

Univerzita Hradec Králové

Fakulta informatiky a managementu

Katedra informatiky a kvantitativních metod

**Návrh rozvrhového systému směn pro personál
nemocnice**

Diplomová práce

Autor: Bc. Jiří Fiala

Studijní obor: Aplikovaná informatika – ai2-K

Vedoucí práce: doc. RNDr. Jaroslava Mikulecká, CSc.

Pracoviště: Katedra informatiky a kvantitativních metod

Prohlášení

Prohlašuji, že jsem diplomovou práci vypracoval samostatně a uvedl jsem všechny použité prameny a literaturu.

V Tlustici, 1. 2. 2016

Poděkování

Tímto bych rád poděkoval doc. RNDr. Jaroslavě Mikulecké, CSc., za poskytnuté konzultace a cenné rady při psaní této práce. V neposlední řadě bych rád poděkoval všem, kteří mi poskytli podporu v době mého studia a při psaní této práce.

V Praze, 14. 4. 2016

Anotace

Tato diplomová práce se zabývá analýzou stávajícího řešení problematiky rozvrhování směn zdravotních sester v rámci nemocničního oddělení, a následným návrhem nového řešení, které by lépe vyhovovalo dnešním trendům s využitím moderních technologií, a hlavně které by lépe vyhovovalo potřebám vedoucím zaměstnancům při vytváření přehledných a kvalitních rozvrhů. Toto ocení také zaměstnanci, pro které jsou rozvrhy utvářeny.

Práce je rozdělena na několik základních částí. Jedná se zejména o části analytické, ve kterých budou rozebrány možnosti tvorby rozvrhů z hlediska přístupu, dostupné softwarové prostředky včetně výhod a nevýhod, analýza stávajícího systému a v neposlední řadě analýza potřeb uživatelů, z důvodu návrhu systému jim na míru.

V práci bude také nastíněna problematika algoritmů a přístupů pro řešení rozvrhování směn zdravotních sester. Tato úloha je známá jako „Nurse Rostering Problem“ a jedná se v podstatě o optimalizační úlohu. V práci budou vymezeny základní pojmy a podmínky, které jsou důležité pro splnění této úlohy. Samotná implementace bude ovšem ponechána na dalším vývoji aplikace, která není součástí práce.

Nejdůležitější částí práce bude samotný návrh rozvrhovacího systému, který bude sestávat z diagramu typových úloh, návrhu vnitřní architektury systému, návrhu databázové architektury a návrhu uživatelského rozhraní. Všechny tyto části budou korespondovat s výsledky analytické části práce.

Klíčová slova: Typová úloha, třída, model, databáze, nurse rostering problém, analýza, rozvrhovací software

Annotation

This Diploma Thesis analyzes the current solution of nurse's shifts rostering in a hospital department with proposing of a new solution which better corresponds with today's trends, with the use of modern technology, and particularly which more suits to needs of leader staff with rostering well-arranged and quality rosters. This will be appreciated by employees included in this rostered shifts.

The thesis is divided into several basic parts. Part One is the analytic part, in which the possibility of creation the rosters from the perspective of approach to this field is discussed, as well as available software on market with advantages and disadvantages, the analysis of currently used method of rostering, and the analysis of requirement of users, for following design of a new tailor-made system.

The thesis also outlines problems of algorithm and approaches, which resolve the problem called the Nurse Rostering problem, which is essentially an optimization problem. The thesis contains description of basic terms, basic constraints important for successful solution of this task. The implementation of this new system is not part of this thesis.

The main objective of the thesis is the design of a new rostering system. This design contains part of Use Case diagram, design of architecture of system, design of database structure and design of user interface. Each part corresponds with results of the analytic part of this thesis.

Keywords: Use Case, class, model, databases, nurse rostering problem, analysis, rostering software

Obsah

1. Úvod.....	1
2. Analýza možných přístupů.....	2
2.1. Vymezení pojmů.....	2
2.2. Možné přístupy k řešení rozvrhu	3
2.3. Problém rozvrhování směn zdravotních sester – NRP [2].....	6
3. Analýza potřeb nemocnice	10
3.2. Analýza prostředí	11
3.3. Analýza současného řešení	12
3.4. Definování uživatelů.....	14
3.5. Analýza požadavků.....	16
3.6. Závěr analýzy.....	23
4. Dostupný software na trhu	24
4.1. Flexi Roster [11]	24
4.2. OptaPlanner [12].....	26
4.3. ABC Roster [13]	27
4.4. Rozvrhy směn verze 1.16.....	28
4.5. Závěr	28
5. Návrh nového systému	29
5.1. Postup návrhu	29
5.2. Diagramy případů užití	30
5.3. Diagramy tříd.....	39
5.4. Datový model.....	48
5.5. Návrh architektury aplikace.....	51
5.6. Návrh uživatelského rozhraní	53
5.7. Shrnutí návrhu systému	57
6. Závěr.....	58
7. Seznam použité literatury	59
8. Seznam obrázků	61
9. Seznam tabulek	62
10. Přílohy	63

1. Úvod

Plánování směn zdravotních sester nebo obecně podřízených zaměstnanců je nedílnou součástí práce každého vedoucího pracovníka, který má za úkol udržovat chod jakéhokoliv oddělení, ve kterém je směnný provoz. Směnný provoz má samozřejmě několik možných podob. Jednosměnné, dvousměnné, třisměnné nebo nepřetržité, což je přesně náš případ. Zajištění nepřetržité péče vyžaduje práci zaměstnanců v nepřetržitém provozu. Obecně řečeno, o pacienty je nutné starat se ve dne i v noc. Je tedy nutné, aby v každou chvíli bylo na pracovišti dostatek zaměstnanců pro poskytnutí prvotřídní péče. Toto samozřejmě vyžaduje vhodné rozvržení jejich směn během daného intervalu.

Toto se může zdát na první pohled jednoduché, ale je zde několik základních faktorů a podmínek, které nám do jisté míry toto rozvrhování stěžují. Je nutné například dodržet maximální počet odpracovaných hodin v měsíci, maximální počet po sobě jdoucích odpracovaných hodin, s čímž blízkce souvisí počet následných hodin pracovního volna. Těchto podmínek je stanoveno několik, a většina z nich vychází ze zákoníku práce.

Práce, jak již bylo nastíněno výše, si ukládá za svůj cíl, poskytnou kvalitní a profesionální podklad pro implementaci rozvrhovacího systému, který ovšem nebude součástí této práce. Pro splnění vytyčených cílů je důležité provést detailní analýzu možných řešení, analýzu dostupných systémů na trhu a analýzu potřeb uživatelů. Na základě těchto analýz bude proveden návrh grafického rozhraní, databázové architektury, architektury systému a v neposlední řadě návrh hardwarového řešení a zabezpečení.

2. Analýza možných přístupů

Před začátkem veškerých aktivit spojených s návrhem rozvrhovacího systému je nutné vymezit několik základních pojmů pro naprosté pochopení problematiky.

Rozvrhování obecně lze vymezit jako přidělování prostředků a času tak, abychom dosáhli optimálního výsledku a splnili jsme všechny zadané podmínky, které jsme při navrhování rozvrhu definovali. Každá aktivita v rozvrhu má předem známou dobu svého trvání. V našem případě, kdy se jedná o systém pro rozvrhování směn zdravotních sester, jsou aktivitami jednotlivé směny, který mají jednoznačně určený začátek a konec. Zdroje jsou reprezentovány zdravotními sestrami, kterým se přiřazuje na jeden den maximálně jedna směna. Možná verze rozvrhu je znázorněna na obrázku číslo 1.

	1							2						
1993 Listopad	01	02	03	04	05	06	07	08	09	10	11	12	13	14
	Po	Ut	St	Ct	Pa	So	Ne	Po	Ut	St	Ct	Pa	So	Ne
Josef Novak				TD	TD				TD	TD		TN	TN	TN
Marie Nova	TN	TN			TD	TN	TN			TD	TD			
Oldřich Stary	TD	TD	TN			TD	TD			TN	TN			
Karel Dobrota	TD	TD	TD			TN	TN			TN	TN			

Obrázek 1 - vzorový rozvrh směn [1]

Během tvorby rozvrhu směn musíme brát v potaz předem daná pravidla a omezení. Mezi pravidla patří například maximální počet po sobě jdoucích odpracovaných hodin, nebo počet celkových odpracovaných hodin v měsíci. Tato pravidla se ve většině řídí zákoníkem práce, a je tedy nutné je dodržovat bez výhrad. Omezení se v podstatě rozdělují na měkká a tvrdá, ale o těch budou pojednávat následující kapitoly.

2.1. Vymezení pojmů

Obecné pojmy jako rozvrhování a plánování se často v literatuře nerozlišují a jejich význam se odvozuje pouze na základě kontextu. Některá literatura ovšem pracuje s drobnými rozdíly mezi těmito výrazy, a proto považuji za vhodné stručně termíny popsat.

2.1.1. Scheduling [2]

V tomto případě se jedná o přidělování zdrojů objektům takovým způsobem, aby byly minimalizovány celkové náklady použitých zdrojů. Například při plánování výroby nebo při plánování logistiky dopravy.

2.1.2. Timetabling [2]

Je přidělování zdrojů objektům vzhledem na nejpřesnějšímu splnění požadovaných cílů, jako je například při tvorbě školních rozvrhů a podobně.

2.1.3. Sequencing [2]

Je konstrukce pořadí vzhledem k zadaným omezením. V tomto případě záleží na pořadí provádění aktivit. Toto rozvrhování lze využít například při plánování proudové výroby nebo při problému obchodního cestujícího.

2.1.4. Rostering [2]

Je umísťování zdrojů vzhledem k daným omezením a pravidlům. Zdroje je v tomto případě možné často opakovat v rámci rozvrhu. Do tohoto druhu rozvrhování spadá problém s naším rozvrhováním směn zdravotních sester.

2.2. Možné přístupy k řešení rozvrhu

Přístupů k řešení problematiky rozvrhování směn zdravotních sester je několik. V podstatě je tyto přístupy možné rozdělit na:

- ruční v tištěné podobě,
- ruční na počítači v některém softwaru,
- poloautomatický,
- zcela automatický – využití „Nurse Rostering Problem“ algoritmu.

Každý z výše uvedených přístupů má své klady a zápory. V kostce se dá ale říct, že čím více zaměstnanců v rozvrhu, tím je sestavení rozvrhu náročnější, a tedy ruční formy nejsou příliš vhodné. Je nutné ještě podotknout, že u žádného z následujících řešení nelze na 100 % určit, zda bude rozvrh optimalizovaný bez výjimek. V drtivě většině

případů se bude jednat o přijatelný kompromis času potřebného na vyřešení a výsledným rozvrhem, a to i u zcela automatických systému, jak bude rozebráno dále.

2.2.1. Ruční řešení

Ruční řešení navrhování rozvrhů směn sestává ve většině případů se sestavování rozvrhů například v aplikaci MS Excel. Každá směna musí být ručně zadána hlavně podle zkušeností vedoucího pracovníka. Obecně se ale dá říct, že řešení je vhodné pouze pro omezený kolektiv pracovníků na pracovištích, kde není definováno mnoho podmínek, omezení nebo mnoho druhů směn. Ve chvíli, kdy je zaměstnanců větší počet, doba pro vyřešení rozvrhu roste neúměrně a pracovník je nepřiměřeně zatížen tvorbou rozvrhu.

Tabulka 1 - ruční řešení

Výhody	Nevýhody
nízké nároky na výpočetní výkon	při vyšším počtu zaměstnanců složité
	přeplánování směn je zdlouhavé
	systém nekontroluje omezení
	absence evidence zaměstnanců

2.2.2. Ruční řešení v sofistikovanějším softwaru

Sofistikovanějším, avšak v dnešní době stále nedostačujícím řešením, je tvorba ručního rozvrhu, za pomoci nějakého z dostupných softwarů. V principu se jedná pouze o rozšíření předchozí varianty. Výhodou oproti ručnímu vytváření rozvrhů je například že software dokáže do jisté míry kontrolovat zadané podmínky jako počet směn, počet odpracovaných hodin atd. Většinou také podporují vedení evidence zaměstnanců a evidenci jejich odpracovaných hodin pro pozdější zpracování.

Tabulka 2 - ruční řešení sofistikovanější

Výhody	Nevýhody
nízké nároky na výpočetní výkon	při vyšším počtu zaměstnanců složité
software kontroluje pravidla	přeplánování směn je zdlouhavé
evidence zaměstnanců	
evidence odpracovaných směn	
přístup více osob	

2.2.3. Poloautomatické

Další vyšší varianta softwarového řešení generování rozvrhů směn je varianta poloautomatická. V tomto případě se jedná o software, který má již základní algoritmy pro rozvrhování směn zdravotních sester implementovány. Nechápejme tyto algoritmy jako kompletní řešení, ale pouze jako šablonu, která vytvoří základní rozvrh, který je nutné ručně upravit a odstranit nedokonalosti. Tento druh softwaru je stále poměrně jednoduchý a nevyžaduje mnoho výpočetního výkonu, jelikož nedochází k žádné optimalizaci. Můžeme ale říci, že při rozumném počtu zaměstnancům je již toto řešení celkem účinné, avšak stále ne dokonalé.

Tabulka 3 - poloautomatické řešení

Výhody	Nevýhody
software kontroluje pravidla	nutnost dodatečných úprav
evidence zaměstnanců	
evidence odpracovaných směn	
přístup více osob	

2.2.4. Automatické

Nejprofesionálnější varianta sestavování rozvrhů směn zdravotních sester je bezesporu zcela automatický software. V rámci tohoto softwaru jsou již implementovány algoritmy pro sestavení téměř bezvadného rozvrhu směn. Samozřejmě jako cena takto sestavený a optimalizovaný rozvrh je samozřejmě vyšší výpočetní a časová náročnost. Nutností pro vytvoření takto bezvadného rozvrhu je také opravdu detailní a bezchybné vymezení měkkých a tvrdých omezení. Pořízení takového softwaru je samozřejmě také nejnákladnější.

Tabulka 4 - automatické řešení

Výhody	Nevýhody
software kontroluje pravidla	vyšší výpočetní požadavky
evidence zaměstnanců	vyšší časové nároky
evidence odpracovaných směn	vyšší nároky na definici omezení
přístup více osob	vyšší pořizovací náklady
rozvrh je optimalizovaný	
snadno lze přeplánovat	

2.3. Problém rozvrhování směn zdravotních sester – NRP [2]

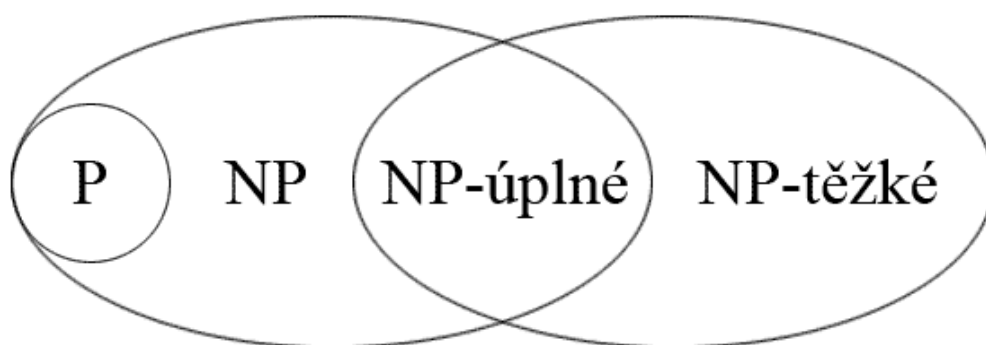
Problém rozvrhování směn zdravotních sester, dále NRP¹, patří mezi rozvrhovací problémy, které je těžké vyřešit. Cílem celého procesu je najít rozvrh směn tak, aby bylo dosaženo nejlepšího poměru mezi kvalitou rozvrhu a časem potřebným na jeho vyřešení. Problém NRP je již několik desetiletí intenzivně zkoumaným problémem, zejména kvůli své ekonomické závažnosti. I s ohledem na takto dlouhou dobu, dodnes nemůžeme říct, že bychom znali univerzálně použitelnou metodu, která by bez výhrady pokryla celý rozsah dané problematiky. První modely, které řešily problém rozvrhování směn zdravotních sester, se objevily již v 70. letech minulého století a byly založeny na matematickém programování. Ty se ale pro tento obsáhlý problém nedají použít. Pro řešení NRP se využívají zejména algoritmy:

- **Aproximační [3] [2]**
 - Algoritmy, od nichž nepožadujeme nutné optimální řešení optimalizační úlohy, ale spokojíme se s řešením, které je optimálnímu velmi blízké
- **Genetické [4] [2]**
 - Algoritmy, které využívají principů evoluční biologie, tedy dědičnosti, mutace atd., a vytváří další generace řešení problému
- **Heuristické [5] [2]**
 - Algoritmy, které pro vyřešení optimalizační úlohy používají heuristiku, řešení je tedy založené na zkušenosti, odhadu, intuici atd.
- **Pravděpodobnostní [6] [2]**
 - Algoritmy, které využívají pro vyřešení problému prvek pravděpodobnosti – mohou se rozhodnout, kterou z variant pro pokračování zvolí

Obecně můžeme říct, že při vytváření rozvrhu, tedy postupné alokace aktivit, musíme brát v potaz jisté omezující podmínky. Ty mohou mít různý charakter a mohou se týkat kterékoliv dimenze rozvrhu, jako jsou zdroje, čas atd. Tyto omezení rozdělujeme ze zásady na dva druhy, a to na tvrdá omezení a měkká omezení. Ty budou popsány dále. Dále je nutné zavést ohodnocení rozvrhu. Toto ohodnocení můžeme chápat jako určitou míru splnění nebo nesplnění požadavků, vycházejících z měkkých omezení. Na základě tohoto omezení řeší rozvrhování tzv. optimalizační úlohu.

¹ NRP – Nurse Rostering Problem

Jak již bylo popsáno výše, jedná se o problém u kterého je obtížné, někdy i nemožné, najít optimální řešení v přijatelném čase. Největším problémem a úskalím rozvrhování je jeho složitost. Složitost problému se řadí mezi NP-těžký, tedy při navyšujícím se počtu zdrojů a času nastává kombinatorické zvětšení stavového prostoru, který je nutné prohledat. Optimální řešení budeme tedy hledat velmi dlouhou dobu, a je velmi pravděpodobné, že ho nenalezneme vůbec. Krátce nastíníme problémy patřící do třídy NP. Tyto problémy můžeme rozhodnout nedeterministickým Turingovým strojem v polynomiálním čase. Jednoduše řečeno, jsme schopni ověřit řešení v polynomiálním čase. Vztahy mezi třídami složitosti jsou znázorněny na obrázku číslo 2. [7]



Obrázek 2 - třídy složitosti [7]

Z tohoto důvodu je nutné se spokojit s téměř optimálním řešením. Jak vyplývá z výše uvedených skutečností, prioritou při rozvrhování je vždy dobrý kompromis mezi dobou potřebnou pro vyřešení a výsledným výsledkem.

Obecně je možné říct, že největší změny kvality probíhají v úvodních fázích algoritmu. S přibývajícím časem ke zlepšováním kvality dochází v podstatě velmi sporadicky. Je na nás, s jakým výsledkem se spokojíme. Zdáli dáme přednost kvalitnějšímu rozvrhu, jehož sestavení bude ovšem trvat velmi dlouhou dobu, nebo dáme přednost méně kvalitnímu, ovšem stále dobrému rozvrhu, sestaveného za přijatelný čas. [2]

2.3.1. Přerozvrhování

Do tohoto momentu jsme řešili problematiku vytváření celého kompletního rozvrhu. V praxi ovšem dost často nastává situace, kdy je potřeba upravit již existující rozvrh. Důvody pro toto přerozvržení vycházejí ve většině případů z praxe. Občas je nutné přeplánovat směny, pokud jeden ze zaměstnanců například onemocní atd. [8] [2]

Je opravdu důležité si uvědomit, že provádění těchto směn nesmí být prováděno bez jisté dávky šetrnosti a opatrnosti. Pro příklad, není vhodné kvůli nemoci jednoho zaměstnance přeplánovat směny všem ostatním zaměstnancům na celý měsíc, nehledě na to, že je nutné dodržet zákoník práce, zejména §85. V podstatě lze zjednodušeně říct, že je nutné provádět dodatečné přerozvrhování směn s takovou dávkou opatrnosti, aby bylo změnami zasaženo co nejméně ostatních zaměstnanců. [8] [2]

2.3.2. Omezení

Jelikož je nutné vymežit rozdíl mezi uskutečnitelným a neuskutečnitelným řešením, musíme, jak již bylo popsáno výše dodržet několik omezení. Tato omezení se dělí zpravidla na dvě skupiny, tvrdá a měkká. Porušení tvrdých podmínek vede k neuskutečnitelnému řešení, zatímco porušení měkkých podmínek vede ke snížení kvality rozvrhu. [2]

2.3.3. Tvrdá omezení [2] [8]

Tvrdá omezení jsou taková omezení, která v případě, že chceme dosáhnout uskutečnitelné² řešení, nesmí být za žádných okolností porušena. V opačném případě nazýváme rozvrh neproveditelným³. Tvrdá omezení zpravidla stanovují jasná a neporušitelná pravidla pro daný rozvrh, a musí být bez výjimky dodržena.

Seznam všech pravidel v rámci této konkrétní práce si popíšeme v následujících kapitolách, kde bude vše detailně vysvětleno.

² V anglické literatuře se můžeme setkat s pojmem feasible

³ V anglické literatuře se můžeme setkat s pojmem infeasible

2.3.4. Měkká omezení [8] [2]

Měkká omezení, na rozdíl od tvrdých, jsou taková omezení, která mohou být porušena. Bylo by dobré zdůraznit, že slovo mohou, je zde opravdu důležité. Čím více měkkých omezení při vytváření rozvrhu porušíme, tím bude rozvrh méně kvalitní. V úvahu musíme vzít také váhu omezení. Tato váha nám určuje, jak je pro nás omezení důležité. Snahou při vytváření rozvrhu je samozřejmě porušit omezení co nejméně, nebo alespoň s co nejmenší vahou. O výsledné kvalitě nakonec rozhoduje tzv. kritériální funkce, tedy součet všech penalizací za porušení pravidel. Mějme příklad, kdy při rozvrhování porušíme pouze jedno omezení s penalizací 500 a na druhé straně porušíme 10 omezení, které každé má hodnotu penalizace 2. Z toho vyplývá, že není vždy nejlepší řešení s nejnižším počtem penalizací, ale to kdy kritériální funkce je nejnižší, v našem případě 500 vs. 200.

Prioritu omezení určujeme, jak je patrné z předchozího odstavce pomocí nastavení váhy. S vyšší váhy roste naše preference dané omezení neporušit, avšak to nám nikdo nemůže zaručit. Naopak omezení s nižší vahou jsou pro nás v podstatě nedůležitá a považujeme je pouze jako něco navíc.

3. Analýza potřeb nemocnice

Analýza potřeb a požadavků při návrhu jakéhokoliv informačního systému, tedy i systému pro rozvrhování směn zdravotních sester, je základní, nedílnou a ve většině případů základní fází, kterou by měl každý analytik začít. Pouze systém, který je navržen a implementován podle požadavků uživatelů dokáže uspokojit všechny jejich potřeby.

Při analýze je nutné brát na zřetel potřeby v první řadě zaměstnance, který bude se systémem pracovat, v našem případě vrchní sestry. Pro toho zaměstnance nesmí být práce se systémem v žádném případě zbytečně složitá nebo dokonce obtěžující, ale naopak má mu práci ulehčit. Další skupinou uživatelů, kteří budou do systému přistupovat, budou dotyční zaměstnanci, pro které budou rozvrhy vytvářeny, a uživatel z účetního oddělení. Tyto dvě skupiny ale nebudou tvořit nejdůležitější uživatele. Jejich role v systému bude v podstatě pouze omezená a jejich aktivity se omezí na přístup pouze ke čtení.

3.1.1. Postup analýzy

V první řadě je potřeba zmapovat prostředí, ve které bude informační systém používán. Z této analýzy je například možné rozhodnout, která část systému je primární a tedy na kterou část systému je zapotřebí klást větší důraz, nebo zjištění počítačové gramotnosti uživatelů kvůli co nejpřívětivěji navrženému rozhraní a posloupnosti úkonů.

Mezi další části analýzy patří identifikace typu uživatelů a poté identifikace uživatelů jako takových. Následně je vhodné shromáždit informace o aktuálně používaném systému. Nejvhodnější metodou sběru těchto dat je série rozhovorů, během kterých bude návrhář shromažďovat informace o tom, jaký obsah a funkčnost považuje za důležitou pro chod informačního systému, nebo jeho části. Podle těchto informací je vhodné uzpůsobit navrhovaný informační systém. Existuje několik obecných, a pro návrháře důležitých otázek:

- k čemu bude informační systém sloužit,
- jaké jsou požadavky na informace,
- jaká je technická zdatnost uživatelů,
- zda uživatelé již podobný systém používají,
- jak by měla aplikace vypadat, aby byla co uživatelsky co nejpřívětivější.

Po vyhodnocení odpovědí na tyto otázky bude pro vývojáře jednodušší navrhnout informační systém tak, aby uživatelům vyhovoval co nejvíce a nevyžadoval změnu způsobu jejich práce.

3.2. Analýza prostředí

Tento systém pro rozvrhování směn zdravotních sester bude primárně určen pro společnost NH Hospital, a.s. Jedná se o nemocniční zařízení v Hořovicích v okrese Beroun. Nyní si rozvrhování směn řeší každé oddělení separátně vlastními silami, ale to tento nový systém změní. V první fázi bude systém nasazen na novorozeneckém oddělení. Jedná se o oddělení, na kterém slouží zdravotní sestry v nepřetržitém provozu. V každý okamžik musí být na oddělení alespoň dvě zdravotní sestry, kvůli poskytnutí prvotřídní péče. V nynějším systému jsou směny rozvrhovány přinejmenším nahodile. Není dodržována žádná periodicitata jako například krátký a dlouhý týden. To samozřejmě přispívá k nepřehlednosti a nespokojenosti zaměstnanců.

U zdravotních sester musí být možné odlišit sestru, která je právě v adaptačním procesu, tedy zaškoluje se a učí se, jak to na oddělení funguje. Tato setra je vždy přidělena jedné zkušené sestře, která je pro ní lektor a zodpovídá za její práci. Sestra v tomto procesu chodí na směny vždy s danou sestrou a nezapočítává se do obyčejného personálu, zjednodušeně můžeme říct, že je na pracovišti navíc.

Ve všední dny pracuje na oddělení také sanitář, či sanitářka, a to vždy pouze v ranní směně v režimu pondělí až pátek, 7:00 až 15:30. Nad těmito zaměstnanci je vrchní sestra oddělení, která zajišťuje plánování směn. To je z pohledu našeho systému uživatel číslo 1.

Vrchní sestra po ukončení každé pracovní periody provede kontrolu odpracovaných hodin a vytvoří výkaz. Tento výkaz je poskytnut účetnímu oddělení, kde slouží jako podklad pro výpočet měsíčních mezd zaměstnanců. Tyto výpočty provádí již zavedený účetní systém, který má ovšem rozhraní pro zdroj externích dat z našeho nového systému.

Veškeré analýzy budou konzultovány přímo s touto hlavní sestrou. Ona, jako v podstatě nejstěžejnější uživatel systému, nám může poskytnout nejdůležitější poznatky ohledně návrhu a vývoje. Ona je také osobou nejpovolanější v problematice rozvrhování směn zdravotních sester s ohledem na dodržování zákoníku práce.

3.3. Analýza současného řešení

Analýza současného stavu řešení je nezbytnou součástí vývoje nového systému. Analytik má možnost v rámci této analýzy zjistit, jaké jsou například zvyklosti zaměstnanců, kteří v našem případě navrhnou rozvrhy směn, a na základě těchto poznatků pokud možno co nejvíce přiblížit fungování nového systému tomu stávajícímu. Jsou ale ovšem také případy, kdy toto možné není, například pokud je původní systém příliš odlišný tak, jak to bude s největší pravděpodobností v našem případě. Jako další výhody této analýzy je i fakt, že analytik může v úvodní fázi identifikovat případné chyby a nedostatky, na které může v následné fázi návrhu reagovat, a vytvořit tak zcela fungující systém.

3.3.1. Aktuální řešení

Při analýze bylo zjištěno, že v rámci celé nemocnice nefunguje jeden centrální systém, nebo alespoň jakákoliv norma pro vytváření rozvrhů. Každá vrchní sestra používá svoji vlastní metodu, což v neposlední řadě vede k nekompatibilitě mezi jednotlivými rozvrhy. Na oddělení, které je pro nás v rámci práce stěžejní, se využívá k plánování směn pouze tabulka vytvořená v MS Excel. Veškeré plánování směn provádí vrchní sestra sama podle svých zkušeností. Toto řešení je nejen zdlouhavé, ale také nesystematické. V případě jakékoliv změny musí vrchní sestra ručně přeplánovat spoustu směn, a je třeba říct, že změna jakékoliv směny vede zákonitě ke spoustě změn ostatních zaměstnanců. Bez nějaké formy optimalizačního algoritmu je toto řešení velmi neúspěšné a sestru nepříjemně zatěžuje. O důvodech, proč v takto velkém zařízení jednotný systém není, můžeme pouze spekulovat a rozhodně rozbor těchto důvodů není součástí práce.

Směny jsou v tomto případě nerovnoměrně rozvržené. To znamená, že není dodržena pro tento druh směn přijatelná forma krátký – dlouhý týden viz tabulka číslo 5.

Tabulka 5 - krátký dlouhý týden

	Pondělí	Úterý	Středa	Čtvrtek	Pátek	Sobota	Neděle
1. týden	Směna	Směna	Volno	Volno	Směna	Směna	Směna
2. týden	Volno	Volno	Směna	Směna	Volno	Volno	Volno

Takto rozvrženými směnami zajistíme rovnoměrné rozvržení směn. Dále z tohoto systému rozvrhování plyne, že zaměstnanec má vždy dva víkendy v měsíci volné, což v nynějším stavu není. Nynější stav je takový, že jsou zdravotní sestry, které mají například volný pouze jeden víkend v měsíci, nebo mají v pátek noční. Z toho plyne, že jiná sestra musí mít volných víkendů více. Samozřejmě, pokud je to nutné z důvodu onemocnění jiné zdravotní sestry, nejedná se o problém. Tato situace ale bohužel nastává v plném stavu zdravotních sester. Tomuto by měl, nebo alespoň mohl nový systém zabránit.

3.3.2. Výhody a nevýhody

Z tohoto systému plyne snad pouze jedna výhoda. Touto výhodou by mohla být to, že tento systém nemá v podstatě žádné nároky na IT infrastrukturu a na výpočetní výkon.

Oproti tomu ovšem převládá dlouhá řada nevýhod, mezi které mohou patřit:

- veliké nároky na čas strávený rozvrhem,
- špatná optimalizace při přeplánování rozvrhu,
- absence periodického střídání směn – krátký dlouhý týden,
- absence jednotného formátu pro další zpracování,
- provozní - nadržování skupině oblíbených sester na úkor ostatních.

3.3.3. Závěr

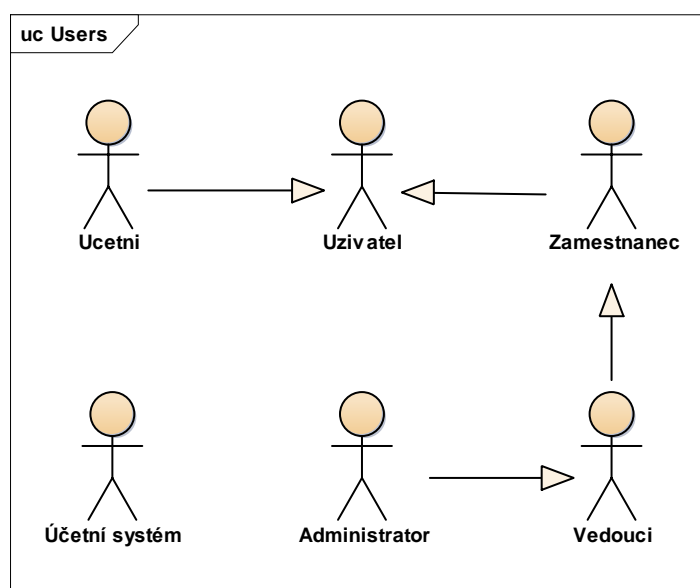
Z analýzy popsané výše jednoznačně vyplývá, že nynější systém rozvrhování směn je naprosto nevyhovující, a do jisté míry vede k nespokojenosti zdravotních sester s takto rozvrženými směnami. Bohužel, takto nespokojené sestry nemohou odvádět svou odbornou práci na 100 %, což vede k nespokojeným zákazníkům, v našem případě pacientům nemocnice. Nový systém by měl přinést kvalitní rozvrhy, které nestraní žádnému ze zaměstnanců. Také díky novému systému klesne čas strávený rozvrhování směn, a vrchní sestra se bude moci věnovat činností v rámci svého oddělení, což by mělo mít za následek rapidní zlepšení úrovně poskytované péče, což pocítí zejména pacienti.

3.4. Definování uživatelů

Definování uživatelů a jejich práv je nezbytnou součástí návrhu informačního systému. Je to důležité také z důvodu jiných nároků na systém jako takový. Každý uživatel může používat pouze část systému a do ostatních nemusí, nebo nesmí mít přístup. V navrhovaném systému bude figurovat několik druhů uživatelů s odlišnými právy:

- administrátor aplikace,
- vedoucí pracovník,
- zaměstnanec,
- pracovník účtárny,
- účetní systém.

Znázornění uživatelů a jejich dědičnosti je na obrázku č. 3



Obrázek 3 - dědičnosti uživatelů

3.4.1. Administrátor aplikace

Jedná se o uživatele s nejvyšším možným oprávněním, které v systému bude. Tento uživatel bude mít možnost spravovat veškerá data a nastavení, které v systému budou. Tento přístup bude mít s největší pravděpodobností osoba, která bude poskytovat uživatelskou podporu pro všechny ostatní uživatele rozvrhovacího systému

3.4.2. Vedoucí pracovník

Tato uživatelská role bude přidělena vrchní sestře, která bude zodpovědná za rozvrhování směn svých podřízených zdravotních sester. Úkolem tohoto uživatele bude:

- správa svých podřízených zdravotních sester,
- generování rozvrhů směn,
- dodatečné úpravy a přeplánování,
- úpravy omezujících podmínek pro vytváření rozvrhu.

3.4.3. Zaměstnanec

Role řadového zaměstnance bude přidělena každé zdravotní sestře, které se bude rozvrh týkat. Tento uživatel bude mít pouze několik málo možností jak se systémem pracovat, z nichž většina je pouze jako náhled. Možné činnosti:

- potvrzení vygenerovaného rozvrhu,
- návrh úprav vygenerovaného rozvrhu – plánovaná absence, doktor atd.,
- zobrazení svého aktuálního rozvrhu.

3.4.4. Pracovník účtárny

Tuto roli dostanou uživatelé účetního oddělení, které bude zpracovávat pracovní výkazy zdravotních sester z důvodu výpočtu odpracovaných směn a následné mzdy. Možná aktivity uživatele jsou:

- zobrazení přehledu směn zaměstnance,
- vygenerování souhrnu za celý měsíc pro všechny zaměstnance oddělení.

3.4.5. Účetní systém

Uživatelská role účetního systému reprezentuje pouze neživou entitu již existujícího informačního systému, který má možnost z našeho nového systému získat automaticky data související s odpracovanými směnami zdravotních sester.

3.5. Analýza požadavků

V této části analýzy se budeme věnovat analýzám požadavků uživatelů na nový rozvrhovací systém. Bez analýzy těchto požadavků je sice možné vytvořit nový systém, avšak není vůbec jisté, jak moc by byl pro uživatele uživatelsky přívětivý. Touto analýzou zjistíme, jak by měl systém vypadat a co by měl umět. Pouze uživatel, který bude se systémem pracovat, může toto definovat. Všechny následující poznatky byly získány v průběhu několika schůzek s vrchní sestrou oddělení.

3.5.1. Část evidence zdravotních sester

V této části by mělo být možné spravovat údaje o zdravotních sestřích, pro které se budou výsledné rozvrhy navrhovat. Mezi údaje patří zejména osobní údaje zaměstnance, jako jméno, příjmení, osobní číslo, pozice, druh pracovního úvazku, rozlišení zda zdravotní sestra bude pracovat noční směny, či víkendy. Přístup do této části bude mít pouze uživatel s oprávněním vedoucího pracovníka a vyššími.

Tabulka 6 - požadavky na vstupy a výstupy

Údaje	Popis	Souhrny k zobrazení
Osobní údaje	Jméno, příjmení atd.	Odpracované směny
Druh úvazku	Plný, zkrácený	Využitá dovolená
Noční	ano/ne	
Víkendy	ano/ne	
Pozice	Sestra, Ošetřovatel atd.	

3.5.2. Část administrace omezení

Pro rozvržení optimálních směn je nutné definovat v systému několik omezení, která je nutné dodržet. Jak bylo popsáno výše, existují dva typy omezení. Měkké, které je možné porušit za cenu horšího rozvrhu, a tvrdé, které není možné porušit v žádném případě. V této části systému se budou tato omezení definovat.

3.5.3. Tvrdá omezení

Jak již bylo popsáno výše, tvrdá omezení jsou taková, která je možno porušit, ovšem jako cena tohoto porušení je méně optimalizovaný rozvrh. Každé omezení má svou předem danou váhu, která určuje, jakou má pravidlo prioritu.

Rozvržení maximálně jedné směny na den

Každá zdravotní sestra může mít pouze jednu směnu denně. Pokud například sestra slouží jeden den ranní směnu, není možné, aby měla rozvrženou také noční. V práci by potom podle výše uvedeného příkladu 24 hodin, což je nejen nepřipustné z hlediska vytížení zaměstnanců, ale také to porušuje zákoník práce. V podstatě můžeme říct, že v jeden den může mít zdravotní sestra pouze **jeden** začátek směny.

Všechny potřebné směny jsou obsazeny

Na každý den a směnu je určený počet zaměstnanců, který musí být splněn. Kdyby nebyl požadovaný počet zaměstnanců splněn, došlo by buď k nedostatečným počtům personálu, nebo naopak k přeplnění.

Minimální rozestup mezi směnami

Do rozvrhu nestačí přidat směny jen tak ledabyle podle požadované obsazenosti, ale je potřeba mít na mysli i horizontální stránku věci. Tou je následnost směn, což neznamená nic jiného, než že mezi naplánovanými směnami pro zaměstnance musí být časová prodleva alespoň 12 hodin dle zákoníku práce §90 (1) [9], který doslova říká: „Zaměstnavatel je povinen rozvrhnout pracovní dobu tak, aby zaměstnanec mel mezi koncem jedné směny a začátkem následující směny nepřetržitý odpočinek po dobu alespoň 12 hodin po sobe jdoucích během 24 hodin“.

3.5.4. Měkká omezení

U měkkých omezení předpokládáme, že všechny při rozvrhování nesplníme. Jedná se o omezení, které neznemožní vytvoření rozvrhu, avšak přispějí k jeho nižší kvalitě a horší optimalizaci. V prvním přístupu k měkkým podmínkám je každé porušení podmínky „potrestáno“ penalizací, která je u každého omezení různá podle priority. Zde se jedná o to, aby součet penalizací byl co nejnižší.

V druhém přístupu k podmínkám mají všechna pravidla naprosto stejnou váhu, tedy penalizaci, a jde tedy o to, aby bylo dosaženo co nejnižšího počtu porušených podmínek. Z toho vyplývá, že pro nás žádné omezení není důležitější než jiné.

Maximální (minimální) počet přiřazených směn

Určuje maximální a minimální počet přiřazených směn pro jednoho zaměstnance za rozvrhovanou periodu. V optimálním případě by neměla mít zdravotní sestra žádné přesčas, a neměla by mít ani méně než 160 odpracovaných hodin.

Maximální (minimální) počet pracovních dnů v řadě

Určuje maximální a minimální počet pracovních dnů v řadě, během kterých může být danému zaměstnanci přiřazena směna. Toto omezení je důležité zejména z hlediska splnění podmínek zákoníku práce, který jednoznačně určuje, že po určitém počtu odpracovaných hodin musí následovat daný počet dnů pracovního klidu.

Maximální (minimální) počet volných dnů v řadě

Určuje maximální a minimální počet volných dnů v řadě, během kterých nemá zaměstnanec přiřazenu žádnou směnu. Toto omezení je v podstatě obdobou předchozího.

Maximální (minimální) počet po sobě jdoucích pracovních víkendů

Určuje maximální a minimální počet přiřazených pracovních víkendů v řadě. Toto omezení je důležité pro zachování spravedlivého pracovního rozvrhu pro všechny zaměstnance. Je důležité, aby se nestala situace, že jedna sestra bude sloužit například 3 pracovní víkendy a jiná pouze jeden.

Maximální (minimální) počet pracovních víkendů za měsíc

Určuje maximální a minimální počet přiřazených pracovních víkendů během jednoho měsíce. V optimálním případě při dodržení krátkého a dlouhého pracovního týdne, má každá zdravotní sestra dva pracovní a dva volné víkendy.

Celý pracovní víkend

Omezení stanovuje, že pokud má zdravotní sestra jednu víkendovou směnu, rozvrhne systém směnu i na druhý den víkendu. S tímto je úzce spjato i následující omezení.

Identické směny během víkendu

Určuje to, že všechny směny během víkendu budou identické.

3.5.5. Část generování rozvrhů

Tato část systému bude sloužit pro samotnou práci s pracovními rozvrhy. Uživatel s právy pro tuto správu zde bude generovat rozvrhy na následující období, včetně případného přerozvrhování z důvodů nemocí, požadavků sester atd. Systém by měl obsahovat tabulku, kde budou zobrazeny vygenerované směny jednotlivých pracovníků. Samozřejmě je export těchto rozvrhů do Excelu či PDF.

3.5.6. Požadavky na dostupnost

Celá aplikace bude dostupná z vnitropodnikové sítě daného nemocničního zařízení. Databáze bude umístěna na centrální server, který bude dodán zadavatelem zakázky. Přístup k aplikaci bude rozdělen na dvě části.

Přístup pomocí tlustého klienta

Přístup do tohoto klienta bude mít pouze pracovník, který bude dané rozvrhy zpracovávat, a hlavní administrátor aplikace. Aplikaci bude nutné nainstalovat přímo na počítač vedoucího pracovníka. Při výběru platformy je nutné brát na zřetel kompatibilitu s operačním systémem používaným nemocničním zařízením.

Přístup přes webový prohlížeč

Tento přístup bude sloužit zejména pro zaměstnance, pro který bude daný rozvrh vytvářen. Touto cestou budou mít možnost zobrazit si svůj vygenerovaný rozvrh a budou v něm moci navrhnout dodatečné úpravy. Pokud bude rozvrh podle jejich představ, provedou potvrzení rozvrhu. Zvolený druh přístupu je vhodný zejména pro zaměstnance, kteří budou mít možnost přistupovat k rozvrhům i z jiných míst než v nemocnici.

3.5.7. Požadavky na funkce

Tyto požadavky, získané od budoucích uživatelů, nám slouží k jasnému definování funkcí budoucího rozvrhovacího systému, a slouží nám jako podklad pro další práci. Naším cílem při analýze požadavků je zejména:

- získat přehled o představách budoucích uživatelů,
- zjistit proč uživatelé daný systém potřebují,
- zjistit, co od systému očekávají.

Nejlepší cesta jak zjistit tyto požadavky jsou pravidelné schůzky se všemi zúčastněnými stranami ve fázi návrhu i následného vývoje daného systému. [10]

Souhrn funkčních požadavků:

- Zadávání a evidence zaměstnanců
- Nastavování směn
- Určení období pro tvorbu rozvrhu
- Definování možností rozvrhu – počty směn, víkendy atd. viz výše v kapitolách.
- Přerozvrhování směn
- Exporty rozvrhů do Excelu a PDF
- Úpravy vygenerovaného rozvrhu
- Potvrzení odpracovaných směn
- Informování zaměstnanců e-mailem

Souhrn obecných požadavků

- Systém bude navržen a implementován v duchu OOP⁴
- Aplikace bude dostupné prostřednictvím tlustého klienta pro vedoucího pracovníka
- Aplikace bude dostupná prostřednictvím webového prohlížeče pro každého zaměstnance
- Systém vyžaduje přihlášení – přístup pouze pro autorizované uživatele
- Aplikace musí mít uživatelsky přívětivý interface a intuitivní ovládání

3.5.8. Analýza systému směn

Systém rozvrhování směn může být v každém zařízení jiný. Z tohoto důvodu je zahrnuta i analýza současného stavu systému. Tato analýza je důležitá pro pochopení návyků zaměstnanců. Na základě této analýzy bude navržen odpovídající systém, který bude lépe korespondovat s dnešními trendy.

⁴ OOP – **O**bjektově **O**rientované **P**rogramování

Druhy zaměstnanců

- Ošetřovatel/ošetřovatelka
 - Směny: ranní
 - Pracovní doba: 7:00 – 15:30
 - Nutnost dohledu: bez dohledu
- Zdravotnický asistent
 - Směny: jednosměnný provoz
 - Pracovní doba: 6:00 – 18:00
 - Nutnost dohledu: nutný dohled zdravotní sestry
- Zdravotní sestra
 - Směny: nepřetržitý provoz
 - Pracovní doba: 6:00 – 18:00 / 18:00 – 6:00
 - Nutnost dohledu: bez dohledu
- Zdravotní sestra v zácvičku
 - Směny: jednosměnný provoz
 - Pracovní doba: 6:00 – 18:00
 - Nutnost dohledu: nutnost dohledu zdravotní sestry
- Vrchní sestra
 - Vedoucí sestra oddělení
 - **Uživatel zodpovědný za tvorbu rozvrhů směn**
 - Není součástí rozvrhu
- Staniční sestra
 - Nadřízená vrchní sestry
 - Není součástí rozvrhu
- Hlavní sestra
 - Vedoucí pracovník všech zdravotních sester
 - Není součástí rozvrhu

Druhy směn

- Ranní
 - Pracovní doba: 7:00 – 15:30
 - Pracovníci: ošetřovatelé a nezdravotnický personál
- Denní
 - Pracovní doba: 6:00 – 18:00
 - Pracovníci: zdravotní sestry
- Noční
 - Pracovní doba: 18:00 – 6:00
 - Pracovníci: zdravotní sestry

Systém rozvrhování směn je nyní celkem nevyhovující. Směny nejsou sestavovány podle žádného opakujícího se systému a jsou náhodné. Výjimkou není ani celkem časté střídání druhu směn, což vede k nepřiměřenému zatěžování zdravotních sester, které nemají možnost si na daný druh směny zvyknout. V nynějším rozvrhu se často stává, že zdravotní sestra má volný víkend, ale v pátek má noční směnu, což znamená, že sobota je pro ní den po směně, či případ kdy má volnou sobotu a neděli noční, což opět nabourává například její víkendový program. Jedná se pouze o několik anomálií, avšak výčet všech by zabral mnohem více času a není pro nás tak důležitý.

V našem systému uvažujeme rozvrhování pouze denních a nočních směn, ranní směny se řídí zavedeným pravidlem 5 * 8, což není potřeba zahrnovat do rozvrhu a tudíž to není pro naši problematiku důležité.

Směny by měly být rozvrženy podle pravidla krátký – dlouhý týden, které bylo popsáno v předešlých kapitolách. Tímto systémem zajistíme rovnoměrné rozvržení směn, spravedlivý počet odsloužených víkendů v rámci dané periody a v neposlední řadě více zmíněné kvalitní střídání druhů směn.

3.6. Závěr analýzy

I přes to, že původní řešení problematiky rozvrhování směn personálu nemocnice je ve své podstatě funkční, v žádném případě se nejedná o vyhovující, nebo dokonce optimální řešení. Stávající řešení se jeví nevhodné zejména kvůli následujícímu:

- nesystematičnost při sestavování rozvrhu směn,
- vysoké nároky na vedoucího pracovníka, který směny rozvrhuje,
- obtížné přerovrhování směn,
- špatná optimalizace při přeplánování rozvrhu,
- absence periodického střídání směn – krátký dlouhý týden,
- absence jednotného formátu pro další zpracování,
- provozní - nadržování skupině oblíbených sester na úkor ostatních.

Tyto problémy by nově navrhovaný systém měl řešit, a je nutné brát na toto zřetel. Pro vedoucího pracovníka by práce se systémem neměla neúměrně zatěžující, a neměla by ho okrádat o čas důležitý k řešení dalších provozních problémů v rámci oddělení.

Na základě analýz byly definovány skupiny uživatelů, které budou se systémem přímo, i nepřímo pracovat, a pro které je nutné systém přizpůsobit. Směny, do kterých budou tito zaměstnanci, nebo skupiny zaměstnanců, obsazováni byly do analýzy zahrnuty také. Dále jsme definovali pravidla a omezení pro tvorbu rozvrhu, která byla na základě řešení problematiky rozvrhování směn zařazena dvou kategorií, jak již bylo popsáno v předchozích kapitolách.

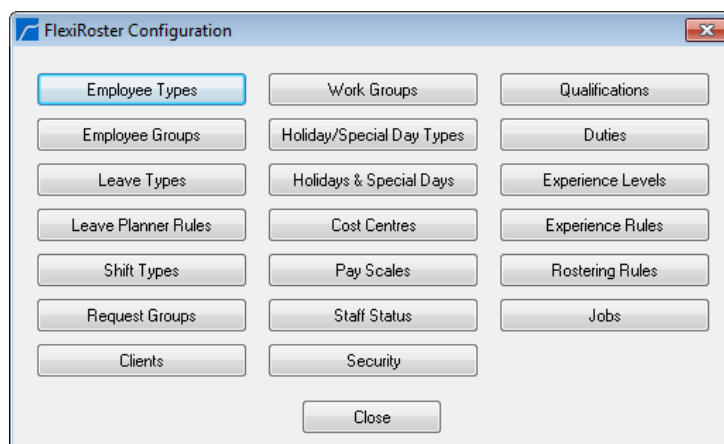
Veškeré tyto analýzy nám poskytnou kvalitní základ a oporu při následné fázi návrhu nového rozvrhovacího systému. Nový rozvrhovací systém by měl vést zejména ke spokojenějším zaměstnancům, a ke zkvalitnění poskytované péče a tím také ke spokojenějším pacientům, o které jde nejvíce.

4. Dostupný software na trhu

Na trhu se softwary sloužící pro rozvrhování směn je možné najít požadovanou aplikaci. U těchto aplikací je většinou komplikace v případě, že bychom chtěli výrazné úpravy vzhledu, funkčnosti nebo následné provázanosti na již fungující systémy, kterým je v našem případě již fungující účetní software pro výpočet mezd zaměstnanců. Ovšem je možné najít i open source software, který je možné si upravit, avšak takovéto úpravy si žádají vývojáře se zkušenostmi v daném programovacím jazyce, či přímo s tímto open source softwarem. Budoucí vývojový tým bude mít v podstatě na výběr ze dvou možností. Buďto naprogramovat danou aplikaci od základu, nebo využít nějaký ze jmenovaných, nebo dalších jiných zde nejmenovaných, naprogramovaných softwarů. Jako podpora při jejich rozhodování poslouží následné analýzy a návrhy.

4.1. Flexi Roster [11]

Jedná se o již už hotový rozvrhovací systém australské firmy 1800 4 ROSTER, který je k dispozici za měsíční poplatek **2 dolary** za uživatele ve standartní verzi, a **7,5 dolaru** za rozšířenou verzi. Systém je komplexní a poskytuje všechny možné funkce pro sestavování rozvrhů směn. Navíc je tento software navržen přímo pro potřeby nemocničních zařízení, které mají určitá svá specifika, která budou analyzována v dalších kapitolách. Nicméně jako jistou nevýhodu lze vidět nemožnost dalších úprav přímo pro potřeby z hlediska zadání našeho rozvrhovacího systému. Mezi další nevýhodu tohoto softwaru patří pro uživatele na první pohled složitá konfigurace, viz obrázek číslo 4. To je samozřejmě do jisté míry dáno rozsáhlostí problému rozvrhování směn zdravotních sester jako takového.



Obrázek 4 - možnosti konfigurace [11]

Další komplikací z hlediska našeho návrhu je nemožnost přístupu k vygenerovaným rozvrhům prostřednictvím webového prohlížeče, tak jak je to v požadavcích na nový systém. Zaměstnanci by neměli například možnost navrhnout požadované změny takovým způsobem, který je požadován v zadání projektu.

The screenshot shows the 'FlexRoster: Assigning Staff to Roster' application. The main window displays a grid where rows represent staff members and columns represent dates from 17/11 to 26/12. Each cell in the grid contains a code indicating the staff member's status or shift type for that day. For example, 'ADN' might stand for a day shift, 'NIN' for a night shift, and 'DCN' for a day cover. The staff list includes names like Adrian Adams, Bryden Bouрке, and Henry Harrison. At the bottom of the window, there is a summary table for shift counts and experience rules.

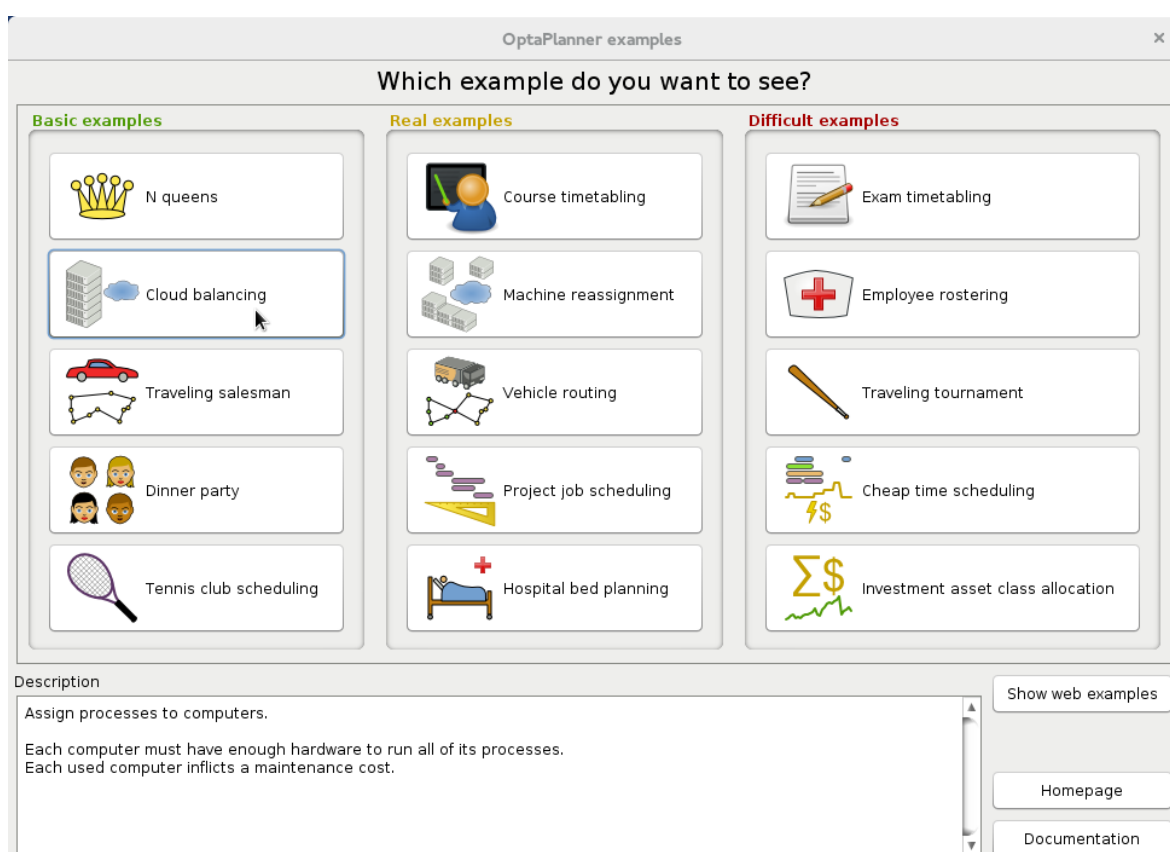
	17/11	18/11	19/11	20/11	21/11	22/11	23/11	24/11	25/11	26/11	27/11	28/11	29/11	30/11	1/12	2/12	3/12	4/12	5/12	6/12	7/12	8/12	9/12	10/12	11/12	12/12	13/12	14/12	15/12	16/12	17/12	18/12	19/12	20/12	21/12	22/12	23/12	24/12	25/12	26/12									
Early 7am (DCN)	5	3	5	5	5	6	6	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5					
Early - 12 Hours (ZCN)	1	1	1	1	2	2	1	1	1	1	1	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
Late - 12:30pm (DNN)	0	0	1	1	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Late - 2:30pm (DRN)	3	4	4	4	4	4	5	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
Night - 12 Hours (TEN)	1	1	1	1	2	2	1	1	1	1	1	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Night - Short (NIN)	2	2	2	3	2	2	3	2	2	2	3	2	2	2	2	3	2	2	2	3	2	2	3	2	2	2	3	2	2	3	2	2	2	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	
Total:	12	11	19	18	20	20	20	14	13	18	18	19	21	20	14	13	17	18	19	20	22	14	13	18	18	17	18	18	17	18	18	17	18	18	17	18	18	17	18	18	13	13	13	13	13	13	13	13	

Obrázek 5 - rozpis směn [11]

Avšak i přes tyto vyjmenované nevýhody, či komplikace software poskytuje kvalitně zpracované rozvrhy směn. Formát rozpisu směn je na obrázku číslo 5. Jednotlivé zkratky znázorňují určitý typ směny, což se dá v systému konfigurovat. V případě potřeby přeplánování se ručně označí směny k úpravě, a následně se provede automatické přiřazení nových směn.

4.2. OptaPlanner [12]

OptaPlanner je jeden s **open source** softwarů pro řešení našeho problému. Na rozdíl od aplikace Flexi Roster (dále FR), se nejedná o plánovač směn zdravotních sester jako takový, ale jedná se o software, který řeší optimalizační úlohy obecně. Výhoda FR oproti tomuto softwaru je hlavně ta, že FR je navržen a naprogramován přímo pro tuto problematiku, tudíž vše je tomuto přizpůsobené. Program OptaPlanner je nutné do jisté míry přizpůsobit. Tato na první pohled nevýhoda na druhou stranu dává možnost přizpůsobit si aplikaci tak, aby nám co nejlépe vyhovovala jak po stránce aplikační, tak i po stránce uživatelského rozhraní. Nicméně všechny tyto změny by musel udělat programátor s dobrou znalostí onoho softwaru, a to hlavně z důvodu, že tento systém je poměrně rozsáhlý.

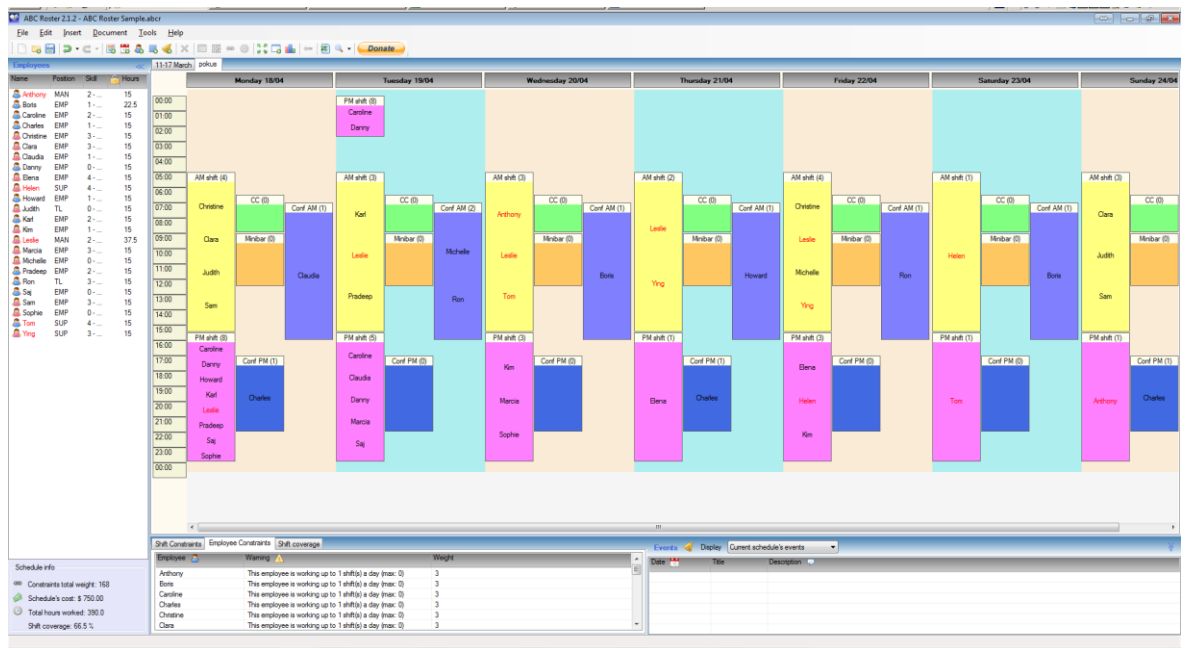


Obrázek 6 - OptaPlanner [12]

Na obrázku číslo 6 je názorně vidět, že tento software řeší i další optimalizační úlohy, které ale pro nás nejsou důležité. Nicméně by se software po jistých úpravách dal použít. Zobrazení naplánovaných směn je v podstatě totožné jako u FR; chybí ale i v tomto případě webová část aplikace, je možné ji ale doimplementovat. Software jako takový je **zdarma**, avšak úpravy si vyžádají investice, které je nyní těžké určit.

4.3. ABC Roster [13]

ABC Roster je aplikace poskytovaná **zdarma**, a to i pro korporátní zákazníky. Na rozdíl od aplikace OptaPlanner se nejedná o open source, takže není možné si aplikaci upravit. Je nutné pracovat s aplikací tak jak byla naprogramována. Z tohoto důvodu by bylo například obtížné doplnit ji o požadovanou webovou část pro zaměstnance. Velkou výhodou aplikace je její jednoduchost i přehlednost a to i při zachování plnohodnotných funkcí. Jedná se zejména o nastavování podmínek pro sestavování rozvrhů, i přehledný design uživatelského rozhraní. Aplikaci lze využít pro automatické plánování směn, ale celkem jednoduše a přehledně lze směny a rozvrhy duplikovat do dalších období pomocí šablon. Při malém množství zaměstnanců je i tato možnost celkem přijatelná.

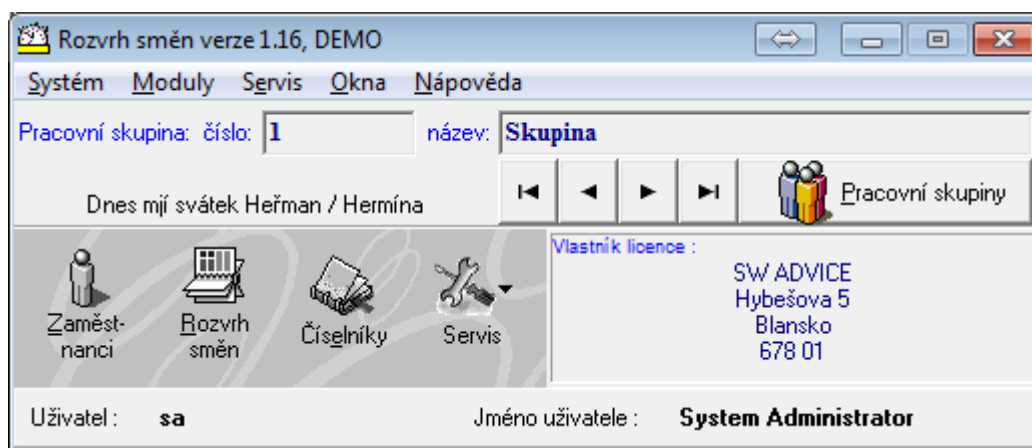


Obrázek 7 - ABC Roster

Na obrázku číslo 7 je znázorněno hlavní okno aplikace, ve kterém jsou zobrazeny rozvržené směny zaměstnanců v jednotlivých dnech. Rozvrh také kontroluje splnění podmínek při ručních úpravách a lze využít i automatického přeplánování. Bohužel tím že se jedná o uzavřený systém, algoritmy pro optimalizaci jsou pro nás neznámé.

4.4. Rozvrhy směn verze 1.16

Jedná se o plánovač české firmy SW Advice. Nicméně svým pojetím a rozsahem ho uvádím spíše jako možnou alternativu. Osobně bych tento software doporučil pouze například v rámci jednoho oddělení s malým počtem zaměstnanců. Rozhodně bych tento software nedoporučil pro větší korporaci s více zaměstnanci, kterou nemocnice rozhodně je.



Obrázek 8 - Rozvrhy směn verze 1.16

4.5. Závěr

Jak je patrné z předchozích kapitol, na trhu existuje několik sofistikovaných softwarů, které by bylo možné použít i v našem případě. V některých případech by to ale bylo za cenu upuštění od některých našich požadavků. Dalším ovlivňujícím faktorem je zde cena produktu. Je nutné spočítat, kolik by stál vývoj nové aplikace od základu v porovnání s měsíčními výdaji, jako je to například u aplikace Flexi Roaster. Při rozvrhu pro 100 osob a plné verzi aplikace činí měsíční nájemné **7,50 dolaru**. Při dnešním kurzu se jedná cca o **200 Kč** na osobu. Takže měsíční rozvrh směn vyjde zařízení na **20 000 Kč**, což není zanedbatelná částka. Ostatní uvedený software je sice zdarma, avšak případné úpravy do požadované podoby si vyžádají jisté investice, které ovšem je nyní poměrně složité vyčíslit.

Bude pouze na odpovědných osobách, aby na základě těchto podkladů a další podkladů a návrhů z této práce zvolili pro ně nejpříjemnější variantu. Ale je nutné si uvědomit, že každé řešení, které se nám nabízí má své klady a své zápory.

5.Návrh nového systému

Návrh informačního systému je proces tvorby nového informačního systému, které následuje jeho vývojem a implementací. Tento proces je řízen a má určitá pravidla a doporučení. Při dodržení těchto pravidel a doporučení, můžeme takový proces označit za metodiku. Metodika určuje, kdo, kdy co a proč má dělat během vývoje a provozu informačního systému. Zde je také namístě rozlišit od sebe několik následujících výrazů. [14]

- Metoda – říká, co je třeba dělat a kdy a využívá různé techniky. [14]
- Metodika – definuje souhrn principů a konceptů, pokrývající celý cyklus. [14]
- Technika – určuje jak dělat danou činnost. [14]
- Nástroj – slouží jako prostředek či pomůcka pro vývojáře. [14]

5.1. Postup návrhu

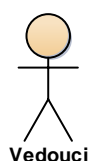
Při návrhu nového rozvrhovacího systému je důležité vycházet z předem vytvořených analýz. Pro splnění požadavků zákazníka, kterým je v našem případě nemocniční zařízení, je důležité, aby systém odpovídal jejich požadavkům jak na funkčnost, tak i na vzhled uživatelského rozhraní. Proto je nutné využít technologie vizualizace, kterým porozumí i člověk v oboru nezběhlý. Pro tuto problematiku výborně poslouží jazyk UML, který je všestranně použitelný. Jazyk využijeme zejména pro sestavení diagramu případů užití, kde jsou znázorněny aktivity jednotlivých uživatelů. Jako software pro tuto část práce byl zvolen Enterprise Architect, ve kterém lze vše demonstrovat.

Další součástí návrhu budou základní diagramy tříd. Tyto návrhy dávají konceptuální náhled na systém. Tyto modely tříd vycházejí z výše popsaných případů užití a jejich scénářů, které jsou sestavovány na základě uživatelských požadavků. Z těchto modelů bude vycházet databázové schéma, které bude vytvořeno na základě modelů tříd.

V dalších fázích návrhu systému bude sestaveno schéma aplikační logiky nového systému, návrh architektury systému, návrh uživatelského rozhraní a návrh hardwarového řešení celého projektu.

5.2. Diagramy případů užití

Use case diagram zobrazuje základní chování systému z pohledu uživatele. Výsledný diagram popisuje požadavky uživatele na daný systém, přičemž se jedná o do jisté míry abstraktní pohled. Diagram například neřeší posloupnosti akcí, nebo metody a postupy k provedení dané akce. Samotný use case diagram se skládá z několika mále komponent. Jednou z nich je aktér, znázorněný na obrázku číslo 9, který reprezentuje uživatele, který se systémem pracuje, ale může reprezentovat například i čas nebo CRON, což je časovač běžící na pozadí operačního systému a slouží pro automatické spuštění skriptů apod. [15]



Obrázek 9 - UC Aktér

Další komponentou použitou v Use Case diagramu je samotný use case, tedy samotný případ užití. Tento případ užití sestává z několika akcí, jejichž vykonání vede ke splnění nějakého předem daného cíle. V našem případě může být use case například akce při generování rozvrhu. V reálném systému se bude jednat o poměrně složitou a komplexní operaci, avšak v rámci Use Case diagramu se bude jednat o jednu úlohu. Grafické znázornění můžeme vidět na obrázku číslo 10 [15]



Obrázek 10 - UC případ užití

Jednotlivé aktéry a UC můžeme mezi sebou propojovat pomocí několika málo možných vazeb. Jedná se zejména o vazby: [15]

- Use – slouží pro základní propojení typové úlohy a aktéra
- Include – typová úloha obsahuje jinou typovou úlohu
- Extend – typová úloha rozšiřuje jinou typovou úlohu
- Generalizace – dědičnost mezi aktéry, ale i mezi typovými úlohami

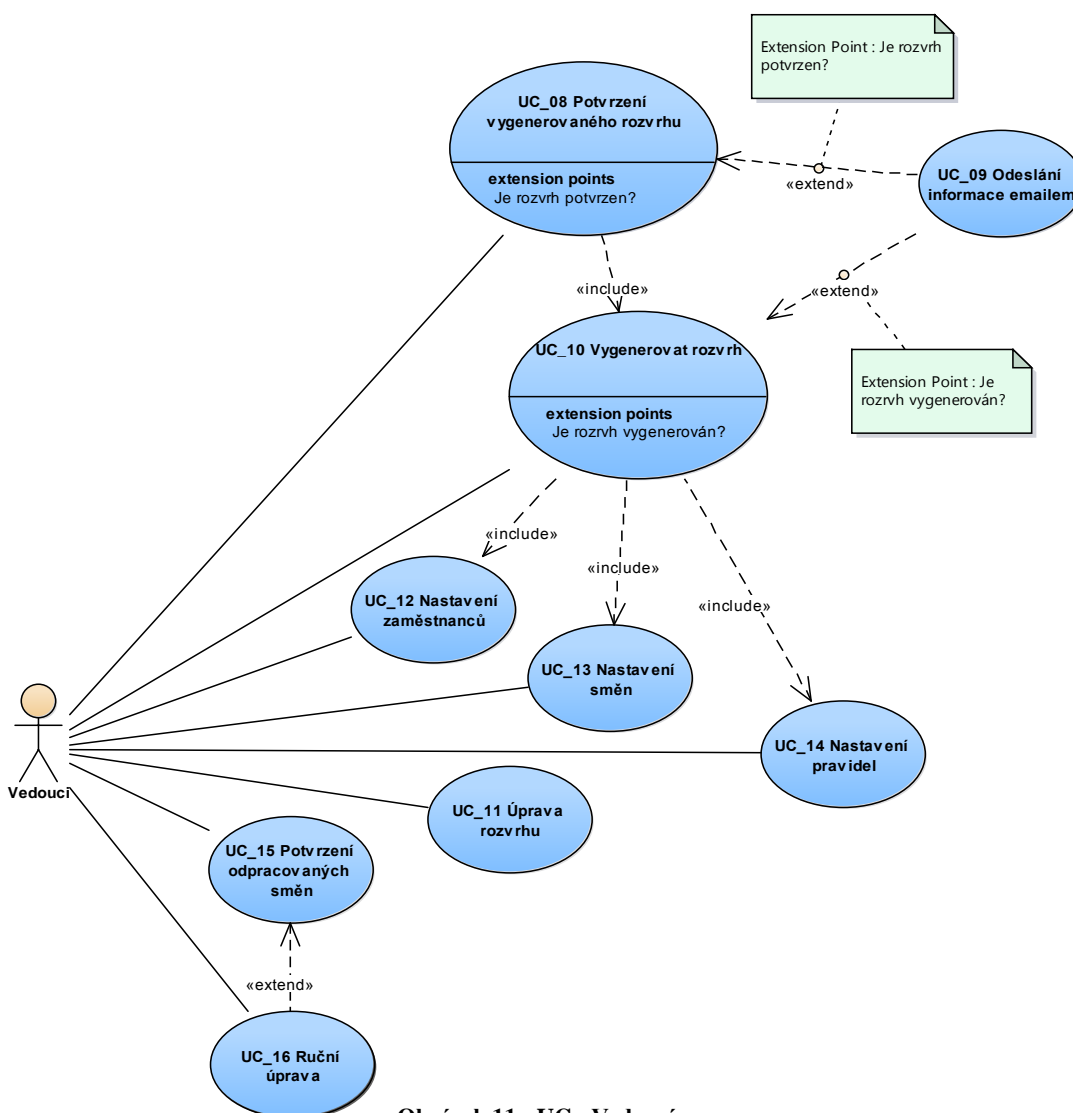
Z důvodu přehlednosti si rozdělíme případy užití na několik částí, rozdělených zejména podle uživatelů, nebo uživatelských rolí. Celý Use Case diagram bude samozřejmě přiložen v příloze této diplomové práce.

5.2.1. Všeobecné předpoklady

Všechny typové úlohy mají několik základních předpokladů. Jedná se zejména o předpoklad toho, že uživatel je přihlášený do systému, a že je ověřena jeho identita.

5.2.2. Vedoucí

Vedoucí pracovník je pracovník, který je zodpovědný za tvorbu rozvrhů, jak již bylo popsáno výše v předchozích kapitolách. Jeho akce a aktivity v navrhovaném systému jsou zobrazeny na obrázku číslo 11.



Obrázek 11 - UC - Vedoucí

Tato část diagramu popisuje obecné činnosti zaměstnance s rolí vedoucí. Následuje stručný popis typových úloh.

5.2.3. UC_08 Potvrzení vygenerovaného rozvrhu

Cílem splnění této typové úlohy je potvrzení vygenerovaného rozvrhu. Rozvrh je zaslán emailem všem zúčastněným zaměstnancům, kteří mají předem určený časový interval pro případný návrh změn, například z důvodů plánování dovolené apod. Rozvrh, který chceme potvrdit pro dané období, musí být již vygenerovaný. V případě, že rozvrh vygenerovaný není, přejdeme k typové úloze **UC_10 Vygenerovat rozvrh**.

Zjednodušený scénář:

1. Aktér inicializuje potvrzení vygenerovaného rozvrhu
2. Systém zkontroluje rozvrhy pro dané období
3. Systém provede potvrzení rozvrhu
4. Systém spustí UC_10 Vygenerovat rozvrh
5. Systém spustí UC_09 a zavře formulář

5.2.4. UC_09 Odeslání informace emailem

Tato typová úloha zajišťuje odesílání potřebných informací e-mailem. V našem případě se e-mailem odesílají informace o vygenerovaných a schválených rozvrzích.

5.2.5. UC_10 Vygenerovat rozvrh

Typová úloha zajišťuje proces generování nového rozvrhu pro dané časové období. Vygenerovaný rozvrh může být dále upravován podle potřeb zaměstnanců, či vedoucího pracovníka. Pro úspěšné splnění úlohy je nutné dodržet několik pravidel, zejména nastavení pravidel pro generování směn, tzv. měkká a tvrdá omezení rozvrhu, která jsou popsána výše. Samozřejmostí je také dostatek zaměstnanců v evidenci. Při nesplnění těchto podmínek spouští systém dané typové úlohy. Takto vygenerovaný rozvrh je dále odeslán všem zúčastněným zaměstnancům e-mailem.

Zjednodušený scénář:

1. Uživatel inicializuje vygenerování rozvrhu
2. Systém provede kontrolu zadaných dat
3. Zaměstnanci nenastaveni
4. Spuštění UC_12
5. Směny nenastaveny
6. Spuštění UC_13
7. Pravidla nenastavena
8. Spuštění UC_14
9. Systém provede generování rozvrhu
10. Systém zobrazí vygenerovaný rozvrh
11. Uživatel zavře formulář

5.2.6. UC_12, UC_13, UC_14 – nastavení

Tyto tři typové úlohy jsou záměrně shrnuty do jednoho odstavce. Tyto úlohy slouží k nastavování či evidenci zaměstnanců, pravidel pro tvorbu rozvrhu a pravidel pro tvorbu směn. Typové úlohy je možno splnit samostatně, či v rámci typové úlohy pro generování rozvrhu. Bez těchto nastavení není možné rozvrhu vytvořit.

5.2.7. UC_11 Úprava rozvrhu

V rámci této typové úlohy je možno již vygenerovaný rozvrh upravovat. Důvodem k této činnosti může být například náhlé onemocnění zaměstnance, kdy je potřeba pokrýt jeho směny jiným pracovníkem. Samozřejmě při dodržení všech tvrdých a co nejvíce měkkých omezení, které jsme nadeřinovali v předchozích kapitolách. V posledním případě je možné provést jemné ruční změny, které se neodrazí na automatickém plánování směn.

Zjednodušený scénář:

1. Uživatel inicializuje úpravu rozvrhu
2. Systém zobrazí formulář s rozvrhem
3. Uživatel provede požadované úpravy
4. Systém zobrazí výzvu k uložení záznamu
5. Uživatel nepotvrdí úpravy

6. Systém zahodí zadaná data
7. Uživatel potvrdí změny
8. Systém provede úpravu rozvrhu
9. Systém zavře formulář

5.2.8. UC_15 Potvrzení odpracovaných směn

Tato typová úloha se stane aktuální při ukončení aktuálního období pro vygenerovaný rozvrh. Uživatel s rolí vedoucí provede kontrolu reálně odpracovaných směn a poté případné potvrzení či nepotvrzení. Tato typová úloha nám zajistí aktuální počet odpracovaných hodin pro další zpracování. Uvažujme například situaci, kdy v již rozpracovaném měsíci onemocní ke konci období zdravotní sestra. Není možné provést kompletní přeplánování rozvrhu. V pohledu systému sestra bude mít odslouženo, avšak realita bude jiná. V rámci této typové úlohy bude možné využít typovou úlohu **UC_13 Ruční úprava**.

Zjednodušený scénář:

1. Aktér inicializuje potvrzení odpracovaných směn zaměstnanců
2. Systém zobrazí formulář se směnami
3. Uživatel potvrdí rozvrh směn
4. Aktér inicializuje úpravu směn
5. Systém spustí UC_16
6. Systém provede potvrzení rozvrhu a zavře formulář

5.2.9. UC_16 Ruční úprava

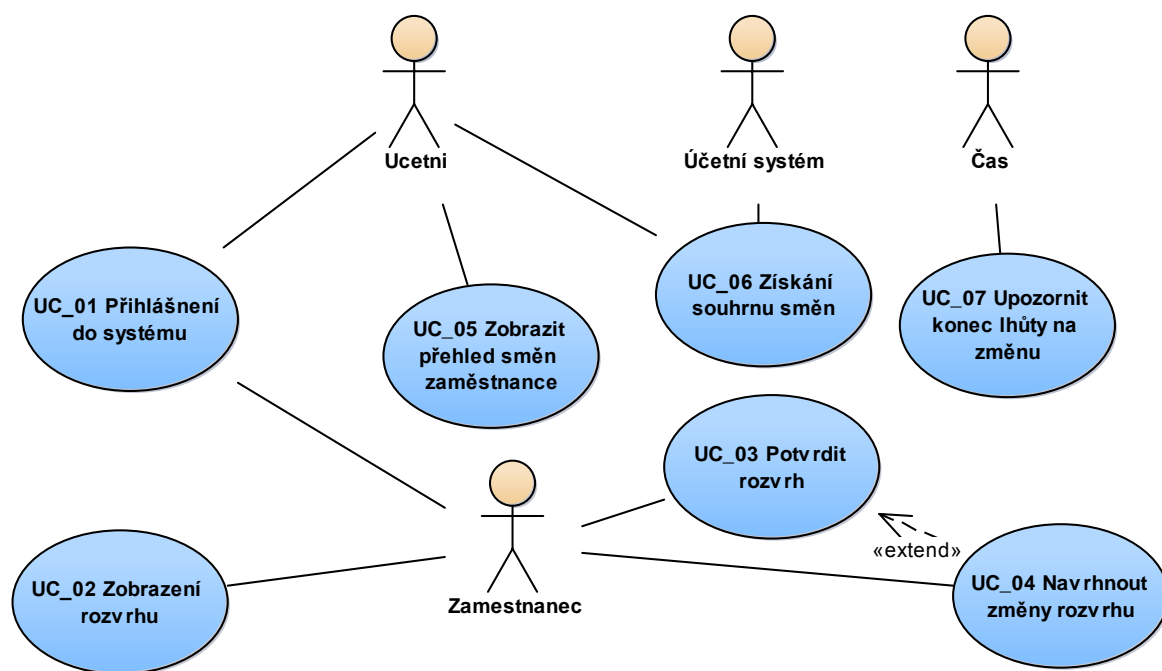
Již vygenerovaný rozvrh je možné upravit v podstatě dvěma způsoby. Za prvé je to automatické přeplánování rozvrhu, což je vhodné zejména před začátkem období, pro které je daný rozvrh vygenerován. Za druhé je to ruční úprava rozvrhu. Tímto druhem úpravy můžeme provést jemné doladění rozvrhu. Tato úprava je součástí této typové úlohy. Uživatel s rolí vedoucí využije tuto typovou úlohu zejména v případě, že bude nutné změnit například jednu směnu v již probíhajícím období. Při této změně je většinou nežádoucí, aby se dotkla ostatních zaměstnanců, a již odpracovaných služeb.

Zjednodušený scénář:

1. Aktér inicializuje ruční úpravu odpracovaných směn
2. Systém zobrazí formulář se směnami
3. Uživatel vybere požadovanou směnu a provede úpravy
4. Systém uloží údaje a zobrazí výzvu k výběru další akce
5. Uživatel zvolí uložení dat
6. Systém uloží data a zavře formulář

5.2.10. Zaměstnanec a ostatní uživatelé

Tato část diagramu popisuje ostatní uživatelské role systému a jejich aktivity. Pro nás je nejdůležitější role zaměstnanec, která reprezentuje zaměstnance, který má do systému přístup a pro kterého je sestavován daný rozvrh. V rámci svých práv bude moci navrhopvat případné změny nově vygenerovaného, ale nepotvrzeného rozvrhu. Další uživatelská role je účetní. Jedná se o pracovníka účetního oddělení, který zpracovává mzdy, a jeho pohyb sys-



Obrázek 12 - UC ostatní uživatelé

témem je omezený pouze pro zobrazení souhrnu směn. Uživatelské role Účetní systém a čas nejsou fyzičtí uživatelé a reprezentují účetní systém a CRON.

5.2.11. UC_01 Přihlášení do systému

Jak bylo již popsáno výše, každý uživatel, který bude chtít v systému cokoli udělat, či zobrazit, musí být přihlášený a autorizovaný. Tato typová úloha toto řeší. V rámci této typové úlohy dojde k autentifikaci a autorizaci daného uživatele.

Zjednodušený scénář:

1. Aktér inicializuje přihlášení do systému
2. Systém zobrazí přihlašovací formulář
3. Uživatel zadá přihlašovací údaje
4. Systém ověří přihlašovací údaje
5. Uživatel zadá nesprávné údaje
6. Systém zobrazí hlášku a neplatném přihlášení a vrací se k bodu 1
7. Systém zobrazí hlavní okno aplikace

5.2.12. UC_02 Zobrazení rozvrhu

V rámci této typové úlohy si již přihlášený uživatel zobrazí svůj aktuální rozvrh. V rámci zohlednění dnešních trendů bude tato část systému dostupná přes webový prohlížeč, a zaměstnanec bude mít možnost zkontrolovat si směny i z domu či z jiného místa.

Zjednodušený scénář:

1. Aktér inicializuje zobrazení svého rozvrhu služeb
2. Systém zobrazí formulář s rozvrhem služeb
3. Aktér inicializuje vygenerování sestavy
4. Systém vygeneruje a zobrazí sestavu
5. Uživatel uzavře formulář

5.2.13. UC_03 Potvrdit rozvrh

Cílem splnění této typové úlohy je potvrdit vygenerovaný rozvrh pro dané období. Jde o to, že vedoucí pracovník vygeneruje rozvrh, který nemusí z různých důvodů zaměstnanci vyhovovat. Například kvůli plánované lékařské prohlídce, dovolené apod. V tomto případě přecházíme k typové úloze **UC_04 Navrhnout změny rozvrhu**. V rámci této úlohy provedeme návrh změn svého rozvrhu, s kterými ovšem vedoucí pracovník nemusí souhlasit

a může je ignorovat. Pokud ovšem vše souhlasí, provede zaměstnanec potvrzení rozvrhu, a tím dá v podstatě najevo, že nemá s takto rozvrženými směnami problém.

Zjednodušený scénář:

1. Uživatel inicializuje potvrzení rozvrhu
2. Systém zobrazí formulář s příslušným rozvrhem
3. Uživatel potvrdí rozvrh směn
4. Uživatel inicializuje úpravu rozvrhu
5. Systém vloží typovou úlohu UC_04
6. Uživatel potvrdí rozvrh směn
7. Systém uloží údaje a zavře formulář

5.2.14. UC_04 Navrhnout změny rozvrhu

Jak bylo popsáno v předchozím odstavci, zaměstnanec který do jisté míry nesouhlasí se svým rozvrhem, má možnost navrhnout dodatečné úpravy. Důvodu pro toto může být několik. Nemoc, vyšetření, výměna směn s jinou sestrou apod. Na tuto úpravu má samozřejmě omezen čas. Po uplynutí lhůty pro tyto úpravy se rozvrh směn automaticky potvrdí.

Zjednodušený scénář:

1. Aktér inicializuje změnu rozvrhu směn
2. Systém zobrazí formulář s příslušným rozvrhem
3. Uživatel provede změny údajů
4. Systém zobrazí výzvu k uložení záznamů
5. Uživatel potvrdí změny
6. Systém provede úpravu rozvrhu
7. Systém zavře formulář

5.2.15. UC_05 Zobrazit přehled směn zaměstnance

K této typové úloze přistupuje uživatel s rolí účetní. Jde o pracovníka mzdového nebo účetního oddělení, který na základě odpracovaných směn vypočítává mzdy. Jelikož je systém propojen s existujícím účetním systémem, jedná se spíše o doplňkovou a nouzovou variantu získávání těchto dat. Cílem této typové úlohy je zobrazit uživateli souhrn směn daného zaměstnance v daném období.

Zjednodušený scénář:

1. Uživatel inicializuje zobrazení směn zaměstnance
2. Systém zobrazí formulář pro výběr zaměstnance
3. Aktér vybere daného zaměstnance
4. Systém zobrazí formulář s přehledem směn vybraného zaměstnance
5. Aktér inicializuje uzavření formuláře
6. Systém zavře formulář

5.2.16. UC_06 Získání souhrnu směn

Do jisté míry se jedná o podobnou typovou úlohu jako UC_05, avšak v tomto případě se získává výpis směn všech zaměstnanců k danému časovému období. Primárně bude tuto úlohu využívat stávající účetní systém, který je schopen zpracovat externí data ve formátu XML. Samozřejmostí je dodržení předem dané struktury dokumentu, která je popsána v dokumentaci a je tedy možné dohledat ji. Při nějaké chybě v průběhu automatického stahování dat je možné stáhnout data ručně jako uživatel účetní, a do stávajícího systému je nahrát pomocí ručního importu dat, také ve formátu XML.

Zjednodušený scénář:

1. Aktér inicializuje získání souhrnu směn
2. Systém provede kontrolu uživatele
3. Systém zobrazí formulář se souhrnem směn
4. Uživatel vybere druh exportu
5. Systém odešle data v požadovaném formátu
6. Systém uzavře formulář
7. Systém vygeneruje XML a odešle ho

5.2.17. UC_07 Upozornit konec lhůty na změnu

Tato typová úloha nám zajišťuje pouze to, že po uplynutí předem daného časového intervalu dojde k odeslání upozornění zaměstnancům, kteří ještě neprovedli požadované dodatečné úpravy svého rozvrhu směn. Jak bylo popsáno dříve, po uplynutí této doby je rozvrh automaticky potvrzen a žádné další úpravy již nejsou možné. Uživatelská role bude v tomto případě využívána například kontrolním časovým mechanismem CRON.

5.2.18. Závěr

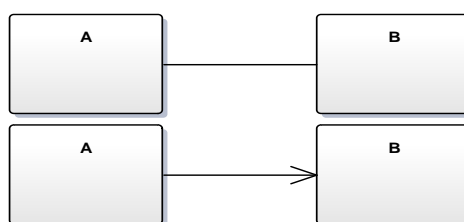
Díky vizualizaci činností uživatelů nového systému pro tvorbu rozvrhů směn, jsme dosáhli přehledného zobrazení jejich aktivit. Tento diagram skvěle posloužil při schůzkách s uživateli nového systému, jelikož i člověk bez předchozí znalosti UML je schopen diagram pochopit a vše jasně definovat. Díky tomu bude možné navrhnout i implementovat systém v takové kvalitě, která bude odpovídat představám budoucích uživatelů. Scénáře uvedené u typových úloh tvoří podklad pro tvorbu aplikační logiky systému.

5.3. Diagramy tříd

V předchozí kapitole jsme vytvořili diagramy typových úloh, které nám reprezentují uživatelské role v systému a aktivity, které tyto uživatelské role využívají. Diagramy tříd jsou základním blokem při návrhu objektově orientovaných aplikací. Jak je již z názvu patrné, základním blokem diagramů bude třída. Třídy jsou mezi sebou spojeny několika druhy vazeb, podobně jako u typových úloh. Jedná se zejména o vazby:

Asociace

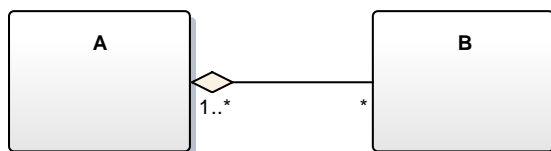
Jedná se o základní typ mezi dvěma entitami. Každá entita může existovat nezávisle na té druhé. Výchozím směrem asociace je obousměrná asociace, to znamená, že jedna entita má referenci na druhou a obráceně. Toto můžeme řídit pomocí přidání jednoduché šipky, což je znázorněno na obrázku číslo 13.



Obrázek 13 - asociace

Agregace

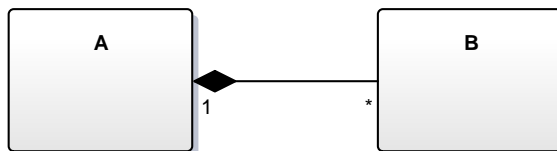
Agregace popisuje vztah celek – část. Jedná se o silnější vzat asociace. Vazba je znázorněna plnou čarou, která je zakončená prázdným kosočtvercem. Entita, u které je kosočtverec reprezentuje celek, druhá entita je reprezentace části. Entita, která reprezentuje část, může existovat i sama o sobě, nebo může být součástí i jiných celků.



Obrázek 14 - agregace

Kompozice

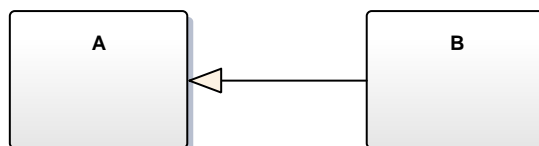
Kompozice je nejsilnější vazba mezi entitami. V případě kompozice nemá entita reprezentující část sama o sobě žádný smysl. Vazba je reprezentována plnou čarou zakončenou jako v předchozím případě kosočtvercem, avšak plným.



Obrázek 15 - kompozice

Generalizace

V tomto případě se jedná o již výše popsaný vztah dědičnosti. Jedna entita dědí chování entity druhé. K tomuto může obsahovat ještě své chování. Vazba je reprezentována plnou čarou, která je zakončená uzavřenou šipkou (trojúhelníkem).

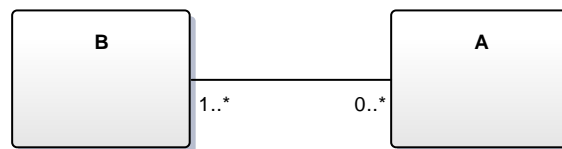


Obrázek 16 - generalizace

Multiplicita

Multiplicita není vazba jako taková, ale je její vlastnost. Multiplicita neboli násobnost, určuje, kolik entit A je vázáno s entitou B. Tuto vlastnost uvádíme u vazeb na obou stranách. Rozdělujeme několik druhů multiplicity:

- 1 – multiplicita je v tomto případě konkrétně 1
- 0..1 – multiplicita je buď 0, nebo 1
- 0..* - multiplicita je v intervalu 0 a ∞
- 1..* - multiplicita podobná jako o bod výše, s rozdílem že musí být alespoň 1
- * - v podstatě se jedná o stejný druh multiplicity jako o bod výše 0 až ∞



Obrázek 17 - multiplicita

5.3.1. Doplnění modelu

Nyní jsme si popsali základní stavební prvky modelu tříd. Pokud bychom nechali model v tomto stavu, jednalo by se opravdu pouze o základní návrh tříd. Pro kvalitnější model je nutné doplnit diagramy o další důležité informace. Jedná se zejména o atributy jednotlivých tříd. Další informace které je třeba doplnit, jsou metody, se kterými budou jednotlivé třídy pracovat. Metody by měly z velké části vycházet z analýzy scénářů Use Case diagramu. Tímto postupem dosáhneme toho, že všechny požadované funkce systému budou v navrhovaném modelu obsaženy.

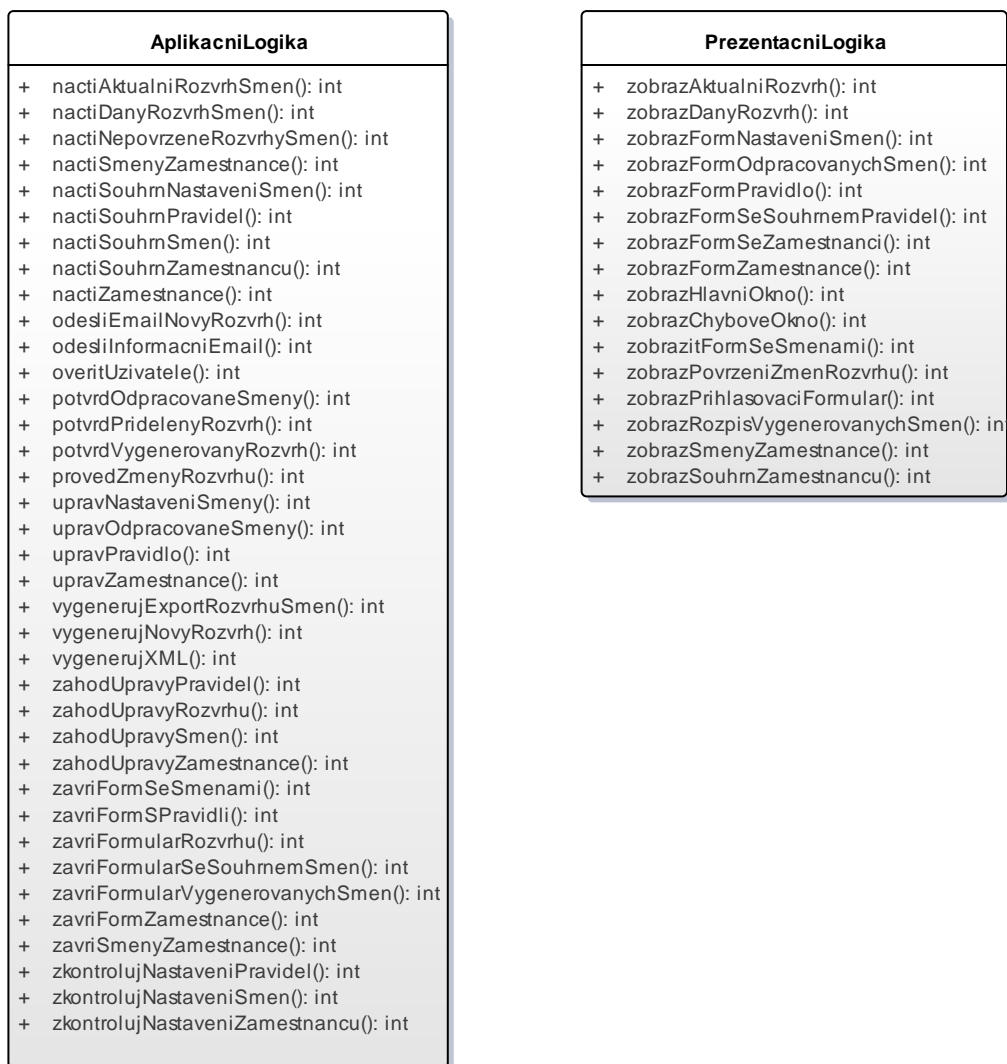
Postup analýzy scénářů a tvorba modelu tříd

1. Postupně projdeme všechny scénáře typových úloh a zformulujeme názvy metod
2. Rozdělíme metody na prezentační a aplikační logiku systému
3. Metody aplikační logiky seskupíme do skupin
4. Vytvoříme analytický model tříd
5. Definujeme atributy tříd
6. Definujeme vazby mezi třídami

Při dodržení těchto kroků a za předpokladu kvalitně zpracovaných scénářů typových úloh, vytvoříme analytický model tříd, který nám slouží jako základní identifikace jednotlivých tříd systému a jejich provázanost. Dále nám tento model může posloužit jako podklad pro vytváření databázového schématu. Třídy nám v tomto případě reprezentují jednotlivé tabulky, jejich proměnné reprezentují atributy a vazby mezi třídami můžeme použít jako podklad pro vytvoření relací, včetně jejich kardinality.

5.3.2. Prezentační a aplikační logika

Jak bylo popsáno výše, ze scénářů zformulujeme jednotlivé názvy metod a určíme, zda se jedná o prezentační nebo aplikační logiku. Ve zkratce, aplikační logika je to co nový systém provádí, a prezentační co zobrazuje uživateli například na monitoru. Na obrázku číslo 18 jsou všechny metody zformulované a pojmenované. V této fázi neřešíme jak budou metody pracovat a jaký bude jejich algoritmus, a neřešíme ani návratové hodnoty metod nebo vstupní parametry. Proto je u každé metody návratový typ `int` a žádný parametr. Tatko pojmenované metody je nyní nutné vhodně seskupit a umístit do jednotlivých tříd. Tyto třídy musí být vhodně pojmenované, zejména z důvodu jednoznačné identifikovatelnosti a přehlednosti. Tedy například metody *odesliEmailNovyRozvrh()* a *odesliInfomacniEmail()* budou zapouzdřeny v jedné třídě.



Obrázek 18 prezentační a aplikační logika

5.3.3. Analytický model tříd

Analytický model tvoří základní strukturální diagram systému, a jeho smyslem je popsat stavební prvky systému a jejich vztahy

Postupujeme tedy dále, tak jak již bylo popsáno a seskupíme všechny metody aplikační logiky do jednotlivých tříd. Pamatujeme na to, že názvy tříd by měly korespondovat s jejich funkcí v navrhovaném systému. Název každé třídy začíná velkým písmenem a je reprezentován podstatným jménem. Slovesa a ostatní slovní druhy nejsou pro názvy tříd vhodné a proto se jím co nejvíce vyvarujeme.

Zamestnanec, Ucetni, Vedouci

Třída reprezentuje zaměstnance, pro kterého je daný rozvrh vytvářen. Jak je znázorněno v diagramu na obrázku číslo 14 je zaměstnanec potomkem třídy **Uzivatel**. Rozdělení uživatelů do těchto třech tříd je zejména z hlediska přehlednosti modelu.

- Generalizace – dědí ze třídy **Uzivatel**
- Vazby – asociace N x M s třídou **Rozvrh**

Uzivatel

V našem modelu se jedná o předka všech dalších uživatelů systému. Jedná se o abstraktní třídu, která sama o sobě nemůže mít svojí vlastní instanci. Všechny třídy reprezentující uživatele systému dědí atributy a metody této třídy.

Rozvrh

Třída **Rozvrh** reprezentuje jeden daný rozvrh pro předem dané období, a zapouzdřuje pod sebe všechny metody, které mají co dělat s rovrhem

- Vazby – asociace N x M s třídami **Email**, **Zamestnanec**, **Smena**

Smena

Třída reprezentuje každou jednotlivou směnu rozvrhu

- Vazby – asociace N x M se třídou **Rozvrh**, **Pravidlo**

Pravidlo

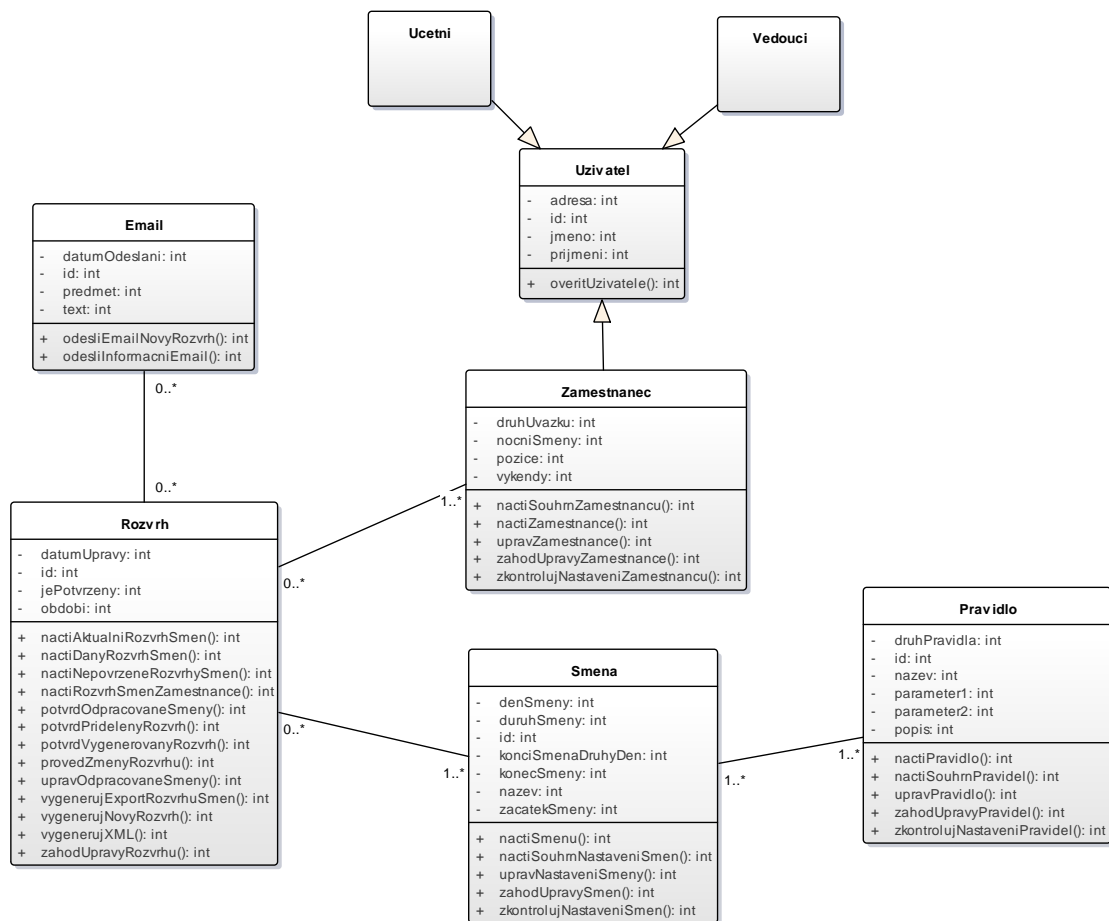
Třída reprezentuje pravidlo, nebo pravidla, která je nutná dodržet při sestavování rozvrhu. Tato pravidla byla popsána již v předchozích kapitolách.

- Vazby – asociace N x M se třídou **Smena**

Email

Třída **Email** obsluhuje všechny operace, při kterých se nějakým způsobem pracuje s E-mailovými zprávami.

- Vazby – asociace N x M se třídou **Rozvrh**



Obrázek 19 - analytický model tříd

No obrázku číslo 19 se znázorněn kompletní analytický model tříd navrhovaného rozvrhovacího systému. Jak lze vidět, do třídy byly doplněny požadované atributy, třídy byly provázány asociacemi s vyznačením multiplicit a generalizační vazby. K tomuto diagramu se vrátíme ve fázi návrhu databázového modelu.

5.3.4. Návrhový model tříd

Dalším bodem při vytváření modelu tříd je tzv. návrhový model tříd. Jedná se v podstatě o upřesněný analytický model. V návrhovém modelu jsou třídy, vztahy a metody upřesněny do takové úrovně, že je již možné je implementovat. Jedná se zejména o:

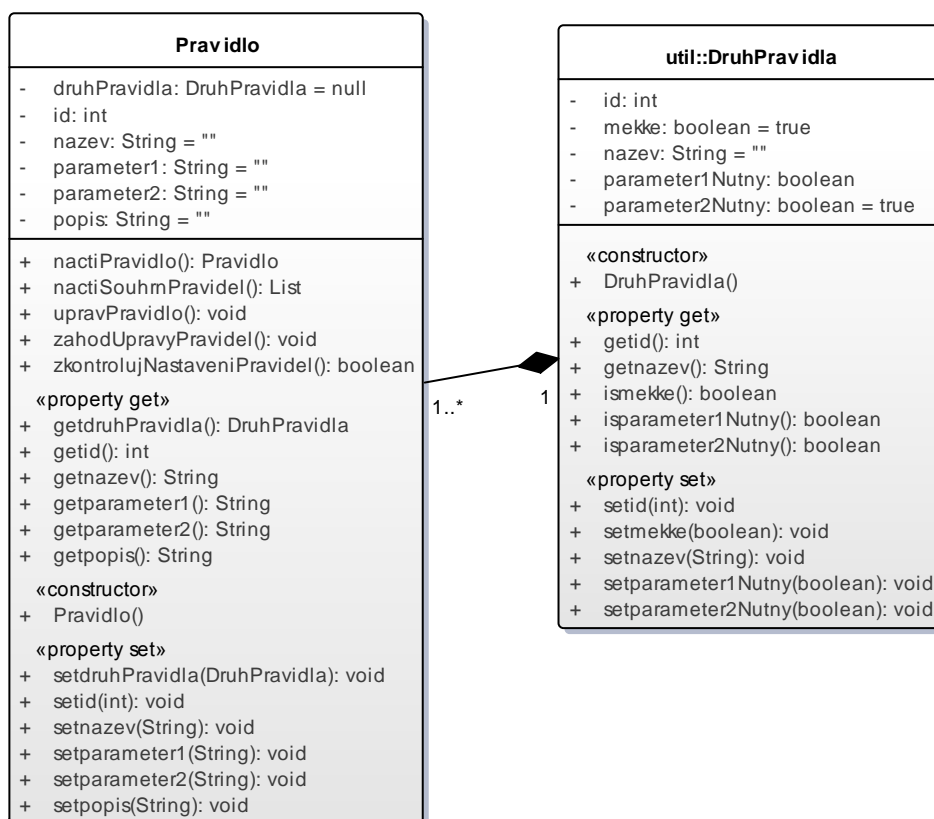
- doplnění datových typů atributů (int, String, Date, ...)
- určení návratových hodnot metod (void, int, boolean, ...)
- doplnění parametrů metod
- doplnění přístupových metod (Gettery a settery)
- doplnění výchozích hodnot atributů event. parametrů

- doplnění konstruktorů
- určení viditelnosti atributů a metod (public, private, protected)
- zavedení pomocných tříd u atributů s vnitřní strukturou (např. Adresa)
- využití tříd v knihovnách implementačního prostředí
- zpřesnění relací (kompozice, agregace, ...)

Tímto upřesněním vzniknou kompletní a detailně rozpracované třídy, které mají úplný výčet atributů a metod, mají konkrétní vztahy s jasným způsobem realizace, například pomocí kolekce, mapy, nebo dalším prostředků. Takto detailní model je poté možné pomocí softwaru přímo vygenerovat do daného jazyka.

Upřesnění modelu

Z důvodu rozsáhlosti modelu uvedeme pouze upravení jedné konkrétní třídy našeho modelu. Celý diagram je přiložen v příloze. Popíšeme si třídu Pravidlo, viz obrázek č. 20.



Obrázek 20 - pravidlo

Můžeme si všimnout, že u atributu byla doplněna výchozí hodnota. Například pro String je touto hodnotou prázdný řetězec. Místo tohoto prázdného řetězce může být jako výchozí hodnota zvolen jakýkoliv jiný řetězec, pokud by to bylo vhodné.

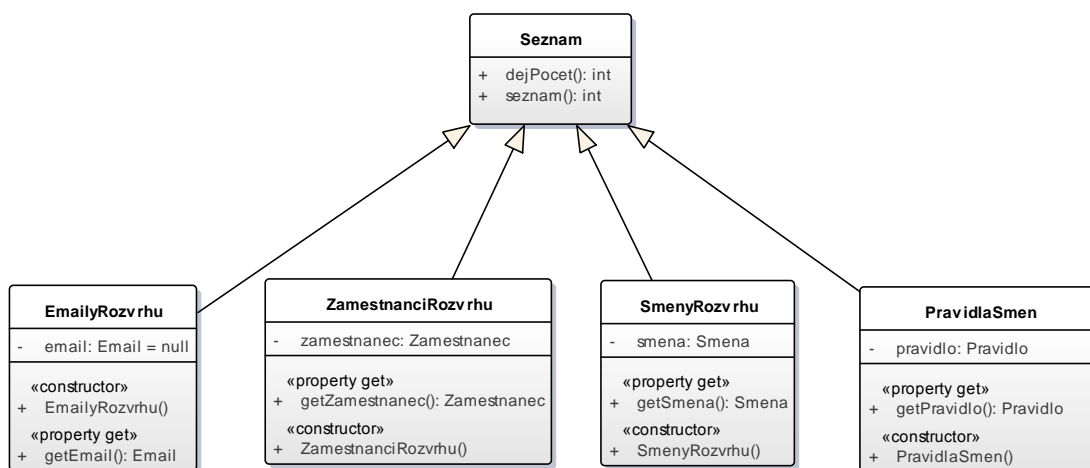
Pro všechny parametry, které jsou privátní, byly doplněny přístupové hodnoty, takzvané gettery a settery, které nám slouží buďto k nastavení hodnoty dané proměnné, nebo k jejímu přečtení například z jiné třídy, v našem modelu například ze třídy Smena.

Můžete si všimnout třídy DruhPravidla, která patří do balíčku util. Jedná se o takzvanou pomocnou třídu. Jedná se o třídu pro atribut s vnitřní strukturou druhPravidla. Do balíčku util patří dále třídy:

- Adresa – adresa uživatele
- DruhSmeny – druh směny (denní, noční, ranní, ...)
- DruhUvazku – druh úvazku (HPP, 50%, ...)
- Obdobi – období pro rozvrh (od do)

Kolekce

Pro znázornění vazeb N x M se v návrhovém diagramu tříd používají nejčastěji kolekce hodnot. Tyto kolekce slouží jako zpřesnění těchto relací. Seznam všech kolekcí použitých v modelu je znázorněn na obrázku číslo 21.



Obrázek 21 - seznam kolekcí

5.3.5. Závěr

V předchozí části práce jsme vytvořili několik základních diagramů a modelů. V první fázi to byl diagram typových úloh. Díky detailně zpracovaným scénářům jsme byli schopni vymodelovat analytické a návrhové diagramy tříd, které následně pomohou vývojářům pro programování rozvrhovacího systému.

5.4. Datový model

Datový model je nedílnou součástí návrhu téměř jakékoliv aplikace. Správným návrhem aplikace datového modelu lze do značné míry ovlivnit bezproblémový chod aplikace, její udržovatelnost a případnou rozšiřitelnost. Prvním krokem vytváření datového modelu je zpravidla konceptuální datový model. V naší terminologii se jedná o ER diagram, tedy Entitně relační diagram. Jedná se o zobecněný model datové struktury, tudíž konkrétní implementace datové struktury nemusíme uvažovat. [16]

5.4.1. Databázové schéma

Databázové schéma bude sloužit jako podklad pro správce databáze. Ten podle tohoto schématu bude schopen vytvořit databázi se strukturou potřebnou pro bezproblémový a sto-percentní chod aplikace. Je samozřejmostí, že tohoto schématu se musí držet i vývojář, nebo tým vývojářů, který bude pracovat na vývoji a implementaci systému.

Souhrn

Díky návrhu při vytváření modelu bylo identifikováno celkem 14 různě provázaných tabulek. Pouze pro doplnění uvedu výpis jednotlivých tabulek.

Tabulka 7 - databázové tabulky

Název	Popis
uzivatel	data každého uživatele s přístupem do systému
druh_uvazku	druh úvazku zaměstnance - částečný, HPP, atd...
adresa	číselník adres
email	tabulka odeslaných emailů
rozvrh	směny přiřazené zaměstnancům v daném období
odbobi	období od - do
smena	jednotlivá směna
pravidlo	pravidlo směny
druh_pravidla	druh pravidla směna - měkké, tvrdé
druh_smeny	druh směny zaměstnanců - noční, denní, atd.
smeny_rozvrhu	přiřazení směny k rozvrhům
pravidla_smen	pravidla pro různé směny
zamestnanci_rozvrhu	vazební tabulka rozvrhů jednotlivých zaměstnanců
emaily_rozvrhu	vazební tabulka pro emaily k rozvrhům

Datový typy jednotlivých tabulek databázového schématu jsou přizpůsobeny prostředí Windows SQL serveru. V našem případě podle zadání se bude jednat o edici MS SQL server 2014, běžící na platformě Windows Server 2012. Popis bližší konfigurace není předmětem tohoto návrhu, a nebude tedy řešena.

5.4.2. Závěr

Nyní máme k dispozici kvalitní podkladový materiál, který můžeme poskytnout vývojovému týmu, nebo databázovému administrátorovi. V případě, že by se zadání pro použité prostředí změnilo po dokončení návrhu, Enterprise Architect poskytuje funkce pro rychlé přetransformování databázového schématu pro dané prostředí, například pro Oracle, Sybase, MySQL nebo pro jiné další.

5.5. Návrh architektury aplikace

Nyní máme hotové podklady pro tvorbu navrhované aplikace pro tvorbu rozvrhů směn. Nyní je vhodné navrhnout základní architekturu. Jedná se zejména o architekturu síťovou, a serverovou. Architektonické řešení je rozděleno v podstatě na dvě části:

1. Interní – přístup k aplikaci z intranetu nemocničního zařízení
2. Externí – přístup k aplikaci přes webový prohlížeč

5.5.1. Interní

Interní část je sama o sobě rozdělena na několik částí. V kostce lze říci, že je to část serverová, část přístupu pro obsluhu rozvrhů a část pro přístup účetního systému, nebo pracovníka účetního oddělení.

Zařízení DEV001 zde reprezentuje hlavní server, jehož konfigurace není zatím důležitá. Na serveru bude nainstalován operační systém Windows server 2012. Jako databázové prostředí bude nainstalován Windows server 2012 a pro přístup přes webový prohlížeč bude použit webový server IIS 5.0

Zařízení DEV002 reprezentuje počítač, který bude mít k dispozici vedoucí pracovník, v našem případě vrchní sestra, která bude sestavovat rozvrhy směn pro své podřízené zaměstnance v rámci oddělení. Toto zařízení nebude s největší pravděpodobností pouze jedno, ale architektura bude totožná.

Zařízení DEV003 je pracovní stanice účetního oddělení. Pracovník na tomto zařízení má v podstatě dvě možnosti, jak přistupovat do aplikace pro sestavování rozvrhů směn. Za prvé je to přístup přes webový prohlížeč, viz popis typových úloh. A za druhé je to přístup přes již zavedený účetní systém, který bude z našeho systému získávat data o odpracovaných směnách, za účelem výpočtu mezd zaměstnanců.

5.5.2. Externí

Poslední zařízení DEV003-N není jedno jediné zařízení, ale jedná se o jakýkoliv počítač, pomocí něhož budou zaměstnanci přistupovat ke svým rozvrhům například z domácího připojení. Z tohoto důvodu zde není rozlišen ani operační systém, jelikož uživatelé používají OS různé.

Počítač účetního oddělení bude komunikovat se serverovým počítačem za pomoci webového prohlížeče pokud, bude do systému přistupovat pracovník oddělení, nebo pomocí webových služeb, která bude poskytovat potřebná data, bude-li se jednat o účetní systém. Na obrázku znázorněno jako „WebPages“ a „XML – účetní souhrny“.

5.5.4. Závěr

Navržený návrh byl vypracován na základě nynějších požadavků na strukturu rozvrhovacího systému. V případě, že by během vývoje systému došlo k výrazným změnám, je vhodné tyto návrhy upravit například tak, aby lépe splňovaly bezpečnostní požadavky.

5.6. Návrh uživatelského rozhraní

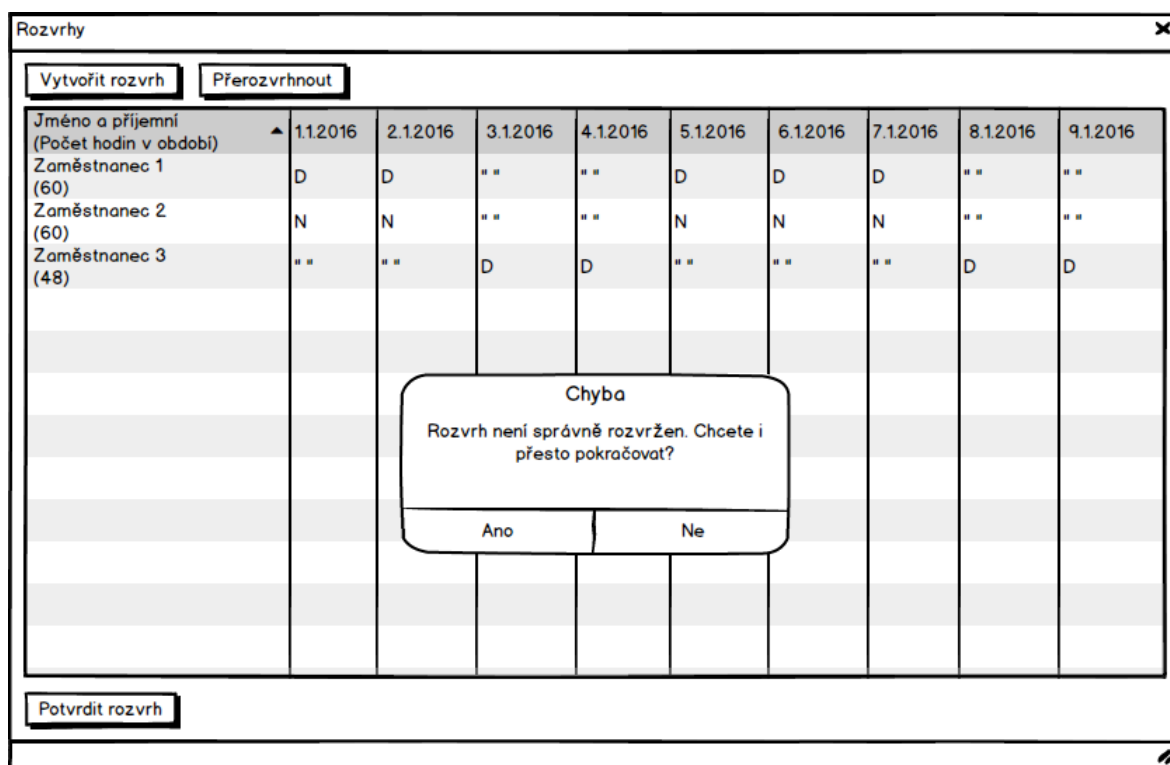
Uživatelské rozhraní je nedílnou součástí každé aplikace. Kdokoliv, kdo bude chtít obsluhovat jakýkoliv systém, bude nějaké rozhraní potřebovat. Vhodnost návrhu grafického rozhraní do značné míry ovlivňuje globální hodnocení nového rozvrhovacího systému. Uživatel systému neocení bezchybně fungující aplikační logiku, pokud pro něj nebude ovládání aplikace dostatečně příjemné. Ovládání aplikace musí být pokud možno co nejjednodušší a intuitivní. Není vhodné volit rozhraní příliš složité a nepřehledné, i když je pravda, že v některých případech není možné rozhraní příliš zjednodušit.

V případě návrhu uživatelského rozhraní našeho rozvrhovacího systému bude vše rozděleno na několik oddělených částí. Za prvé půjde o rozhraní samotné aplikace, kterou bude obsluhovat vedoucí pracovník, a za druhé půjde o webové rozhraní, skrze které budou do systému přistupovat zaměstnanci. V druhém případě půjde o opravdu jednoduché rozhraní, které nebude poskytovat skoro žádné funkce. Bude sloužit zejména pro zobrazení rozvrhů či export do požadovaných formátů, jako je XLS či PDF.

Jako nástroj pro vytvoření jednoduchého uživatelského rozhraní jsem využil software Balsamiq Mockups. Enterprise architect tuto funkcionalitu poskytuje také, ale již v minulosti se mi vytváření uživatelského rozhraní v něm neosvědčil jako jednoduchý nástroj. Rozhraní není navrhované přímo pro Windows, ale poskytuje pouze podklad budoucímu vývojáři. Na něm bude rozhodnout se, jaké prostředí pro vývoj využít. V následujících odstavcích budou prezentovány pouze podstatné části rozhraní.

5.6.1. Výpis rozvrhu

Toto je pravděpodobně nejdůležitější část celého uživatelského rozhraní. V této části jsou zobrazeny vygenerované rozvrhy směn zaměstnanců. Ve vygenerovaných směnách by mělo být možné provádět jemné ruční úpravy, například při vyměňování směn mezi sestrami a jiné úpravy. Těchto úprav by mělo být co nejméně z důvodu popsaných výše.



Obrázek 24 - okno rozvrhů s chybovou hláškou

Při kliknutí pravým tlačítkem myši na příslušnou směnu se zobrazí dialogové okno, ve kterém uživatel vybere, jaký druh změny chce se směnou provést (volno, denní, noční). Při potvrzení rozvrhu dojde ke kontrole správnosti rozvrhu. O výsledku bude uživatel informován buďto informační, nebo chybovou hláškou, viz obrázek číslo 24.

5.6.2. Evidence zaměstnanců

V sekci evidence zaměstnanců bude pracováno se záznamy zaměstnanců, pro které je daný rozvrhový systém navrhován. Dostupné budou informace o zaměstnanci, o jeho bydlišti, statistika jeho odpracovaných směn, druhy úvazku a další podstatné informace. První formulář při otevření evidence zaměstnanců bude výpis všech zaměstnanců v tabulce či datagridu, ve kterém bude možné filtrovat a najít tak požadovaného zaměstnance. Po zvolení daného zaměstnance, či při zakládání nového zaměstnance, se zobrazí detailní formulář s výpisem dat vybraného pracovníka. Toto je na obrázku číslo 25.

Zaměstnanec - "Jméno zaměstnance"

Jméno Osobní číslo Ulice ČP
Příjmení Město PSČ

Noční Víkendy

Druh úvazku

Pozice

Směny za kalendářní rok

Měsíc	Počet směn	Z toho nočních	Z toho víkendů	Čerpáno dovolené (zbývá)	Celkem odpracovaných hodin
Leden	15	6	4	0 22	180
Únor	18	8	6	0 22	216 !

Obrázek 25 - karta zaměstnance

5.6.3. Nastavení směn

V této sekci bude možné konfigurovat jednotlivé směny, které se budou poté přiřazovat do daného rozvrhu. V sekci směn budou vypsány směny po jednotlivých dnech. Ke každé takovéto směně budou vypsána požadovaná pravidla, která jsou pro generování přesných rozvrhů důležitá, která již musí být definována. Mezi tato pravidla, která rozvrh ovlivňují, může patřit např. počet pracovníků směny, minimální rozestup mezi směny, maximální počet odpracovaných hodin a další.

Směny

Generovat další směny Upravit vybranou směnu Upravit pravidla směň

Směna	Vikend?	Počet pracovníků	Denní	Noční Končí další den?
1.2.2016		3	6:00-18:00	18:00-6:00 Ano
2.2.2016		3	6:00-18:00	18:00-6:00 Ano
3.2.2016		3	6:00-18:00	18:00-6:00 Ano
4.2.2016		3	6:00-18:00	18:00-6:00 Ano
5.2.2016		3	6:00-18:00	18:00-6:00 Ano
6.2.2016	ANO	2	6:00-18:00	18:00-6:00 Ano
7.2.2016	ANO	2	6:00-18:00	18:00-6:00 Ano
8.2.2016		3	6:00-18:00	18:00-6:00 Ano
9.2.2016		3	6:00-18:00	18:00-6:00 Ano
10.2.2016		3	6:00-18:00	18:00-6:00 Ano

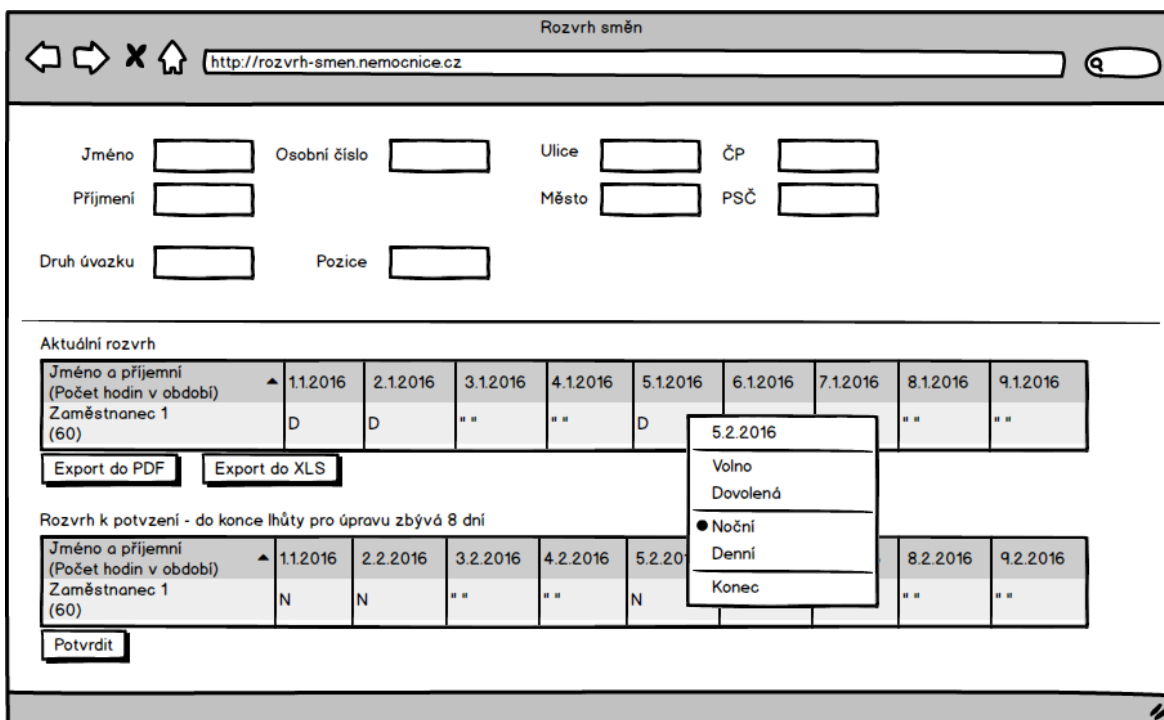
Uložit

Obrázek 26 - nastavení směň

Další části uživatelského rozhraní rozvrhovacího systému tvoří v podstatě formuláře pro konfiguraci triviálních údajů a není proto důležité je v práci uvádět. Důležitá je ale druhá část systému, a to webové rozhraní, které bude popsáno dále.

5.6.4. Webové rozhraní

Webové rozhraní aplikace tvoří druhou část uživatelského rozhraní. Jedná se o rozhraní pro přístup zaměstnanců. Ti budou mít možnost skrze toto rozhraní přistupovat ke svým rozvrhům. Pokud bude uživatel potřebovat upravit nepotvrzený rozvrh, aplikace by mu toto měla umožnit možná co nejjednodušeji. Uživatel musí být nejprve ověřen. V případě úspěšného ověření se zobrazí aktuální rozvrh pro dané období, a nově vygenerovaný rozvrh, ve kterém bude možné provádět změny, pokud tedy bude dostupný. Žádné další funkce aplikace nebude potřebovat, jelikož jakékoliv jiné úpravy údajů musí být prováděny pouze osobou k tomu oprávněnou, což je v našem případě vrchní sestra. Ta bude veškeré tyto operace provádět před hlavní aplikací. Při kliknutí pravým tlačítkem vyskočí nabídka akcí pro návrh změn rozvrhu. Možné by měly být návrhy jako změna druhu směny, návrh na volno či dovolenou, jak je znázorněno na obrázku číslo 27.



Obrázek 27 - webové rozhraní

5.6.5. Závěr

V předchozích kapitolách byly znázorněny nejdůležitější části uživatelského rozhraní navrhované aplikace. Výsledný vzhled naprogramované aplikace se může do jisté míry lišit, zejména v závislosti na zvoleném vývojovém prostředí. Nicméně základní principy zde nastíněné, by měly zůstat beze změny.

5.7. Shrnutí návrhu systému

Nyní po dokončení návrhu systému máme v ruce dostatek kvalitních podkladů, podle kterým bude budoucí vývojový tým navrhnout rozvrhovací systém, který bude co nejvíce pokrývat požadavky a potřeby uživatelů s tímto systémem pracujících. Vymezili jsme základní uživatele systému i s jejich oprávněními, základní funkce navrhovaného systému, databázovou strukturu potřebnou pro bezproblémový chod aplikace, hardwarovou architekturu a v neposlední řadě uživatelské rozhraní, které bylo navrženo ve spolupráci s budoucími uživateli rozvrhovacího systému. Jen ti jsou totiž schopni určit, v jakém systému se jim bude pracovat co možná nejjednodušeji a nejefektivněji. Případné další úpravy v rámci vývoje systému nejsou nemožné, hlavně díky agilnímu pohledu na vývoj aplikací.

6. Závěr

Tato práce se kladla za svůj cíl vytvoření návrhu rozvrhovacího systému směn pro potřeby nemocničního zařízení. Pro splnění tohoto cíle bylo nutné provést několik základních kroků. Stěžejní a první krok bylo provedení analýzy potřeb uživatelů. Podkladem pro tyto analýzy byly osobní schůzky se zaměstnanci zařízení. Nejvíce požadavků na funkce systému vzešly od vedoucího pracovníka, tedy pracovníka, který bude se systémem pracovat nejvíce. Tento pracovník, v našem případě vrchní sestra, bude zodpovědný za sestavování rozvrhů směn pro své podřízené. Na základě těchto požadavků bylo vytvořeno hned několik částí návrhu. Bylo to zejména sestavení typových úloh a návrh uživatelského rozhraní.

Další skupina pracovníků, kteří nám poskytli potřebné informace, byli pracovníci účetního oddělení. Od těchto pracovníků jsme získali důležité informace ohledně požadavků na rozpisy směn. Šlo například o informace o maximálním počtu za sebou jdoucích směn, maximálním počtu směn v měsíci a další důležité informace.

Poslední, ale rozhodně ne nedůležitá skupina pracovníků byli pracovníci IT oddělení. Tito pracovníci nám definovali požadavky na architekturu systému a na zabezpečení celé aplikace tak, aby nebyla žádným způsobem napadnutelná a tak aby za žádných okolností neohrozila chod ostatních systému, které již v nemocnici fungují.

Na základě těchto požadavků, námětů a postřehů byl vytvořen návrh rozvrhovacího systému obsahující tyto všechny důležité části, ať se jedná o část typových úloh, návrhů databázové architektury, či návrhu uživatelského rozhraní. Návrh je vhodné použít jako plnohodnotný podklad pro budoucí vývojový tým, který bude mít za úkol aplikaci naprogramovat a přivést k životu, jelikož veškeré důležité informace jsou v práci zahrnuty, a nebude tedy nutné získávat je od uživatelů znovu.

Obě hlavní části práce, tedy analytická a návrhová, byly vypracovány, čímž byly hlavní cíle celé práce splněny. Práce poskytuje základní podklad pro budoucí vývoj nového rozvrhovacího systému. Rozhodnutí jaký postup zvolit dál je nyní na osobách odpovědných za tato rozhodnutí.

7. Seznam použité literatury

1. CT Labs. *CZ Labs*. [Online] [Citace: 15. 2 2016.] <http://www.ctlabs.cz/rozpis/prik3.html>.
2. Punnakitikashem, Prattana. Integrated nurse staffing and assignment under uncertainty. *Doctoral Dissertation*. Arlington : University of Texas at Arlington, 2007. 839770934. 9780549144519 054914451X.
3. Aproximační algoritmy. *wikipedie*. [Online] 25. 10 2014. [Citace: 16. 2 2016.] https://cs.wikipedia.org/wiki/Aproxima%C4%8Dn%C3%AD_algoritmy.
4. Genetický algoritmus. *Wikipedie*. [Online] 13. 1 2016. [Citace: 16. 2 2016.] https://cs.wikipedia.org/wiki/Genetick%C3%BD_algoritmus.
5. Heuristické algoritmy. *Wikipedie*. [Online] 9. 29 2015. [Citace: 16. 2 2016.] https://cs.wikipedia.org/wiki/Heuristick%C3%A9_algoritmy.
6. Pravděpodobnostní algoritmus. *Wikipedie*. [Online] 4. 1 2015. [Citace: 16. 2 2016.] https://cs.wikipedia.org/wiki/Pravd%C4%9Bpodobnostn%C3%AD_algoritmus.
7. Demel, Jiří. *Grafy a jejich aplikace*. místo neznámé : Academia, 2015. 978-80-260-7684-1.
8. Chiaramonte, Michael Vincent. Competitive nurse rostering and rerostering. *Doctoral Dissertation*. Tempe : Arizona State University Tempe, 2008. 978-0-549-39897-4.
9. Zákoník práce §90. *Zákony online*. [Online] [Citace: 17. 2 2016.] <http://zakony-online.cz/?s2&q2=90>.
10. Wiegers, Karl Eugene. *Software requirements: practical techniques for gathering and managing requirements throughout the product development cycle*. místo neznámé : Microsoft Press, 2003. 978-0735618794.
11. FlexiRoster. FlexiRoster. *Rosterig*. [Online] 1. 5 2016. [Citace: 7. 4 2016.] <http://www.flexiroster.com.au/>.
12. OptaPlanner. *OptaPlanner*. [Online] 1. 2 2016. [Citace: 7. 4 2016.] <http://www.optaplanner.org>.
13. ABC Roster. *ABC Roster*. [Online] ABC Roster, 1. 3 2016. [Citace: 7. 4 2016.] <http://www.abc-roster.com/>.
14. Král, Jaroslav. *Informační systémy*. Praha : Science, 1998. 80-860-8300-4.
15. Čápka, David. 2. díl - UML - Use Case Diagram. *ITNetwork*. [Online] 21. 9 2013. [Citace: 10. 3 2016.] <http://www.itnetwork.cz/navrhove-vzory/uml/uml-use-case-diagram>.

16. Sumathi, S a S ESAKKIRAJAN. *Fundamentals of Relational Database Management Systems*. London : Springer, 2007. 3540483977.

8. Seznam obrázků

Obrázek 1 - vzorový rozvrh směn [1]	2
Obrázek 2 - třídy složitosti [7]	7
Obrázek 3 - dědičnosti uživatelů.....	14
Obrázek 4 - možnosti konfigurace [11]	24
Obrázek 5 - rozpis směn [11].....	25
Obrázek 6 - OptaPlanner [12]	26
Obrázek 7 - ABC Roster	27
Obrázek 8 - Rozvrhy směn verze 1.16.....	28
Obrázek 9 - UC Aktér	30
Obrázek 10 - UC případ užití.....	30
Obrázek 11 - UC - Vedoucí	31
Obrázek 12 - UC ostatní uživatelé	35
Obrázek 13 - asociace	39
Obrázek 14 - agregace.....	40
Obrázek 15 - kompozice	40
Obrázek 16 - generalizace.....	40
Obrázek 17 - multiplicita	41
Obrázek 18 prezentační a aplikační logika	43
Obrázek 19 - analytický model tříd.....	45
Obrázek 20 - pravidlo.....	46

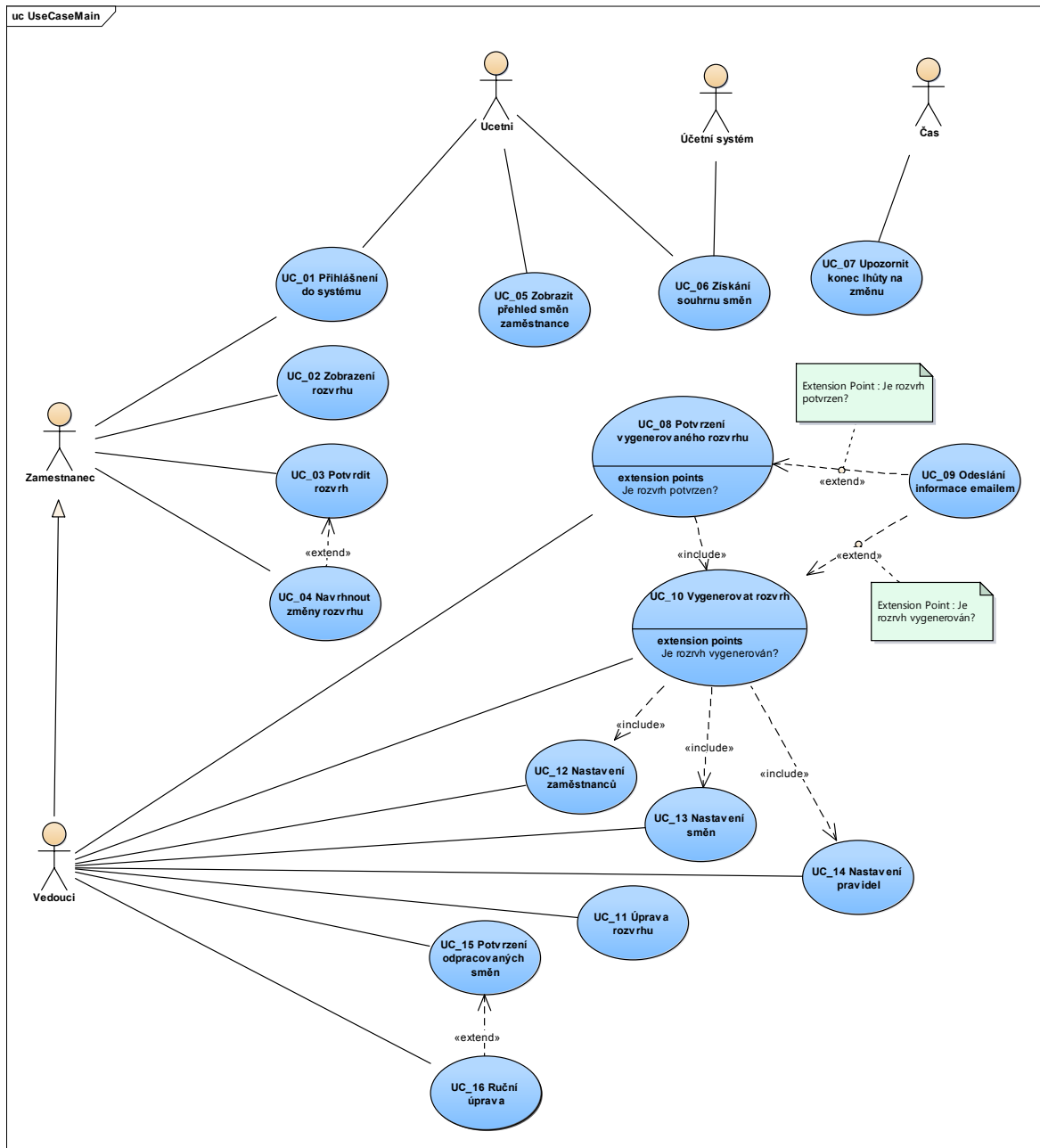
Obrázek 21 - seznam kolekcí	47
Obrázek 22 - databázové schéma	49
Obrázek 23 - architektura	52
Obrázek 24 - okno rozvrhů s chybovou hláškou.....	54
Obrázek 25 - karta zaměstnance	55
Obrázek 26 - nastavení směn	56
Obrázek 27 - webové rozhraní	57

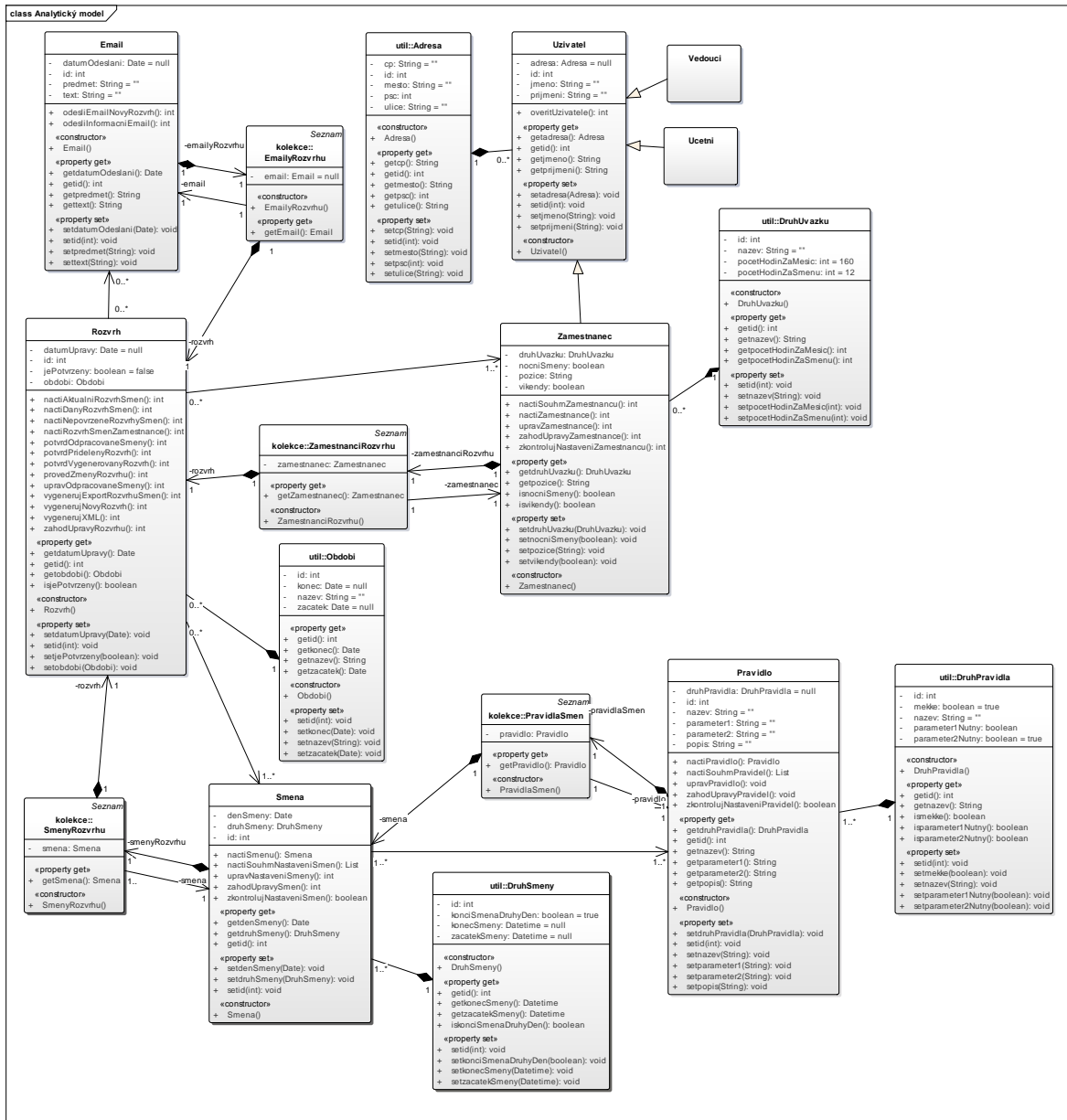
9. Seznam tabulek

Tabulka 1 - ruční řešení.....	4
Tabulka 2 - ruční řešení sofistikovanější	4
Tabulka 3 - poloautomatické řešení	5
Tabulka 4 - automatické řešení	5
Tabulka 5 - krátký dlouhý týden	12
Tabulka 6 - požadavky na vstupy a výstupy	16
Tabulka 7 - databázové tabulky	50

10. Přílohy

Příloha číslo 1.





Podklad pro zadání DIPLOMOVÉ práce studenta

PŘEDKLÁDÁ:	ADRESA	OSOBNÍ ČÍSLO
Fiala Jiří	Tlustice 70, Tlustice	I14284

TÉMA ČESKY:

Návrh rozvrhového systému směn pro personál nemocnice

TÉMA ANGLICKY:

Design of hospital staff timetabling system

VEDOUcí PRÁCE:

doc. RNDr. Jaroslava Mikulecká, CSc. - KIKM

ZÁSADY PRO VYPRACOVÁNÍ:

1. Přehled přístupů k rozvrhování směn pro zdravotnický personál
2. Dostupné systémy na podporu rozvrhování
3. Analýza potřeb konkrétního zdravotnického zařízení
4. Návrh systému
 - 4.1 Aplikační logika
 - 4.2 Uživatelské rozhraní
 - 4.3 Architektura systému
5. Závěr - silné a slabé stránky navrženého systému

SEZNAM DOPORUČENÉ LITERATURY:

Alvarez Oh, H-J, Muriel, A., and Balasubramanian, H., A user-friendly Excel simulation for scheduling in primary care practices, Proceedings of the Winter Simulation Conference, 2014, In.: Proceedings of the 2014 Winter Simulation Conference A. Tolk, S. Y. Diallo, I. O. Ryzhov, L. Yilmaz, S. Buckley, and J. A. Miller, eds BURKE, E. K., CAUSMAECKER P., BERGHE, G., LANDEGHEM, H., The state of the art of nurse rostering, Journal of Scheduling 7: 441-499, 2004. ?2004 Kluwer Academic Publishers Meignan D., Knust S., Frayret J, Pesant G., and Gaud N. 2015. A review and taxonomy of interactive optimization methods in operations research. ACM Trans. Interact. Intell. Syst. 5, DOI: <http://dx.doi.org/10.1145/2808234> Patrick De Causmaecker, Greet Vanden Berghe. A categorization of nurse rostering problem. 2010 <http://www.springerlink.com/content/a821277755438620/fulltext.pdf>

Podpis studenta:

Datum:

Podpis vedoucího práce:

Datum: