vlastní práce

Pokud chce pořadatel vytvořit událost, zvolí v menu položku „Vytvoření události“. Tam vyplní požadované položky jako název, složení události a podobně. Aby se na událost mohli přihlásit účastníci, je nutné připojit k události ještě časový návrh. To lze udělat ihned po vytvoření události, nebo později na obrazovce „Moje události“. Kromě časových návrhů lze u události editovat také další věci jako název, přidávat pořadatele a podobně.

Vytvoření události	Vytvoření časového návrhu
<p>Název*</p> <input type="text" value="Událost 1"/>	<p>Číslo místnosti*</p> <input type="text" value="1"/>
<p>Popis*</p> <div style="border: 1px solid #ccc; padding: 5px; min-height: 100px;"> Popis události 1 </div>	<p>Datum ve formátu yyyy-MM-dd*</p> <input type="text" value="2015-5-2"/>
<p>Průběh události*</p> <input type="text" value="Měření a dotazování před a "/>	<p>Odhadovaný počet pořadatelů*</p> <input type="text" value="2"/>
<p>Délka dotazování před měřením v minutách*</p> <input type="text" value="10"/>	<p>Místnost je již obsazena v časech:</p> <ul style="list-style-type: none"> • 10:00 - 16:00
<p>Délka měření v minutách*</p> <input type="text" value="15"/>	<p>Čas začátku události ve formátu hh:mm*</p> <input type="text" value="16:00"/> Čas začátku události je povinný
<p>Délka dotazování po měření v minutách*</p> <input type="text" value="10"/>	<p>Čas konce události ve formátu hh:mm*</p> <input type="text" value="17:00"/> Čas konce události je povinný
<p>Vytvořit událost a přidat časový návrh</p>	<p>Uložit časový návrh</p> <ul style="list-style-type: none"> • Nepřidávat (Lze přidat později)

Obrázek 42: Vytvoření události a časového návrhu, zdroj: vlastní práce

<p>Moje události:</p> <ul style="list-style-type: none"> • Událost 2 <ul style="list-style-type: none"> ◦ 2015/05/02 16:00 do 17:00 • Událost 1 <ul style="list-style-type: none"> ◦ 2015/05/02 10:00 do 16:00 ◦ 2015/05/02 20:00 do 23:00 	<p>Editace události "Událost 1"</p> <ul style="list-style-type: none"> • Detail události • Upravit název nebo popis události • Přidat pořadatele • Editace časových návrhů události • Smazat událost • Zpět
--	--

Obrázek 43: Moje události a Editace události, zdroj: vlastní práce

Nyní se již může účastník hlásit na událost. Tu může náleznou pomocí vyhledávání „Přihlásit se na událost“, nebo pomocí linku na událost, který mu někdo pošle externě, nebo ji může nalézt v kalendáři událostí. Přihlášení a odhlášení probíhá na obrazovce „Detail události“, kde účastník vidí souhrn informací o události a volná časová okna, na která se může přihlásit.

<h2>Kalendář událostí</h2> <p>Další měsíc Předchozí měsíc</p> <h3>Květen</h3> <p>2</p> <p>Událost 1 2015/05/02, od 20:00 do 23:00</p> <p>Událost 1 2015/05/02, od 10:00 do 16:00</p> <p>Událost 2 2015/05/02, od 16:00 do 17:00</p> <p>3</p> <p>Událost 2 2015/05/03, od 17:00 do 19:00</p>	<h2>"Událost 1"</h2> <p>"Popis události 1"</p> <p>Průběh události:</p> <p>Dotazování před: 10 minut Měření: 15 minut Dotazování po: 10 minut Celková délka události: 35 minut</p> <p>Pořadatelé:</p> <ul style="list-style-type: none"> • admin admin <p>Časové návrhy</p> <ul style="list-style-type: none"> • Místnost: 1, Datum: 2015/05/02 od 10:00 do 16:00 <p>Účastníci:</p> <ul style="list-style-type: none"> o 10:00 - Přihlásit o 10:20 - Přihlásit o 11:00 - Přihlásit o 11:15 - Tomáš Kolařík o 11:55 - Přihlásit o 12:10 - Přihlásit o 12:50 - Přihlásit o 13:05 - Přihlásit o 13:45 - Přihlásit o 14:00 - Přihlásit o 14:40 - Přihlásit o 14:55 - Přihlásit o 15:35 - Přihlásit <p>Emaily účastníků</p> <p>elukok@gmail.com</p>
---	--

Obrázek 44: Kalendář a detail událostí, zdroj: vlastní práce

11 Testování

11.1 Unit Testy

Unit testy mají za úkol otestovat funkčnost jednotlivých tříd, používaných v aplikaci a jejich metod. Kontroluje se zde návratová hodnota oproti očekávané. Metodám se předá určitý parametr a kontroluje se vrácená hodnota. Předaný parametr může být i chybný vstup a potom se unit testem kontroluje reakce metody na špatný vstup. Tímto způsobem je možné testovat všechny vrstvy aplikace. Tento typ testů se vytváří především během vývoje. Konkrétně byl použit framework junit.

11.2 Selenium testy

Funkčnost kompletní aplikace byla poté otestována pomocí frameworku Selenium, který slouží k testování pomocí uživatelského prostředí aplikace. Framework simuluje používání aplikace reálným uživatelem. Pomocí tohoto nástroje lze simulovat zadávání dat do aplikace, procházet už zadaná data v aplikaci, kontrolovat jejich správnost atd.

12 Závěr

První část bakalářské práce byla použita k vyhodnocení požadavků na budoucí systém. Bylo třeba zjistit, jaké uživatelské role budou v systému použity, jaké aktivity budou v systému probíhat a podobně. Poté byla provedena analýza již existujících služeb a zjištěno, že neexistuje volně dostupná služba, která 100% vyhovuje požadavkům na systém. Proto jsem se rozhodl systém sám vytvořit.

Nejdříve jsem se seznámil se všemi vhodnými technologiemi a vybral framework Play. Vybral jsem ho kvůli rychlému vývoji a modernímu přístupu k implementaci. Měl jsem s touto technologií již lehké zkušenosti z minula, ale bylo potřeba se mnoho věcí v průběhu doučit.

Úspěšně se podařilo vytvořit aplikaci dle počátečních požadavků. Cílem implementace bylo vytvořit jednoduché uživatelské prostředí, které by ale umožňovalo používat plnou funkčnost systému, což se nakonec podařilo. Obtížnější částí implementace bylo automatické generování oken, kde bylo nutné najít správný postup výpočtu tak, aby byla okna generována co nejefektivněji se splněním předem daných podmínek. Implementace této funkčnosti se v práci podařila.

Aplikace dodržuje konvence MVC a lze tedy snadno upravit vzhled aplikace či dodat další funkčnost.

Další rozvoj aplikace bude především v jejím nasazení. Přenést ji do funkční podoby na produkčním prostředí spolu s potřebnými nástroji pro běh aplikace.

Do budoucna by se také mělo upravit a vylepšit uživatelské prostředí, které je v tuto chvíli dodělané do té podoby, aby pokrylo 100% požadované funkčnosti a umožňovalo obsluhu všech funkcí. Úprava se bude konzultovat s případnými budoucími uživateli systému.

Po prvním nasazení bude nutné testovacím provozem zjistit, zda uživatelům nechybí nějaká funkčnost a tu případně dodělat.

Zadání je tedy analýzou a implementací splněno.

Během analýzy a především vývoje jsem se naučil mnoho nových věcí a framework Play bych se rád věnoval znovu v nějaké z dalších prací. Práce pro mne byla přínosem.

13 Seznam zkratek a pojmů

- Diagram aktivit – grafické znázornění všech částí průběhu aktivity
- MVC – Model View Controller
- Framework – je softwarová struktura, obsahující různé knihovny a prostředí s již hotovými funkcemi. Často podporuje různé postupy při vývoji pomocí návrhových vzorů, díky čemuž vzniká čistší a čitelnější kód.
- JDBC – API pro jazyk java definující rozhraní pro přístup k relačním databázím
- Java EE – Enterprise edice programovacího jazyka Java. Je určena především pro provoz a vývoj podnikových aplikací a informačních systémů.
- REST – cesta, jak pomocí http volání upravovat, číst, mazat nebo tvořit data v databázi
- SOAP – protokol pro výměnu zpráv přes síť pomocí protokolu http
- Scala – programovací jazyk, který je plně objektový ale s rysy funkcionálního programování
- Template – v kontextu této práce znamená konstrukt, který určuje způsob zobrazení dat, které se templatu předají
- Ebean – nástroj sloužící k překladu relačních dat získaných z databáze na objekty, je napsaný v javě
- Anorm - nástroj sloužící k překladu relačních dat získaných z databáze na objekty, je napsaný ve scale
- Plaintext – nezměněný vstup od uživatele (bez jakékoli úpravy systémem)
- Hash – plaintext zadaný uživatelem převedený šifrovací funkcí do zašifrované podoby

14 Použité zdroje

- [1] JANOUŠ, Marek. *Reservio by rádo nahradilo tlusté rezervační knihy* [online]. 2012 [cit. 2014-11-11]. Dostupný z WWW: <<http://www.lupa.cz/clanky/rozhovor-bosiak-reservio/>>
- [2] *Reservio* [online]. 2014 [cit. 2014-11-11]. Dostupný z WWW: <<http://www.reservio.com/cs/>>
- [3] *Orario - alokační a rezervační systém* [online]. 2014 [cit. 2014-11-11]. Dostupný z WWW: <<http://www.stafle.cz/orario.php>>
- [4] *Doodle – funkce a produkty* [online]. 2014 [cit. 2014-11-11]. Dostupný z WWW: <<http://doodle.com/cs/funkce/>>
- [5] *Doodle premium* [online]. 2014 [cit. 2014-11-11]. Dostupný z WWW: <<https://doodle.com/cs/premium>>
- [6] *Erezervuj* [online]. 2014 [cit. 2014-11-11]. Dostupný z WWW: <<http://www.erezervuj.cz/uvod/>>
- [7] *Erezervuj – ceník* [online]. 2014 [cit. 2014-11-11]. Dostupný z WWW: <<http://www.erezervuj.cz/cenik/>>
- [8] *Bookappo – features* [online]. 2014 [cit. 2014-11-11]. Dostupný z WWW: <<http://www.bookappo.cz/features>>
- [9] *Bookappo – pricing* [online]. 2014 [cit. 2014-11-11]. Dostupný z WWW: <<http://www.bookappo.cz/pricing>>
- [10] *Planyo - features* [online]. 2014 [cit. 2014-11-11]. Text v angličtině. Dostupný z WWW: <<http://www.planyo.com/features.php>>
- [11] *Planyo – pricing informations* [online]. 2014 [cit. 2014-11-11]. Text v angličtině. Dostupný z WWW: <<http://www.planyo.com/pricing.php>>
- [12] *Technology usage comparison* [online]. 2014 [cit. 2014-11-11]. Text v angličtině. Dostupný z WWW: <<http://w3techs.com/technologies/comparison/pl-java,pl-php>>
- [13] *Programming languages used in most popular websites* [online]. 2014 [cit. 2014-11-11]. Text v angličtině. Dostupný z WWW: <http://en.wikipedia.org/wiki/Programming_languages_used_in_most_popular_websites>
- [14] *Java EE Platform Specification* [online]. 2014 [cit. 2014-11-11]. Text v angličtině. Dostupný z WWW: <<https://java.net/projects/javaee-spec/pages/Home>>

- [15] JAKOUBĚ, Jaroslav. *ORM test PHP frameworků* [online]. 2013 [cit. 2014-11-11]. Dostupný z WWW: <<http://www.zdrojak.cz/clanky/letni-test-php-frameworku/>>
- [16] *Nette Framework* [online]. 2014 [cit. 2014-11-11]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/Nette_Framework>
- [17] *Zend Framework* [online]. 2014 [cit. 2014-11-11]. Text v angličtině. Dostupný z WWW: <http://en.wikipedia.org/wiki/Zend_Framework>
- [18] *Introduction to Zend Framework* [online]. 2014 [cit. 2014-11-11]. Text v angličtině. Dostupný z WWW: <<http://framework.zend.com/manual/1.12/en/introduction.html>>
- [19] *Symfony* [online]. 2014 [cit. 2014-11-11]. Text v angličtině. Dostupný z WWW: <<http://en.wikipedia.org/wiki/Symfony>>
- [20] *Symfony* [online]. 2014 [cit. 2014-11-11]. Text v angličtině. Dostupný z WWW: <http://symfony.com/legacy/doc/doctrine/1_2/en/01-Getting-Started>
- [21] MAPLE, Simon. *Symfony* [online]. 2013 [cit. 2014-11-11]. Text v angličtině. Dostupný z WWW: <<http://zeroturnaround.com/rebellabs/the-curious-coders-java-web-frameworks-comparison-spring-mvc-grail>>
- [22] *Play documentation* [online]. 2014 [cit. 2014-11-11]. Text v angličtině. Dostupný z WWW: <<https://www.playframework.com/documentation/2.3.x/Home>>
- [23] *Developing a Spring Framework MVC application step-by-step* [online]. Text v angličtině. 2014 [cit. 2014-11-11]. Dostupný z WWW: <<http://docs.spring.io/docs/Spring-MVC-step-by-step/>>
- [24] *Appointment scheduling and online booking software* [online] Text v angličtině. 2014 [cit. 2014-11-11]. Dostupný z WWW: <<http://reservio.en.softonic.com/web-apps/>>
- [25] *Schedule a Meeting with Doodle.com* [online] Text v angličtině. 2010 [cit. 2014-11-11]. Dostupný z WWW: <<http://www.techadvice4smb.com/?p=33>>
- [26] JANOUŠ, Marek. *Rezervační systém Bookappo* [online] 2012 [cit. 2014-11-11]. Dostupný z WWW: <<http://www.lupa.cz/clanky/bookappo-rezervace-rozhovor/>>
- [27] Selenium [online]. 2015 [cit. 2015-4-27]. Dostupný z WWW: <<http://www.seleniumhq.org/projects/ide/selenium-ide.gif>>
- [28] *Play documentation* [online] Text v angličtině. 2015 [cit. 2015-4-27]. Dostupný z WWW: <<https://www.playframework.com/documentation/2.4.x/Home>>

[29] *Selenium - browser automation* [online] Text v angličtině. 2015 [cit. 2015-4-27]. Dostupný z WWW: <<http://www.seleniumhq.org/>>

[30] *Bcrypt – Blowfish File Encryption* [online] Text v angličtině. 2015 [cit. 2015-4-27]. Dostupný z WWW: <<http://bcrypt.sourceforge.net/>>