

Analýza a návrh informačního systému pro zubní ordinace

Bakalářská práce

Vedoucí práce:

doc. Ing Ivana Rábová, Ph.D.

Jiří Fiala

Brno 2016

Poděkování

Rád bych zde poděkoval především paní doc. Ing. Ivaně Rábové, Ph.D. za vedení, cenné rady a připomínky při tvorbě této bakalářské práce.

Čestné prohlášení

Prohlašuji, že jsem tuto práci: **Analýza a návrh informačního systému zubní ordinace**

vypracoval/a samostatně a veškeré použité prameny a informace jsou uvedeny v seznamu použité literatury. Souhlasím, aby moje práce byla zveřejněna v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů, a v souladu s platnou *Směrnicí o zveřejňování vysokoškolských závěrečných prací*.

Jsem si vědom/a, že se na moji práci vztahuje zákon č. 121/2000 Sb., autorský zákon, a že Mendelova univerzita v Brně má právo na uzavření licenční smlouvy a užití této práce jako školního díla podle § 60 odst. 1 Autorského zákona.

Dále se zavazuji, že před sepsáním licenční smlouvy o využití díla jinou osobou (subjektem) si vyžádám písemné stanovisko univerzity o tom, že předmetná licenční smlouva není v rozporu s oprávněnými zájmy univerzity, a zavazuji se uhradit případný příspěvek na úhradu nákladů spojených se vznikem díla, a to až do jejich skutečné výše.

V Brně dne 2.ledna 2017

Abstrakt

Fiala, J. Analysis and design of information system for dental surgery. Bachelor thesis. Brno: Mendel University, 2016

This bachelor thesis is interested about analysis and design of information system for dental surgery. For the creating analysis and design will be used object oriented approach. Based on the results of UML diagrams in CASE tools will be created a prototype of information system. Final solution will be implemented using by script language PHP, HTML, cascading style sheets and database system MySQL. A part of bachelor thesis gonna be an economic assessment of information system.

Klíčová slova

UML, information systems, dental surgery, PHP, MySQL

Abstrakt

Fiala, J. Analýza a návrh informačního systému pro zubní ordinace. Bakalářská práce. Brno: Mendelova univerzita v Brně, 2016.

Tato bakalářská práce se zabývá analýzou a návrhem prototypu informačního systému používaného v zubních ordinacích. Pro tvorbu analýzy a návrhu bude využit objektově orientovaný přístup. Na základě výsledků z UML diagramů v CASE nástroji Visual Paradigm bude vytvořen prototyp modelu informačního systému. Výsledné řešení bude implementováno pomocí skriptovacího jazyka PHP, HTML, kaskádových stylů a databázového systému MySQL. Součástí práce bude v závěru provedeno ekonomické zhodnocení.

Klíčová slova

UML, informační systémy, zubní ordinace, PHP, MySQL

Obsah

1	Úvod a cíl práce	11
1.1	Úvod.....	11
1.2	Cíl a metodika práce.....	11
2	Literární rešerše	12
2.1	Základní principy tvorby IS/ICT.....	12
2.1.1	Obecné principy metodik tvorby IS	13
2.2	Životní cyklus IS/ICT	14
2.3	Modely životního cyklu.....	15
2.3.1	Model prototyp	15
2.3.2	Inkrementální dodávání.....	16
2.3.3	Boehmův spirálový model.....	16
2.3.4	Proces RUP.....	17
2.4	Architektura IS	18
2.4.1	Pohledy na architekturu.....	18
2.4.2	Architektonické vzory	19
2.5	Modelování interakcí v IS.....	20
2.5.1	Use case diagram.....	20
2.5.2	Sekvenční diagramy.....	21
2.5.3	Diagram aktivit.....	21
2.6	Modelování tříd.....	22
2.6.1	Generalizace tříd.....	22
2.6.2	Relace.....	23
2.7	Spolehlivost a bezpečnost systému.....	23
2.7.1	Dostupnost a spolehlivost.....	23
2.7.2	Bezpečnost.....	23
2.7.3	Zabezpečení.....	24
3	Použité vývojové nástroje	25
3.1	UML.....	25

3.1.1	Struktura UML.....	25
3.2	HTML & CSS.....	25
3.2.1	HTML.....	25
3.2.2	CSS.....	25
3.3	PHP.....	26
3.4	Latte.....	26
3.5	MySQL.....	26
4	IS ve zdravotnictví	27
4.1	Dentist +.....	28
4.2	AIS – Ambulantní informační systém	29
4.3	MEDICUS Stomatolog.....	30
5	Požadavky na systém	31
5.1	Funkční požadavky.....	31
5.2	Nefunkční požadavky	31
6	Návrh řešení	32
6.1	Popis aktérů.....	32
6.2	Use-case diagram	34
6.3	Popis Use-case	34
6.4	Scénáře případů užití.....	35
6.5	Class diagram.....	37
6.6	ER diagram.....	38
7	Implementace	39
7.1	Návrh databáze	39
7.2	Rozvržení stránky	40
7.3	Funkcionalita	41
7.3.1	Lékař	41
7.3.2	Pacient.....	42
8	Diskuse	44
8.1	Návrh HW zařízení.....	44
8.2	Ekonomické zhodnocení	44

8.3	Možnosti rozšíření	45
9	Závěr	46
10	Literatura	47
A	Obsah přiloženého CD	50
B	Ukázky výsledné práce	51

Seznam obrázků

OBR. 1	MODEL VÝVOJE IS/IT ZDROJ: HTTP://WIKI.VALEK.NET/LIB/EXE/FETCH.PHP?W=350&TOK=8EDD40&MEDIA=STATNICE:SLOVNIK:MMDIS_-_DIMENZE.JPG	12
OBR. 2	PROCES VÝVOJE PROTOTYPU. ZDROJ: HTTP://IMAGES.SLIDEPLAYER.CZ/10/2913472/SLIDES/SLIDE_23.JPG	15
OBR. 3	BOEHMŮV SPIRÁLOVÝ MODEL. ZDROJ: HTTP://TESTOVANISOFTWARU.CZ/WP-CONTENT/UPLOADS/2011/07/SPIRALA.PNG	16
OBR. 4	FÁZE PROCESU RUP ZDROJ: HTTP://DOCPLAYER.CZ/DOCS-IMAGES/24/3551036/IMAGES/20-0.PNG	17
OBR. 5	MODEL SOFTWAREVÉ ARCHITEKTURY 4+1 ZDROJ: HTTP://WWW.TREETEK.COM/PORTAL2/IMAGES/4+1B.JPG 19	
OBR. 6	IS DENTIST+ VYSTAVENÍ FAKTURY ZDROJ: HTTP://WWW.DENTIST.CZ/O-PROGRAMU-DENTIST/FAKTURA_VYSTAVENA.PNG	28
OBR. 7	IS AIS - KARTA PACIENTA ZDROJ: HTTP://WWW.CTMOS.CZ/CM/UPLOADS/IMAGES/AIS/KARTA_PACIENTA.PNG 29	
OBR. 8	MEDICUS - AMBULANTNÍ DOKLAD ZDROJ: HTTP://WWW.MEDICUS.CZ/STOMATOLOGIE/PROGRAMY/MEDICUS-STOMATOLOG-START/	30
OBR. 9	UŽIVATELÉ - GENERALIZACE.....	32
OBR. 10	USE CASE DIAGRAM.....	34
OBR. 11	CLASS DIAGRAM	37
OBR. 12	ER DIAGRAM	38
OBR. 13	SQL PŘÍKAZY.....	39
OBR. 14	ÚVODNÍ OBRAZOVKA.....	40
OBR. 15	OBRAZOVKA PO PŘIHLÁŠENÍ DO IS.....	41

Seznam tabulek

TAB. 1 SCÉNÁŘ - PŘIHLÁŠENÍ DO SYSTÉMU	35
TAB. 2 SCÉNÁŘ - OBJEDNÁVKA ZÁSOB	36
TAB. 3 SCÉNÁŘ - OBJEDNAT SI SCHŮZKU	36
TAB. 4 CENOVÉ POROVNÁNÍ KONKURENČNÍCH PRODUKTŮ	45

1 Úvod a cíl práce

1.1 Úvod

V posledních letech prochází svět informačních a komunikačních technologií obrovským vývojem. Věci, které si naše prarodiče neuměli ani představit jsou dnes realitou. To platí i ve zdravotnictví. Nevyvíjí se pouze lékařské procedury a výzkum, ale také komunikace mezi jednotlivými lékaři, odděleními a pacienty. Doby, kdy kartotéka zabírala půlku místnosti a v případě návštěvy více lékařů bylo nutné nosit sebou zdravotní kartu a spoustu jiných papírů již neplatí. Dnes jakmile lékař provede nějaký úkon, nebo předepíše léky, tak se to ihned prostřednictvím integrovaného informačního systému zobrazí ostatním institucím, které těchto informací využijí. Ostatní lékaři budou znát detailně pacientovu anamnézu, lékárníkům se ihned zobrazí, jaké léky mají vydat a v případě pojistných událostí se požadované papíry odešlou automaticky příslušným organizacím.

1.2 Cíl a metodika práce

Cílem této práce je návrh nového informačního systému pro zubní ordinaci. Po seznámení se současným stavem je nutné definovat požadavky na nový systém, který bude splňovat nároky a to jak z pozice lékaře, tak i pacienta. Práce bude obsahovat teoretickou část, kde se seznámíme se základy informačních systémů a jejich architekturou. Bude popsán životní cyklus systému, jeho vývoj a zabezpečení.

K vývoji praktické části budeme vycházet z výsledků analýz současného stavu a konzultací s lékaři z ambulantní praxe. Po zjištění dostupných informací definujeme funkční i nefunkční požadavky, které jsou na systém kladeny. Na základě výsledků z analýz vytvoříme za použití UML diagramů návrh možného řešení, který nám poskytne statický a dynamický pohled na systém. Ve výsledném modelu se zaměříme na zachycení jednotlivých toků dat a způsobu jejich zpracování.

Výsledkem bude webová aplikace, která bude sloužit uživatelům k přihlášení do systému a zároveň k propagaci zubní ordinace. Pacienti budou mít v aplikaci omezená práva, budou moci například vidět svojí zdravotní kartu a budou schopni se objednávat na pravidelné kontroly. Lékaři a zdravotní sestry, již budou mít plný přístup k funkcionalitě, kterou bude informační systém nabízet. Budou moci evidovat informace o stávajících pacientech i vytvářet složky o nově příchozích. Lékaři budou moci komunikovat s externím systémem laboratoře, kam budou moci zasílat požadavky. Další částí bude administrace, ve které bude nutné zajistit, aby si lékař mohl objednávat potřebné zboží a materiály u dodavatelů, evidovat faktury a další věci spojené s vedením ordinace.

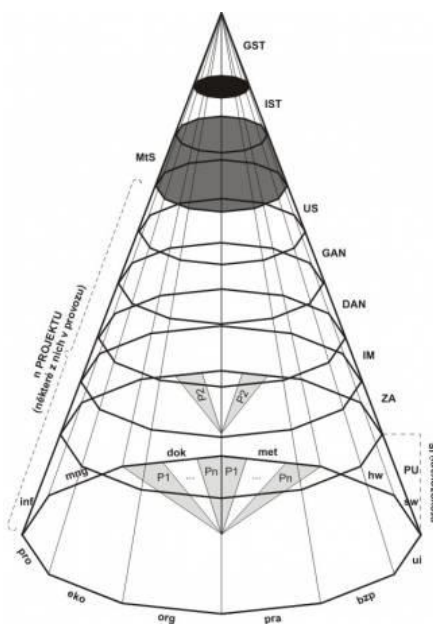
2 Literární rešerše

Informační systém je komplex informací, lidí, použitých informačních technologií, organizace práce, řízení chodu informací a metod, které slouží ke sběru, přenosu a následnému zpracování dat. (Rábová, 2008).

2.1 Základní principy tvorby IS/ICT

Metodika tvorby informačního systému se zabývá globálními aspekty procesu vývoje IS. Jsou definovány jednotlivé etapy vývoje a vztahy mezi nimi. Jedná se o souhrn zásad, pravidel a nástrojů pro tvůrce informačních systémů, který pokrývá celý životní cyklus informačních systémů a určuje co, kdy a proč se má dělat během vývoje a provozu IS (Chlapek, Řepa, Stanovská, 2011).

Každá část vývoje a provozu IS/ICT je abstrakcí celku, která se detailně zabývá danou problematikou. Jednotlivé části od sebe nelze časově ani věcně oddělit, protože se natolik navzájem ovlivňují, že jedna bez ostatních nemůže existovat. (Voříšek, 2002). Úlohou metodiky v každé etapě životního cyklu IS je definovat význam a náplň jednotlivých částí a zajistit za pomoci příslušných technik podstatné souvislosti a celkovou konzistenci.



Obr. 1 Obrázek 1 Model vývoje IS/IT

Zdroj:http://wiki.valek.net/lib/exe/fetch.php?w=350&tok=8edd40&media=statnice:slovník:mmdis_-_dimenze.jpg

Mezi největší přínosy zavedení metodiky tvorby informačních systémů patří především zvýšení kvality vyvíjených systémů, což zvyšuje i konkurenceschopnost firmy, která je využívá.

2.1.1 Obecné principy metodik tvorby IS

Bez ohledu na využití informačního systému, by měla každá tvorba vycházet z následujících principů:

1. Orientace na cíle a problémy

2. Účast zadavatele projektu

Vedení organizace, pro kterou je systém vytvářen by mělo být aktivně zapojeno během celého vývoje IS.

3. Klíčové produkty a jejich schvalování

Výsledkem činnosti při vývoji systému jsou dokumentace vývoje a průběžné zprávy o vývoji, které se předávají vedení organizace. Bez schválení jejich správnosti není možné ve vývoji pokračovat. Toto je také jeden ze způsobů zajištění aktivní spolupráce vedení na projektu.

4. Zapojení uživatele do návrhu

5. Modelování a abstrakce (princip tří architektur)

Vytvoření tří úrovní modelů systému (konceptuální, technologická, implementační) vede k jasnému oddělení podstaty systému od dodatečně přidávaných omezení a implementačních změn. Důsledkem je snížení rizika nepříznivých dopadů změn na návrh systému.

6. Ověřování a testování návrhu během celého vývoje

Na konci každé činnosti se ověřuje, zda výsledky odpovídají cílům organizace a požadavkům uživatelů, dále potom zda jsou v pořádku po formální i logické stránce.

7. Analýza a návrh

V jednotlivých etapách se analyzují požadavky na systém, které jsou specifikovány do takové míry, aby bylo možné na základě takovéto analýzy pokračovat v etapě následující. V případě změny nadřazeného požadavku je vždy zapotřebí se vrátit k etapě předchozí, kde se požadavek poprvé projeví a ze které se dále vyplývající změny promítnou do následujících etap.

8. Otevřenost metodiky

Metodiky tvorby IS by měli být postaveny na běžně používaných technikách vývoje. To umožňuje i užití dostupných počítačových prostředků pro vývoj IS (nástroje CASE). Každá metodika by tak měla být do značné míry kompatibilní se standardy definovanými pro danou oblast. V současné době je jedním takovým standardem pro modelování UML. (Buchalcevová, 2004)

2.2 Životní cyklus IS/ICT

Životní cyklus vymezuje základní etapy vývoje IS a jejich obsah. Začíná prvotním nápadem něco řešit nebo vylepšit za pomoci IS/ICT a to v souladu s informační strategií podniku (Rábová, 2008). Rozdělení celého vývoje a provozu podnikového systému do etap, je úzce spojeno s používanou metodikou.

Životní cyklus se rozděluje na následující části: (Jaroslav Žáček [online])

- **Předběžná analýza (specifikace cílů)**
Základem celkového návrhu vývoje systému jsou požadavky uživatelů a cíle organizace. V této části se musí požadavky shromáždit, rozebrat a odhadnout doba realizace s potřebnými náklady na vývoj. V konečném dokumentu této části je specifikován účel systému, jsou identifikováni uživatelé a jejich požadavky, jsou definovány části systému a návrh jejich řešení.
- **Analýza systému (specifikace požadavků)**
Jedná se o podrobný rozbor jednotlivých částí z předběžné analýzy.
- **Projektová studie (návrh)**
Tato část je výsledkem analýzy systému, zde jsou jednotlivé požadavky na systém specifikovány až na úroveň, kdy je možné začít navržený systém implementovat. Projektová studie by měla obsahovat:
 - Základní informace o tvůrcích systému
 - Základní informace o organizaci, pro kterou je systém vyvíjen
 - Popis současného stavu organizace
 - Logický datový model, obsahující návrh funkcí a dat systému
 - Fyzický model, který obsahuje funkční analýzu systému
 - Detailní popis nasazení IS v praxi
 - Detailní popis testovacího provozu systému
- **Implementace**
Tato část životního cyklu je vlastním programováním. Na základě získaných faktů z fyzického návrhu se definují vstupy a výstupy jednotlivých operací. Naprogramují se veškeré funkce a doladí se jejich vzájemné propojení. Nakonec se jednotlivé funkce ověří pomocí připravených dat, která by měla obsahovat alespoň procento z konečných reálných dat.
- **Testování**
V této etapě se provádí připravené testy na hotovém IS. Je nutné vyzkoušet veškeré reakce systému na zadávaná data a nalezené nedostatky popřípadě odstranit. Testování systému se provádí na systému, který ještě není v reálném prostředí, neboť případné chyby v systému mohou mít rozsáhlé následky.
- **Zavádění systému**
Zavedením systému spočívá v instalaci a zapojení do provozu organizace.

2.3 Modely životního cyklu

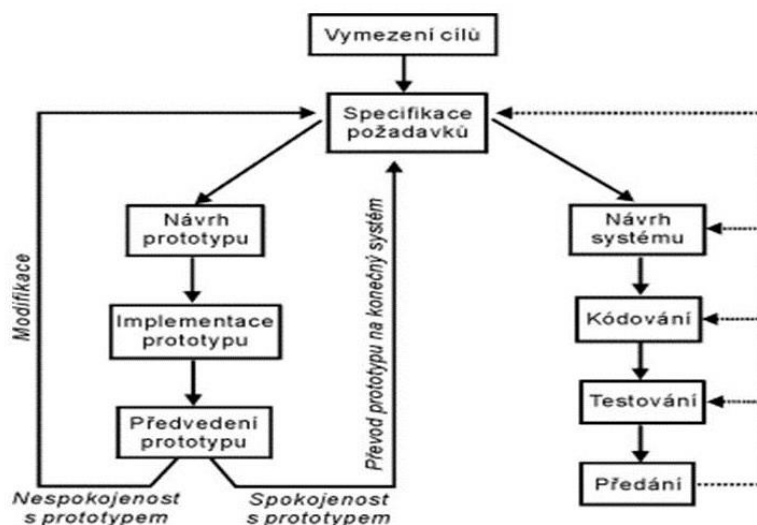
Modely určují postup při tvorbě systémů, zvolená podoba výsledných modelů je závislá na rozhodnutí manažera projektu a charakteru celého týmu. Modely jsou jádrem metodik a určují posloupnosti aktivit v metodikách (Rábová, 2008)

2.3.1 Model prototyp

Systémové prototypy umožňují uživatelům zjistit, nakolik jim systém dokáže pomoci při jejich práci. Pomáhá vývojářům při odhalování chyb a opomenutí v již navržených požadavcích, je schopen rozpoznat silné a slabé stránky systému. Systémový prototyp použitý při návrhu systému může sloužit k návrhovým experimentům, které ověří proveditelnost výsledného návrhu. (Žáček,[online])

Cíle prototypu by měli být zřejmé již na začátku celého procesu. Jediný prototyp nemůže obsahovat veškeré cíle celého projektu. Pokud cíle nejsou stanoveny, nemusí vedoucí pracovníci, či koncoví uživatelé prototypu rozumět správně. V důsledku toho nemusí vývoj prototypu poskytnout očekávané výhody.

Závěrečnou fází prototypu je jeho hodnocení. V této fázi je nutné zajistit uživatelská školení. Uživatelé potřebují čas, aby se s novým systémem obeznámili. Jakmile začnou systém normálně používat, mohou se objevit různé chyby v požadavcích, které bude potřeba vyřešit.(Sommerville,2013)



Obr. 2 Obrázek 2 Proces vývoje prototypu.

Zdroj:http://images.slideplayer.cz/10/2913472/slides/slide_23.jpg

2.3.2 Inkrementální dodávání

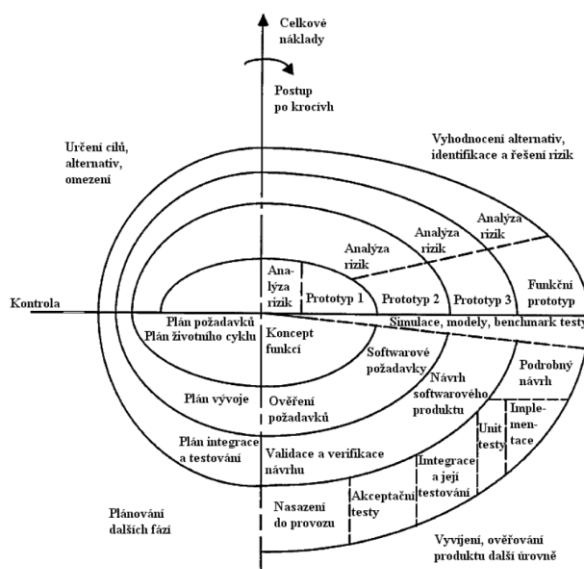
Inkrementální dodávání je přístup k vývoji softwaru, kdy se některé z vyvinutých inkrementů dodávají zákazníkovi a nasazují se k používání v provozním prostředí. Zákazníci si potom určují, které služby jsou pro ně více důležité a které ne. Po identifikaci důležitých inkrementů systému lze podrobně definovat požadavky na služby a zahájit vývoj (Sommerville,2013).

U některých systémů není tento přístup nejvhodnější, jedná se zejména o velmi rozsáhlé systémy, které může vyvíjet více týmů pracujících v různých lokalitách. Tyto systémy se také musí vypořádat s nejasnými a proměnlivými požadavky. Aby bylo možné tyto problémy vyřešit a získat určité výhody inkrementálního vývoje, lze použít proces, který slouží jako platforma pro experimenty s požadavky na systém a jeho návrhem. (Žáček, [online])

2.3.3 Boehmův spirálový model

Softwarový proces je zde reprezentován pomocí spirály a každá smyčka zde představuje jednu fázi softwarového procesu. Tento model kombinuje prevenci a toleranci změn. Předpokládá, že změny jsou výsledkem rizik projektu a zahrnuje explicitní aktivity řízení rizik, které je mají omezovat. (Sommerville,2013)

Hlavní rozdíl spirálového modelu od jiných softwarových procesů je v explicitním rozpoznání rizik. Cyklus spirály začíná rozborem cílů a poté se sepisují alternativní způsoby, jak těchto cílů dosáhnout. Každá alternativa se hodnotí vzhledem k příslušnému cíli a identifikují se zdroje projektových rizik. (Sommerville, 2013)



Obr. 3 Obrázek 3 Boehmův spirálový model.

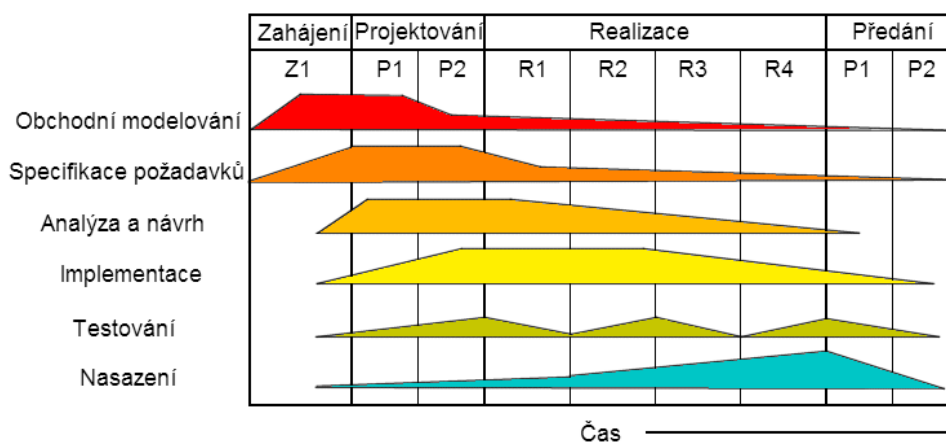
Zdroj: <http://testovanisoftware.cz/wp-content/uploads/2011/07/spiral.png>

2.3.4 Proces RUP

Rational Unified Process. Jedná se o příklad moderního modelu procesů, který kombinuje prvky ze všech obecných procesů. Oproti běžným modelům RUP popisuje proces ze tří perspektiv, které popisují optimální postupy softwarového inženýrství. (Krutchen, 2003)

- Dynamická perspektiva, která znázorňuje fáze modelu v průběhu času
- Statická perspektiva, která zobrazuje odehrávající se aktivity procesu
- Perspektiva postupů, která doporučuje optimální postupy pro nasazení v procesu

Model RUP definuje čtyři oddělené fáze softwarového procesu. Oproti vodopádovému modelu, kde fáze odpovídají aktivitám procesu, zde jednotlivé fáze spíše souvisejí s podnikovými hledisky. Jedná se o fáze: (Sommerville, 2013)



Obr. 4 Obrázek 4 Fáze procesu RUP

Zdroj: <http://docplayer.cz/docs-images/24/3551036/images/20-0.png>

1. Zahájení

Cílem fáze zahájení je vytvořit podnikový případ příslušného systému. Je zapotřebí identifikovat všechny externí entity, které budou se systémem v interakci.

2. Projektování

V této fázi se rozvíjí znalosti problémové domény, jsou vytvořeny schémata architektury, projektový plán a projektová rizika. Po dokončení této fáze by měl existovat model požadavků na systém, popis architektury a plán vývoje softwaru.

3. Realizace

Fáze realizace zahrnuje návrh systému, programování a testování. Na konci by měl být k dispozici funkční softwarový systém a související dokumentace v podobě, která může být dodána uživatelům

4. Předání

Závěrečná fáze řeší předání systému od vývojové části k uživatelské komunitě a zprovoznění systému v reálném prostředí. Po dokončení této fáze by měl být správně fungující dokumentovaný softwarový systém v provozním prostředí. (Sommerville, 2013)

2.4 Architektura IS

Návrh architektury umožňuje porozumět tomu, jak má být systém uspořádán a jak by měla vypadat jeho struktura. Jedná se o důležitou spojnicu mezi návrhem inženýrských požadavků, identifikujících hlavní strukturní komponenty a vztahů mezi nimi. Výstupem návrhu architektury je model, který popisuje, jak je systém organizován do sady komunikujících komponent. (Arlow, 2007)

Architektura softwaru je důležitá, protože její návrh má podstatný dopad na samotný výkon systému a možnosti údržby. Architekturu je možné navrhovat ve dvou úrovních abstrakce, které můžeme označit za **architekturu v malém** a **architekturu ve velkém rozsahu**:

- Architektura v malém se zabývá architekturou jednotlivých programů. Na této úrovni nás zajímá, jakým způsobem lze jednotlivé programy rozložit na komponenty. (Sommerville, 2013)
- Architektura ve velkém se týká architektury komplexních podnikových systémů, které zahrnují další systémy, programy a programové komponenty. Tyto podnikové systémy se distribuují do různých počítačů, které mohou být majetkem ve více různých společnostech. (Sommerville, 2013)

2.4.1 Pohledy na architekturu

Jediný model architektury nedokáže reprezentovat všechny podstatné informace o architektuře modelovaného systému, protože každý model znázorňuje pouze určitý pohled na systém. Může vysvětlovat, jak se systém dělí na modely, jak se systém dělí na modely, jak mezi sebou reagují run-time procesy, nebo znázorňovat jak jsou jednotlivé komponenty distribuovány v síti. (Sommerville, 2013)

V praxi se koncepční pohledy vždy vyvíjejí během procesu návrhu a slouží k podpoře ohledně architektury. Krutchen ve svém modelu softwarové architektury s pohledem 4+1 navrhuje, že je vhodné používat čtyři základní pohledy na architekturu. (Krutchen, 1995)

1. Logický pohled

Představuje klíčové abstrakce v systému jako objekty nebo objektové třídy. Systémové požadavky by mělo být možné vztáhnout na entity tohoto logického pohledu.

2. Procesní pohled

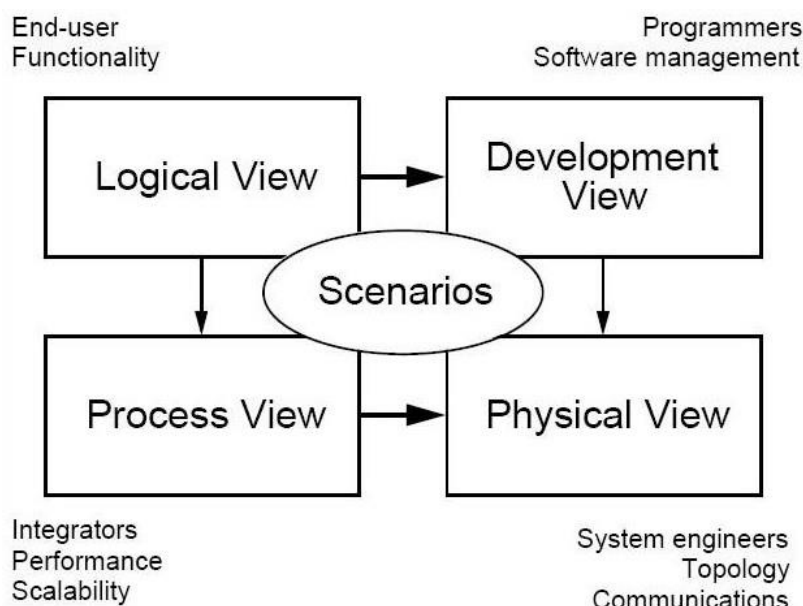
Znázorňuje, jak se ve fázi run-time systém skládá z mezi sebou reagujících procesů. Tento pohled se hodí k posuzování mimo funkčních vlastností systému.

3. Vývojový pohled

Informuje o dekompozici softwaru při vývoji. Charakterizuje rozložení softwaru na komponenty, které implementují jednotliví vývojářské týmy. Tento pohled je užitečný pro softwarové manažery a programátory

4. Fyzický pohled

Popisuje systémový hardware a způsob, jakým jsou softwarové komponenty rozděleny mezi procesory v systému. Tento pohled je vhodný pro techniky, kteří plánují nasazení systému. (Krutchen, 1995)



Obr. 5 Obrázek 5 Model softwarové architektury 4+1
Zdroj: <http://www.treetek.com/portal2/images/4+1b.jpg>

2.4.2 Architektonické vzory

Architektonický vzor je stylizovaný abstraktní popis optimálního postupu, který byl vyzkoušen a otestován v různých systémech. Architektonický vzor by měl tedy popisovat organizaci systému, která byla v předchozích systémech úspěšná. Měl by zahrnovat informace o tom, kdy je vhodné daný vzor použít a kdy nikoli. (Sommerville, 2013)

1. Vrstevnatá architektura

Vzor vrstevnaté architektury představuje způsob jak dosáhnout nezávislosti, protože odděluje prvky systému a umožňuje jejich nezávislou změnu. Tuto ar-

chitekturu je možné změnit a přenášet, za předpokladu že se nezmění rozhraní, je možné určitou vrstvu nahradit ekvivalentní vrstvou. (Sommerville,2013)

2. Architektura uložště

Vzor uložště popisuje, jak mezi sebou může sada reagujících komponent sdílet data. Tento model se hodí pro aplikace, ve kterých jedna komponenta data generuje a jiná je používá. Není nutné explicitně přenášet data z jedné komponenty do druhé, avšak jednotlivé komponenty musí být kompatibilní s dohodnutým datovým modelem uložště. (Sommerville,2013)

3. Architektura klient/server

Systémy, které využívají vzor klient/server, jsou uspořádány jako sada služeb a přidružených serverů. Důležitou výhodou představuje oddělení a nezávislost, jednotlivé služby a servery je možné měnit, aniž by to ovlivnilo jiné části systému. (Sommerville,2013)

4. Architektura kanál a filtr

Jedná se o model run-time organizace systému, kde funkční transformace zpracovávají své vstupy a produkují výstupy. Data proudí od jedné transformace k další a při postupu celou sekvencí dochází k jejich zpracování. Příklady této architektury se využívají často v aplikacích dávkového zpracování. (Sommerville,2013)

2.5 Modelování interakcí v IS

Identifikace a definování funkcionality je dalším krokem při specifikaci a analýze vyvíjeného systému. (Chlapek, Řepa, Stanovská, 2011) Ve všech systémech dochází k různým interakcím, může se jednat o interakce s uživateli, s jednotlivými komponentami v systému nebo dokonce s jinými systémy. Modelování interakcí je velmi důležité, protože pomáhá identifikovat uživatelské požadavky a odhaluje, jaké problémy mohou nastat při komunikaci. (Sommerville, 2013) Při vývoji aplikace budeme využívat objektově orientovaný přístup, proto veškeré modely budou tvořeny pomocí UML diagramů, které budou vytvářeny za pomoci Visual Paradigmu.

2.5.1 Use case diagram

Use case diagramy jsou jedním z hlavních diagramů jazyka UML, který zobrazuje chování systému, tak jak to požaduje uživatel. Zachycují funkčnost a požadavky, které bude informační systém vykonávat. Každý use case diagram však popisuje pouze jeden z požadavků systému, tedy jednu jeho požadovanou funkčnost. (Kanišová, Müller, 2006)

Use case diagramy simulují využití reálného systému externími aktory. V rámci tvorby těchto diagramů modelujeme vztahy systému a jeho okolí. (Chlapek, Řepa, Stanovská, 2011) K tomu jsou využívány následující prvky.

- Aktéři

Aktér představuje roli, kterou určitá externí entita přijímá v okamžiku, kdy začíná daný systém bezprostředně používat. (Arlow, 2007) Aktéři spouštějí případy užití a v systému může jeden aktér provádět řadu případu užití a zároveň jeden případ užití může být vykonáván více aktéry. (Kanisová, Müller, 2006)

V UML diagramech bývají aktéři označováni symbolem postavičky, avšak nemusí jimi být vždy lidé. Jako aktér může v modelovaném systému vystupovat i jiný systém nebo čas. Čas bývá aktérem v situacích, kdy potřebuje v systému provádět určité operace, např. zálohy nebo aktualizace vždy v přesnou dobu.

- Use case

Use case by se dal popsat jako soubor scénářů, které vedou k uspokojení požadavků jednoho nebo více aktérů.

- Hranice systému

Hranice systému oddělují systém od okolního prostředí, při tvorbě je důležité definovat co je součástí systému a co naopak není. V UML2 se hranice systému také označují jako subjekt, který definuje, kdo bude systém využívat a jaký přínos z použití bude mít (Arlow, 2007)

Hranice systému jsou v diagramech vyjádřeny jako rámeček s popisem obsahující název systém. Aktéři jsou vyznačeni mimo vyznačené území, zatímco případy užití jsou uvnitř hranic.

2.5.2 Sekvenční diagramy

Sekvenční diagramy graficky znázorňují interakce mezi čarami života a průběh zpracování informací v podobě zasílaných zpráv. Tyto diagramy jsou nejbohatší a nejpružnější formou diagramů interakce. Nejčastěji zobrazují chování a spolupráci jednotlivých objektů v rámci jednoho případu použití. (Arlow, 2007)

Zprávy mohou být v sekvenčním diagramu posílány jak mezi jednotlivými objekty, třídami či dokonce aktéry. Proto prvky, které mezi sebou v diagramu komunikují, se nazývají souhrnně klasifikátory. Z každého klasifikátoru vede tzv. lifeline, který reprezentuje, jakým způsobem se instance příslušného klasifikátoru účastní interakce.

2.5.3 Diagram aktivit

Jedná se o objektově orientované vývojové diagramy, které se skládají z kolekce uzlů spojených hranami. Diagramy aktivit lze připojit k libovolnému modelovanému prvku a definovat jeho chování. Obvykle jsou připojeny k případům užití, třídám, rozhraním nebo operacím. (Arlow, 2007)

Aktivity se skládají ze sítí uzlů spojených hranami. Při modelování rozlišujeme tři kategorie uzlů: (Arlow, 2007)

1. Akční uzly – reprezentují samostatné a v rámci dané aktivity neoddělitelné jednotky

2. Řídící uzly – mají za úkol řídit cestu uvnitř aktivity
3. Objektové uzly – zastupují konkrétní objekty použité v rámci dotyčné aktivity

Hrany znázorňují cestu v rámci aktivity a v diagramech rozlišujeme dvě kategorie hran: (Arlow, 2007)

1. Řídící hrany – zastupují postup řízení v rámci aktivity
2. Objektové hrany – zastupují cestu objektů v rámci aktivity

2.6 Modelování tříd

UML diagramy tříd jsou základem všech prostředků objektové analýzy a návrhu. Umožňují zachytit statickou strukturu reality systému. Zobrazují třídy, jejich vzájemné vztahy, atributy a operace. Diagramy tříd lze v jazyce UML vyjádřit na více úrovních podle míry abstrakce. Když jsme v první fázi vývoje modelu, obvykle se podíváme do reálného světa, kde identifikujeme klíčové objekty a reprezentujeme je formou tříd. (Sommerville, 2013)

Před začátkem modelování diagramu tříd, je zapotřebí si ujasnit pro jaké potřeby budeme diagram vytvářet. Jestli nám diagram bude sloužit jako podklad pro modelovaný software, nebo jako popis designu. Podle jeho využití se dělí na tyto úrovně abstrakce modelu tříd. (Arlow, 2007)

- Konceptuální model
Tento typ modelu je tvořen za účelem analýzy požadavků na software. Obsahuje pouze business classes, které modelují problémovou oblast. U jednotlivých tříd se uvádí obvykle jen názvy klíčových atributů a metod
- Designový model
Vychází s konceptuálního modelu, který rozšiřuje o viditelnost atributů, metod nebo datových typů. Dále jsou do modelu přidány uživatelská rozhraní a třídy obsahující systémové události.
- Implementační model
Při tvorbě implementačního modelu je výhodné reprezentovat jednotlivé třídy modelu co nejjednodušším způsobem. Třídy v modelu obsahují pouze informace o atributech a metodách. (Arlow, 2007)

2.6.1 Generalizace tříd

Generalizace je běžná metoda, díky které jsme schopni zvládat řešit komplexní problémy abstrakce. Místo abychom se seznamovali s podrobnými vlastnostmi každé entity, řadíme je do obecnějších tříd podle jejich společných vlastností. (Sommerville, 2013) Generalizace se znázorňuje pomocí šipky, která směřuje k obecnější třídě. Z toho je zřejmé určit na jaké úrovni abstrakce se třída nachází a jaké třídy ji dále rozšiřují.

2.6.2 Relace

Relace umožňují ukázat na modelu, jaký je vztah mezi dvěma třídami. Asociace je vztah mezi třídami, který je charakterizován násobností. Násobnost určuje, kolik objektů jedné třídy může být ve vztahu s jedním objektem druhé třídy. Každá asociace má svůj název a může existovat mezi dvěma třídami nebo v rámci jedné třídy, tzv. **rekurzivní asociace**.

Vazby mezi jednotlivými třídami mohou být vyjádřeny pomocí speciálními typy asociace, kompozicí nebo agregací, které vyjadřují pevnější spojení mezi jednotlivými třídami.

2.7 Spolehlivost a bezpečnost systému

V dnešní době, kdy se počítačové systémy staly neoddělitelnou součástí našeho osobního i pracovního života, je zásadně důležité zajistit jejich spolehlivost a bezpečnost. Většina dnes používaných systémů obsahuje pro uživatele velmi citlivá data, například hodnoty bankovních účtů, osobní údaje nebo výrobní postupy firem.

Spolehlivost počítačového systému je vlastností systému, která odráží jeho důvěryhodnost. (Sommerville, 2013) To v zásadě znamená důvěru uživatelů, že systém bude fungovat dle jejich očekávání a během běžného používání neselže. Avšak spolehlivost systému je příliš obecný pojem, proto jsou specifikovány další čtyři dimenze.

2.7.1 Dostupnost a spolehlivost

Dostupnost a spolehlivost systému jsou úzce související vlastnosti, které lze vyjádřit jako pravděpodobnost, že systém bude v libovolnou funkční dobu schopen reagovat na požadavky uživatelů a dodávat jim správné odpovědi.

Definice spolehlivosti se vztahuje k implementaci daného systému a k jeho specifikaci. Systém se chová spolehlivě, jestliže je jeho chování konzistentní s tím, jak bylo definováno v jeho specifikaci. Dostupnost však nezávisí jen na počtu havárií systému, ale také na tom, jak dlouho trvá odstranění chyb a znovuoobnovení systému. (Sommerville, 2013)

2.7.2 Bezpečnost

Systémy pro které je zásadně důležité zajistit neustálou bezpečnost, se nazývají bezpečnostně kritické. To znamená, že systém by ani při svém selhání nikdy neměl ohrozit osoby ve svém prostředí, nebo své okolí. (Sommerville, 2013) Mezi bezpečnostně kritické systémy patří například řídicí a monitorovací systémy v letadlech, systémy řízení procesů v továrnách, nebo automobilové aktivní systémy.

Bezpečnostně kritický software se dále člení do dvou tříd:

1. Primární bezpečnostně kritický software

Software, který je integrován do systému jako jeho aktivně řídicí prvek. Chyba takového softwaru může vést k chybě hardwaru, která má za následek zranění osob nebo škody na svém prostředí. (Sommerville,2013)

2. Sekundární bezpečnostně kritický software

Tento typ software může způsobit zranění nepřímo. Příkladem takového softwaru může být například systém konstrukčních návrhů, jehož chyba může způsobit vady při navrhování nových objektů. (Somerville, 2013)

2.7.3 Zabezpečení

Zabezpečení je schopnost systému chránit se před externími útoky, které mohou být náhodné nebo úmyslné. (Sommerville,2013) Téměř všechny systémy dnes fungují online a výměna informací je prováděna prostřednictvím sítě, kde k nim prostřednictvím virů a trojských koní mohou přistupovat vzdálení uživatelé.

U každého systému připojeného k síti existují tři hlavní typy hrozeb zabezpečení: (Pfleeger a Pfleeger, 2007)

1. Hrozby důvěrnosti systému a jeho dat

Tyto hrozby mohou vést k tomu, že se informace dostanou k osobám či programům, které nejsou pro přístup k daným informacím autorizovány

2. Hrozby integrity systému a jeho dat

Jedná se o hrozby, které mohou poškodit nebo zničit software a jeho příslušná data

3. Hrozby dostupnosti systému a jeho dat

Do této kategorie patří hrozby, které mohou omezit přístup autorizovaným uživatelům k softwaru nebo k jeho datům (Pfleeger a Pfleeger, 2007)

U některých systémů je zabezpečení nejdůležitější dimenzí spolehlivosti systému, protože uchovávají natolik citlivé informace, které se nesmějí dostat do neautorizovaných rukou. Pro takové systémy bez rozumné úrovně zabezpečení nemůžeme mít jistotu, že systém bude dostupný, spolehlivý a bezpečný.

3 Použité vývojové nástroje

3.1 UML

Jedná se o univerzální jazyk, který vznikl za účelem vizuálního modelování systémů. Byl navržen, aby spojil nejlepší postupy modelovacích technik a softwarového inženýrství. Je důležité si uvědomit, že UML nenabízí žádný druh metodiky modelování. Samotný jazyk UML však poskytuje pouze vizuální syntaxi, kterou můžeme využít při sestavování jednotlivých modelů. Unified Process již naopak metodikou je. Vyjadřuje nám například, jaké pracovníky musíme využít, jaké činnosti vykonat, abychom byli schopni sestavit funkční model softwarového systému. (Arlow,2007)

3.1.1 Struktura UML

Funkci jazyka UML jako vizuálního jazyka porozumíme nejlépe za pomoci jeho struktury, která se skládá z těchto částí:

1. Stavební bloky – základní prvky modelu, relace a diagramy
2. Společné mechanismy – obecné způsoby, jimiž v jazyku UML dosáhneme předem zadaných cílů
3. Architektura – pohled na architekturu navrhovaného systému v jazyku UML (Arlow,2007)

3.2 HTML & CSS

3.2.1 HTML

HTML vzniklo v 90. letech ve výzkumném středisku CERN a současně s ním i standard WWW. Jazyk vznikl pro sdílení interních dokumentů a rozlišení jednotlivých částí dokumentu. Jak se internet rozšiřoval mezi lidi, rozvíjelo se i HTML a začaly se přidávat nové značky. (ITnetwork, [online])

Jazyk HTML (Hyper Text Markup Language) je základem pro tvorbu webových stránek skládající se ze značek, tzv. tagů a jejich atributů, které umožňují dodávat jednotlivým prvkům na stránce určitý význam. HTML se dříve používalo i na grafické stylování stránek, protože vzniklé weby byly nepřehledné, byl omezen pouze na webový obsah.

3.2.2 CSS

CSS (Cascading style sheets) je jazyk, který byl speciálně vyvinutý pro stylování HTML stránek. CSS umožňuje nastavit vzhled jednoho až několika dokumentů, takže změna daného stylu na jednom místě se projeví i ve všech ostatních dokumentech. Dále jsme schopni přesně definovat styly pro různá média odděleně od dokumentu. Stránky, které mají oddělenou definici stylu, zabírají mnohem méně místa v paměti, což činí jejich načítání rychlejší. (ITnetwork, [online])

3.3 PHP

PHP je skriptovací jazyk používaný pro vytváření dynamických webových stránek a webových aplikací. Při používání PHP k tvorbě webových stránek jsou používány skripty, které je možné volat z příkazového řádku, dotazovacích metod http nebo pomocí webových služeb. (Wikipedia, [online])

Díky velkému rozvoji internetu a využití webových stránek se hledaly způsoby, jak obohatit statické stránky o dynamickou funkčnost. Výsledkem tohoto vývoje je, že v dnešní době jsme schopni dosáhnout toho, aby se webová stránka chovala úplně stejně, jako desktopová aplikace. Takovým stránkám říkáme webové aplikace a mají mnoho výhod: (ITnetwork, [online])

1. Snadná správa

Výslednou verzi aplikace nahrajeme na server a v tom okamžiku k ní mají přístup i všichni ostatní

2. Vysoká bezpečnost

Webové stránky i databáze jsou uloženy na serveru, který je chráněn. Pokud neobsahuje bezpečnostní chyby, měla by být naše aplikace v bezpečí

3. Velký uživatelský potenciál

Výhodnou webových aplikací je, že se oproti desktopovým nemusí instalovat, stačí načíst požadovanou stránku a můžeme s aplikací ihned pracovat.

4. Vysoká kompatibilita

Jelikož aplikaci spouštíme přes webový prohlížeč, nemusíme brát v úvahu uživateleův operační systém. Aplikace jsou spustitelné prakticky téměř odkudkoliv a dokonce i z mobilních zařízení.

3.4 Latte

Jedná se o šablonovací systém, který je určený pro jazyk PHP kde pomáhá vývojářům zefektivnit práci a zabezpečit výstup před napadením.

3.5 MySQL

MySQL je relační databázový systém, vytvořený švédskou společností MySQL AB. Jedná se o open-source databázi, což znamená, že lidé po celém světě mohou opravovat chyby nebo přidávat svá vlastní vylepšení.

Každá databáze vytvořená v MySQL je tvořena jednou nebo více tabulkami, které jsou spolu vzájemně propojeny. Tabulky mají jeden nebo několik řádků a sloupců. Do řádků se zapisují jednotlivé záznamy a sloupce udávají jméno záznamu. Jednotlivé sloupce jsou podmíněny předem definovaným datovým typem, který se nastavuje na začátku při vytváření tabulky, mezi nejběžnější typy patří například text, číslo nebo datum.

4 IS ve zdravotnictví

Nemocniční systémy patří k základním pilířům eHealth. Tento pojem je velmi obecný, protože pod něj spadá správa a využití zdravotních systémů. Součástí jsou i například elektronické zdravotní záznamy, elektronické předepisování léků, poskytování zdravotních informací apod. Mezi služby elektronického zdravotnictví spadá také vztahy mezi pacienty a jejich lékaři, předávání dokumentace mezi jednotlivými zdravotnickými zařízeními nebo mezi lékařskými odborníky. Specifickou náplní jsou služby telemedicíny, které umožňují pacientům dálkovou komunikaci s lékařem. To zahrnuje možnosti monitoringu pacientů prostřednictvím internetu nebo mobilních aplikací. (eHealth,[online])

Cílem státní správy je vybudovat centrální informační systém, který by se aplikoval do všech zdravotnických zařízení v České republice. V reálném použití by takové řešení bylo velmi nákladné a pravděpodobnost na jeho realizaci velmi nízká. Znamenalo by to zrušení všech doposud používaných systémů a nahradit je novým jednotným systémem. O problematice eHealth se hovoří nejen na národní úrovni, ale do projektu jsou zapojeny i instituce členských států Evropské unie. Mezi hlavní výhody digitalizace ve zdravotnictví patří efektivní uložení informací a pozdější přístup k nim. Tím by mělo dojít k účinnějšímu léčení a omezení plýtvání nadbytečných zdrojů a duplicitní péče, a to vše díky dostupným informacím získaným z jednotlivých subsystémů (eGovernment [online]). Tím že v databázích budou přesné záznamy o provedených vyšetření a předepsaných lécích si ostatní lékaři mohou ověřit vývoj léčby a porovnat své zjištění s předchozími lékaři se zamezí rostoucím nákladům za provedené výkony. Dále lze minimalizovat náklady při kontrole předepsaných lécích, kdy lékař uvidí, jaké přípravky pacient užívá, v jakém množství a bude schopen posoudit, zda některé přípravky nekolidují s jinými a zda se při jejich užívání dostává požadovaného cíle.

V České republice k pokusům zavedení elektronického zdravotnictví patří projekt IZIP. Tento projekt je založen na existenci elektronických zdravotních knížek pacientů na internetu. Elektronická zdravotní knížka může obsahovat veškeré informace o zdravotním stavu pacienta. V rámci rozvoje eGovernmentu lze nyní internetovou zdravotní knížku zřídit mimo jiné i na pobočkách České pošty, se službou Czech POINT. Tento projekt je prozatím dostupný pro pojištěnce Všeobecné zdravotní pojišťovny a je volně přístupný na internetu prostřednictvím přístupového ID a hesla. (eGovernment [online])

V následujících podkapitolách jsou vybrány některé informační systémy používané ve zdravotnictví a především ve stomatologických ambulancích.

4.1 Dentist +

Program Dentist + je určen pro malé i větší zubní ordinace, nebo také pro stomatologické laboratoře. Umožňuje vedení kompletní zdravotní dokumentace od založení pacienta do evidence, jeho registrace k příslušné zdravotní pojišťovně, přes vykázaní zdravotních úkonů se zápisem do zdravotní karty a zakreslení značek v grafickém zubním modelu až po vyúčtování provedené péče zdravotní pojišťovně. Ovládání programu je optimalizované jako ve většině aplikací a je funkční ve všech verzích systémů Windows. (Dentist+,[online])

The screenshot shows the 'Vystavená Faktura' window with the following data:

Supplier (Dodavatel): Soukromá zubní praxe Zoubek & company, Adresa Čermákova 123 Zlín, 76001 Zlín 1, IČO 12345678, DIČ 123456789, Banka 9999.

Customer (Odběratel): (Empty fields)

Invoice Details: Rok - Pořadí faktury: 2004-1, Zformátované číslo: 2004-1, Datum vystavení: 31.12.2004, Datum plnění: 31.12.2004, Splatnost (počet dní): (empty), Forma úhrady: (empty).

Kód	Název položky	MJ	Počet	Jednotková cena (základ)
*	Dávka č. 1, za období 12/2...	Ks	1,000	480,00000

Summary: Celková suma: 480,00 Kč, Haléřové vyrovnání: 0,00 Kč, Zálohově uhrazeno: 0,00 Kč, Suma k úhradě: 480,00 Kč, Úhrazená suma: 0,00 Kč, Zůstává uhradit: 480,00 Kč.

Obr. 6 Obrázek 6 IS Dentist+ vystavení faktury

Zdroj: http://www.dentist.cz/o-programu-dentist/faktura_vystavena.png

4.2 AIS – Ambulantní informační systém

Program AIS je informační systém, určený pro vedení administrativy lékařských ordinací, rehabilitačních pracovišť i poliklinik. Mezi přednosti programu patří jednoduché grafické rozhraní a přehledné vedení zdravotnické dokumentace. Systém dále umožňuje vytváření vlastních formulářů a zajišťuje komunikaci se systémem IZIP. Součástí je i komplexní agenda pro kontakt se zdravotními pojišťovnami a zubními laboratořemi. (CTMOS,[online])

Rodné č.: 6156261254	Příjmení: Adamcová	Jméno: Marie	Titul:	
Pojišťovna: 111	číslo pojistky:	Narozen(a): 26.06.1961	Pohlaví: žena	
Registrace: 1.9.2005	=A=	Stát. přís:	Stav: svobodná	
<input type="radio"/> Trvalé bydl. <input type="radio"/> Přechodné bydl. <input type="radio"/> Kontakt		Připojištění: 1	<input checked="" type="radio"/> Doručovací adresa <input type="radio"/> Trvalé bydliště <input type="radio"/> Přechodné bydliště	
Ulice: Národní třída		čp: 7	Telefon:	
PSČ: 110 00	Obec: Praha	Část:	Reg. do IZIP	
<input type="radio"/> Zaměstnání <input type="radio"/> Poznámka <input type="radio"/> Ostatní <input type="radio"/> Telefon a e-mail		Péče: pravidelná		
Zaměstnavatel:	Zetor Brno a.s.	Pracovní zařazení:	účetní	
Zaměstnavatel:		Pracovní zařazení:		
<input type="radio"/> Alergie <input type="radio"/> Riziko		Krevní skupina: Neznámé	<input type="button"/> Tlak <input type="button"/> Hm. <input type="button"/> Obv. <input type="button"/> Výš. <input type="button"/> Očk.	
Alergie na pyl		Disp. sk: D,E10	TAT:	
Kód	Základní diagnózy	Rok	Měsíc	Disp
K739	Chronická hepatitis, NS	2005	1	E10

Obr. 7 Obrázek 7 IS AIS - karta pacienta

Zdroj: http://www.ctmos.cz/cm/uploads/images/AIS/karta_pacienta.png

4.3 MEDICUS Stomatolog

Ve verzi START se jedná o jednoduchý informační systém pro nenáročného uživatele. Vede jednoduchou zdravotní dokumentaci pacientů, umožňuje vystavovat recepty a účtuje za poskytnuté zákroky pojišťovně. Díky snadnému a intuitivnímu ovládní usnadní a především urychlí administrativní práci lékaře. Zároveň poskytuje lékaři užitečný nástroj pro získání základního přehledu o vykázaných výkonech a předepsaných léčích. (MEDICUS,[online])

The screenshot shows the 'Stomatologický ambulantní doklad' window. At the top, there are menu items: Nový, Zruš, Další, Jiný, Výrobky, Poplatek, Nastavení, and Ilačítka. Below the menu, patient information is entered: Rodné číslo: 880808/8883, Příjmení: Abel, Jméno: Jan, Pojišťovna: 111, Druh poj.: 1, IČP: 11-111-014, Odb.: 014, Variabilní symbol: (empty), Základní dg: K021, Ostatní dg: (empty), Náhrady: (empty). The main table lists dental procedures:

Datum	Výkon	Lok.	Poč.	Dg.	Název	Body	Mater.	Odb.	K.	Pacient	Nositel
25.11.2011	00921	53	1	K021	OŠETŘENÍ ZUBNÍHO KAZU -	0	220,00			0,00	
25.11.2011	00950	84	1	K051	EXTRAKCE STÁLÉHO ZUBU	0	168,00			0,00	
25.11.2011	09543		1	K021	REGULAČNÍ POPLATEK ZA N	0	0,00			0,00	NOJ

At the bottom, there is a summary row: Součty hodnot, 0, 388,00, 0,00. The footer text is: Zdravotnické zařízení Novák s.r.o.\D-oddělení\014-STD.

Obr. 8 Obrázek 8 MEDICUS - ambulantní doklad

Zdroj: <http://www.medicus.cz/stomatologie/programy/medicus-stomatolog-start/>

5 Požadavky na systém

Ještě před začátkem tvorby objektově orientované analýzy návrhu, je důležité mít přehled o tom, čeho chceme při tvorbě systému dosáhnout, jaký je smysl požadavků a jejich specifikace.

Hlavním cílem informačních systémů ve zdravotnictví je dosáhnout co nejlepší podpory v oblasti péče o pacienta a schopnost poskytnout důležité údaje v případě potřeby. Z technického hlediska jsou informační systém kladeny především tyto požadavky: správné ukládání dat, spolehlivost, rychlá dostupnost a bezpečnost dat. Data musejí být uchována na bezpečném místě a systém kontroluje, kdo a za jakých podmínek k nim bude mít přístup.

5.1 Funkční požadavky

Popisují služby, které by měl systém poskytovat, reakce systému na určité vstupy a chování systému v určitých situacích. V některých případech mohou také explicitně vyjadřovat, co systém provádět nesmí. (Sommerville,2013) V námi navrhovaném systému budou funkční požadavky následující:

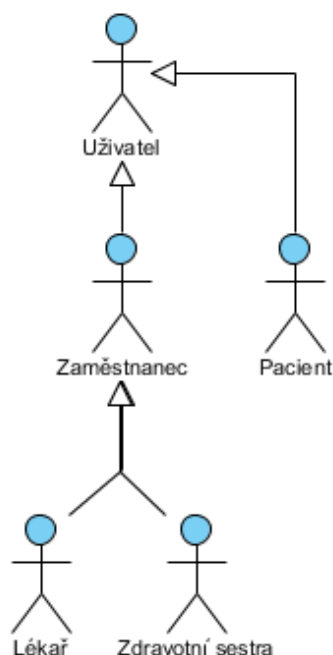
- Ze strany zaměstnance
 - Evidence a příjem nových pacientů
 - Přehled o provedených zákrocích u jednotlivých pacientů
 - Evidence zásob a možnost vytváření objednávek
 - Evidence spolupracujících pojišťoven a dodavatelů
 - Evidence a vytváření nových faktur
 - Tisk reportů (pojišťovnam, objednávek, faktur)
- Ze strany pacienta
 - Evidence osobní karty
 - Možnost domluvení prohlídky u lékaře
 - Editace osobních a přihlašovacích údajů

5.2 Nefunkční požadavky

Jedná se o omezení služeb či funkcí, které systém poskytuje. Nefunkční požadavky se často týkají systému jako celku a nikoli jeho jednotlivých funkcí a služeb. (Sommerville,2013) V navrhovaném systému definujeme:

- Jednoduché a intuitivní uživatelské prostředí
- Chráněné přihlašovací údaje uživatelů informačního systému
- Nízké náklady na vytvoření a provoz informačního systému
- Zajištění pravidelných záloh dat a aktualizací

6 Návrh řešení



Obr. 9 Obrázek 9 Uživatelé - generalizace

6.1 Popis aktérů

- **Uživatel**
Uživatel je kdokoliv kdo využívá informační systém. Uživatelé se do systému přihlašují prostřednictvím přihlašovacího jména a hesla. Po přihlášení mohou systém plně využívat podle předem nastavených práv
- **Pacient**
Pacienti se do systému sami nemohou registrovat, ale na první schůzce u lékaře dostanou přihlašovací údaje, které mohou později sami změnit. V systému mají přístup ke své zdravotní kartě, kde mohou vidět, jaké procedury na nich lékař v minulosti prováděl a kolik za ně bylo účtováno. Dále mohou upravovat a měnit své osobní údaje.

- Zaměstnanec

Tito uživatelé mají v systému po přihlášení už podstatně větší možnosti. Zaměstnanci se starají o objednávky zásob, které jsou nutné k provozu ordinace. S objednávkou souvisí i evidence, díky které mají přehled o množství jednotlivých nástrojů a zásob. Zaměstnanci mají na starosti také fakturaci, kde se fakturují služby pacientům, nebo za dodané zboží od dodavatelů. Faktury za poskytnuté služby mohou být hrazeny přímo zaměstnanci nebo jsou posílány k příslušné pojišťovně.

- Zdravotní sestra

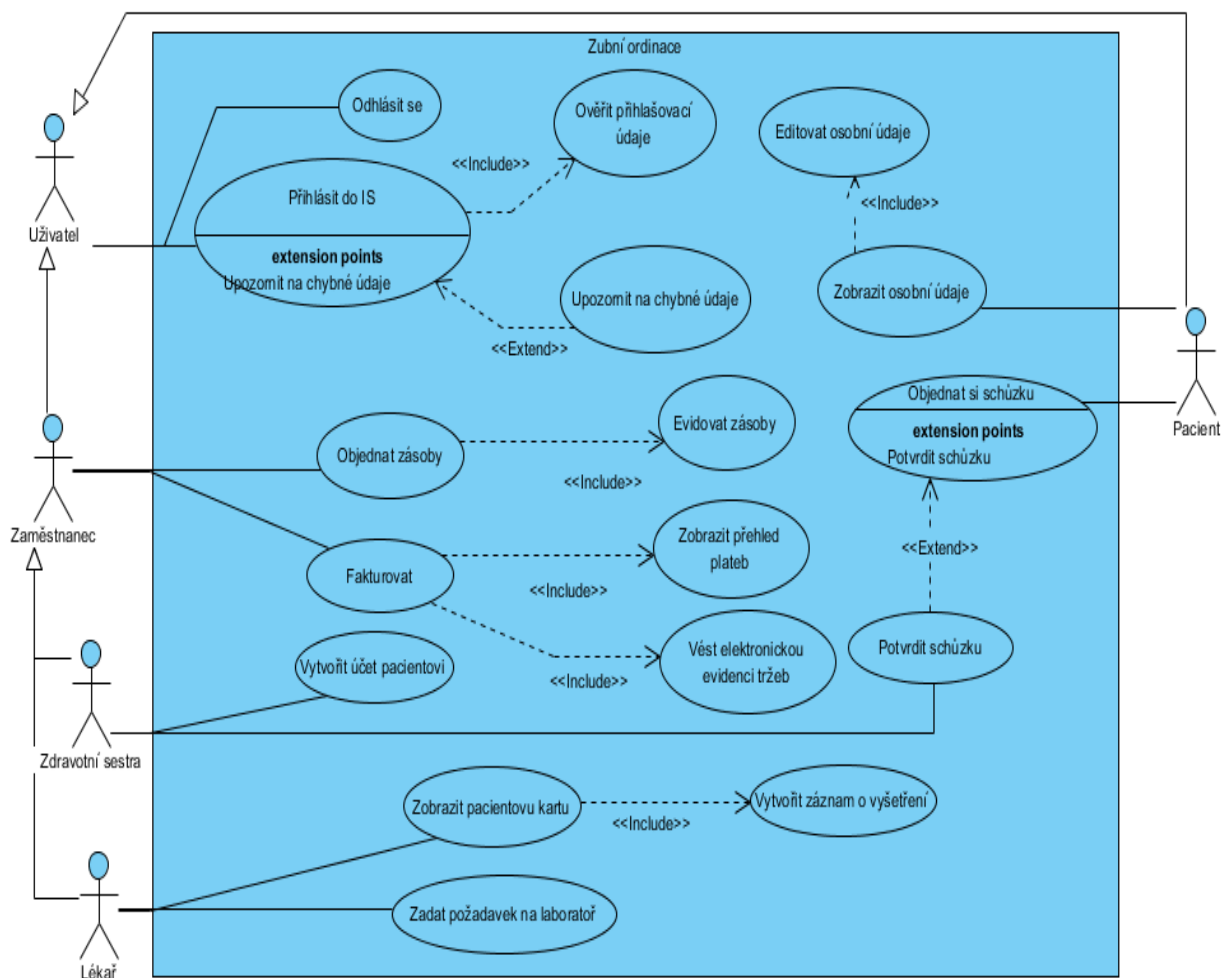
Zdravotní sestra, jakož to zaměstnanec ordinace má přístup k většině funkcionality, kterou systém disponuje. Mimo výše zmíněných možností, kterými disponují zaměstnanci, sestra vytváří účty nově příchozím pacientům a potvrzuje objednávky schůzek pacientů. Jakmile se pacient objedná na prohlídku, tak mu zdravotní sestra termín potvrdí.

- Lékař

Lékaři mají neomezený přístup v systému, mohou kontrolovat výše zmíněné funkce plus mohou nahlížet do osobních karet pacientů. Zde si zobrazí jejich anamnézu a již provedené zákroky. V průběhu zákroku, mohou evidovat jejich výkon, jaké použili nástroje nebo jaké medikamenty při vyšetření použili. V případě potřeby mohou posílat požadavky do laboratoře, ohledně zhotovení dentálních modelů.

6.2 Use-case diagram

Diagram zachycuje komunikaci systému s externími entitami a předkládá vývojákům představu o tom, jaké interakce budou v systému během jeho používání probíhat.



Obr. 10 Obrázek 10 Use case diagram

6.3 Popis Use-case

- Objednávka zásob

Lékaři nebo zdravotní sestry mohou prostřednictvím systému objednávat potřebné zásoby nebo lékařské vybavení od svých dodavatelů. S tím souvisí i možnost evidence současného stavu, kdy budou mít přehled o aktuálním stavu jednotlivého materiálu.

- Fakturace
S provozem ordinace souvisí také spousta plateb. Proto zde bude navržena možnost vytváření faktur za poskytnuté služby, nebo naopak přijímání faktur od dodavatelů za přijaté zboží. Faktury budou posílány příslušným pojišťovám, v případě speciálních procedur budou hrazeny přímo pacienti.
- Elektronická evidence plateb
V souvislosti se současným politickým stavem, kdy od 1. září vyšel v platnost zákon o elektronické evidenci tržeb, která by zatím neměla postihnout ordinace, jsem do svého systému navrhl modul, který by měl v budoucnu lékařům ušetřit čas a práci.
- Objednat si schůzku
Pacienti se budou moci prostřednictvím systému objednávat na pravidelné kontroly.
- Přihlásit do IS
Všichni uživatelé se musí do systému přihlásit pomocí předem vygenerovaného přihlašovacího jména a hesla. Zaměstnanci mají přihlašovací údaje explicitně zadané, zatímco pacientům jsou přihlašovací údaje zadány, avšak později si je mohou změnit podle svých preferencí.

6.4 Scénáře případů užití

1. Přihlášení do systému

Scénář k případu užití přihlášení do systému popisuje sled interakcí mezi uživatelem a systémem při přihlášení do systému

Tab. 1 Tabulka 1 Scénář - přihlášení do systému

Krok	Actor Input	System response
1	Uživatel klikne v navigačním panelu na kartu přihlášení	
2		System zobrazí uživateli přihlašovací formulář
3	Uživatel zadá své přihlašovací údaje a potvrdí přihlášení	
4		System ověří, zda byly zadány přihlašovací údaje a následně i jejich platnost.
5	Uživatel potvrdí přihlášení	

2. Objednat zásoby

Scénář k případu užití objednat zásoby popisuje sled interakcí mezi lékařem a systémem při evidenci zásob a následného objednání.

Tab. 2 Tabulka 2 Scénář - Objednávka zásob

Krok	Actor Input	System response
1	Zaměstnanec klikne na kartu přehled zásob	
2		System zobrazí uživateli přehled jednotlivých zásob na skladě a jejich množství
3	V případě potřeby – zaměstnanec klikne na objednat zboží	
4		System zobrazí objednávkový formulář pro příslušný druh zboží
5	Zaměstnanec vyplní dodatečné informace k objednávce	
6		System vytvoří novou objednávku
7	Zaměstnanec si zobrazí report objednávky a odešle příslušnému dodavateli zboží	

3. Objednat si schůzku

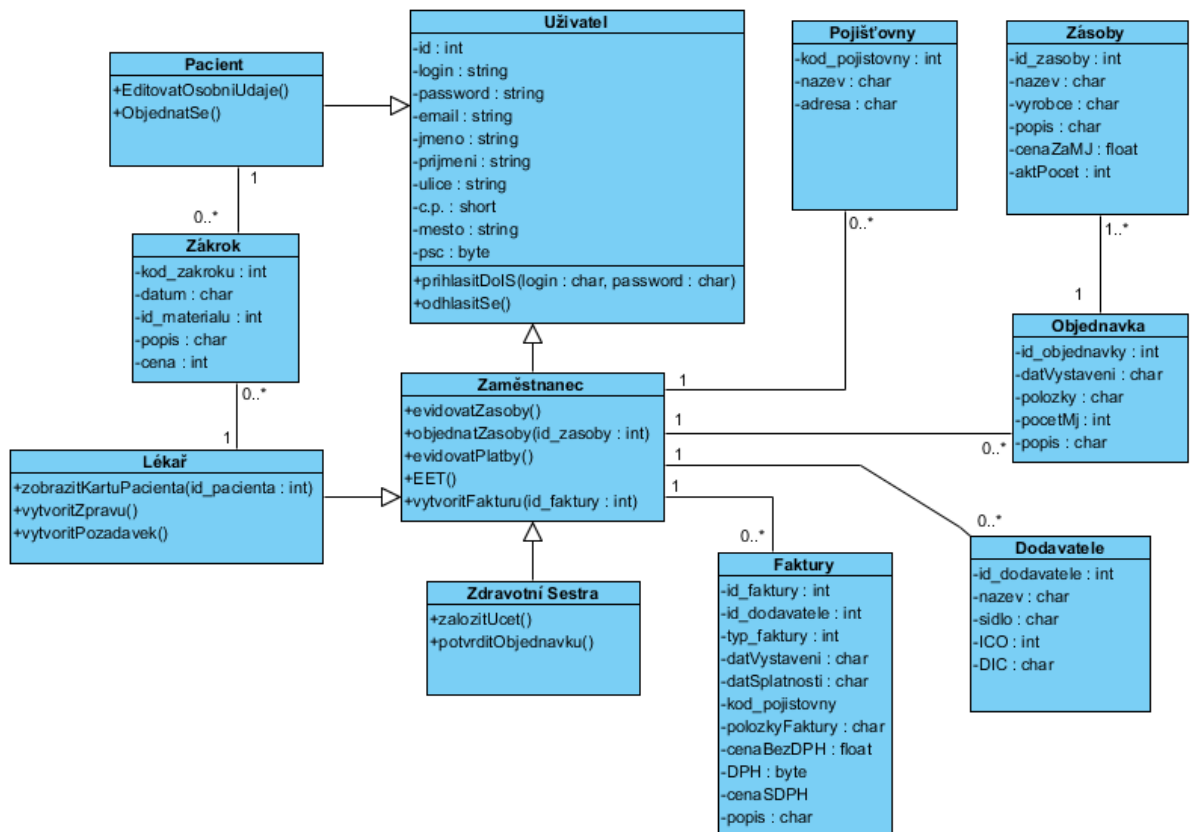
Scénář k případu užití objednat zásoby popisuje sled interakcí mezi pacientem a systémem při objednání se na prohlídku.

Tab. 3 Tabulka 3 Scénář - objednat si schůzku

Krok	Actor Input	System response
1	Pacient klikne na kartu objednat schůzku	
2		System zobrazí uživateli možnosti termínů
3	Pacient si vybere termín a čas podle svých preferencí	
4		System ověří, že vybraný termín byl validně zadán
5	Zaměstnanec potvrdí vybraný termín	
6		System odešle lékaři informace o nové prohlídce pacienta

6.5 Class diagram

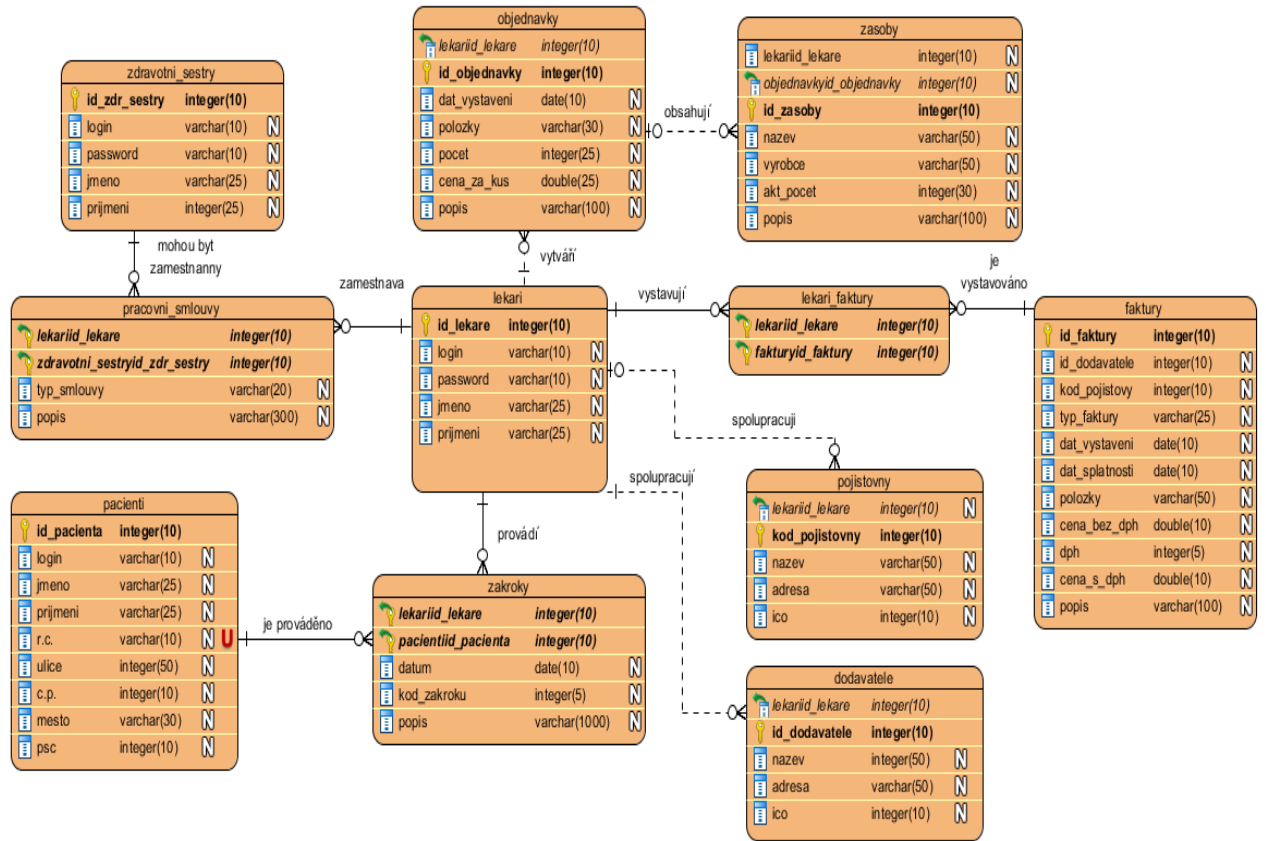
Diagram tříd zachycuje statickou část systému.



Obr. 11 Obrázek 11 Class diagram

6.6 ER diagram

Entitní relační diagram zobrazující databázový model a jeho strukturu.



Obr. 12 Obrázek 12 ER diagram

7 Implementace

Výsledný prototyp aplikace je dostupný k nahlédnutí a otestování na této adrese: <https://akela.mendelu.cz/~xfiala8/WWW/xfiala8/>

7.1 Návrh databáze

Databáze je nedílnou součástí naší aplikace, jsou v ní uloženy informace o pacientech, lékařích a dalších entitách spolupracující s ordinací. V současné době přichází v úvahu několik vhodných variant k výběru databáze.

Ve vlastním projektu probíhal návrh databáze nejdříve pomocí lokálního serveru XAMPP. Tento balíček umožňuje vytvoření virtuálního Apache serveru a podporuje PHP a MySQL databázi, ke které je i v instalaci připojený phpMyAdmin. Po vytvoření celé databáze a naplnění tabulek testovacími daty bylo nutné kvůli zajištění univerzality a přenositelnosti aplikace využít free hostingu. Pro aplikaci jsem využil FreeMySQLhosting, který nabízí prostor 50MB a zajišťuje pohodlný přístup a bezpečnost dat.

Výsledná podoba celé databáze, jednotlivých tabulek a vazeb mezi nimi odpovídá výše uvedenému ER diagramu. Pro vytvoření databáze a vytvoření jednotlivých tabulek je provedeno na základě těchto příkazů:

```
1
2 CREATE DATABASE BP_XFIALA8 CHARACTER SET utf8 COLLATE utf8_czech_ci;
3
4
5
6 CREATE TABLE pacienti
7 (
8 id_pacienta INTEGER NOT NULL ,
9 login VARCHAR(10) NOT NULL ,
10 heslo VARCHAR(10) NOT NULL ,
11 jmeno VARCHAR(25) NOT NULL ,
12 prijmeni VARCHAR(25) NOT NULL,
13 r_c INTEGER(15) NOT NULL,
14 ulice VARCHAR(50) NOT NULL,
15 c_p integer(20),
16 mesto VARCHAR(50) NOT NULL,
17 psc integer(10),
18 kontakt VARCHAR(50),
19 email VARCHAR(50)
20 );
21 ALTER TABLE pacienti ADD CONSTRAINT pacienti_PK PRIMARY KEY ( id_pacienta ) ;
22 ALTER TABLE pacienti ADD CONSTRAINT pacienti_UN UNIQUE ( r_c ) ;
23
24
```

Obr. 13 Obrázek 13 SQL příkazy

7.2 Rozvržení stránky

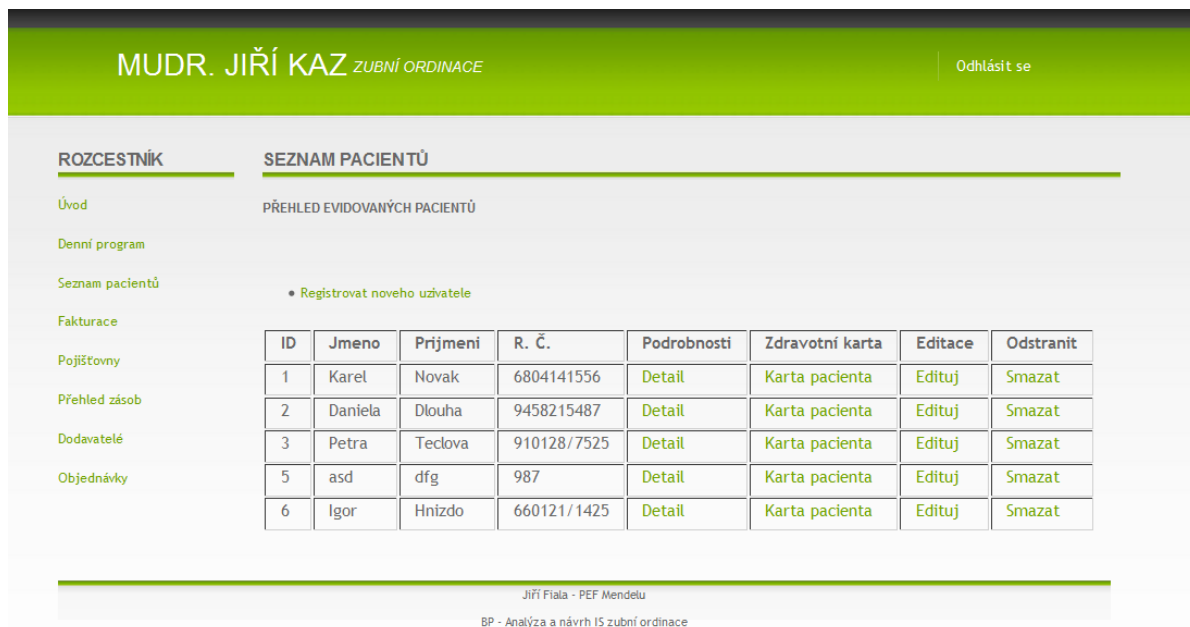
Pro webovou aplikaci byly vytvořeny tři různé layouty, které odlišují situaci, kde se uživatel právě nachází a umožňují mu provádět příslušné operace.

První layout je určen prezentaci ordinace, po spuštění stránky je zobrazena úvodní obrazovka, která je rozdělena do tří částí. V horní je logo a název ordinace spolu s horizontálním navigačním panelem. Hlavní část stránky je rozdělena na obsah a postranní pravý panel, který obsahuje plugin propojující web ordinace s jeho facebookovou stránkou.



Obr. 14 Obrázek 14 Úvodní obrazovka

Další rozvržení stránek je určeno až na práci v samotném systému a je upraven zvláště pro pacienty a lékaře. Navigační panel se kvůli přehlednosti mění na vertikální a přesouvá se na levou stranu obrazovky. V horní části stále zůstává název ordinace a z navigačního menu zde zůstává pouze tlačítka pro odhlášení ze systému.



Obr. 15 Obrázek 15 Obrazovka po přihlášení do IS

Více ukázek obrazovek je k dispozici v příloze.

7.3 Funkcionalita

Práce s informačním systémem začíná ihned po přihlášení, které je uzpůsobeno zvláště pro zaměstnance ordinace, tak pro pacienty. Pacienti se do aplikace nemohou sami registrovat, ale domlouvají si prostřednictvím formuláře vstupní prohlídku, kde jim jsou kromě samotné prohlídky vytvořeny i přihlašovací údaje do systému.

7.3.1 Lékař

Lékaři po přihlášení do systému mají následující možnosti:

1. Zobrazení denního programu

Po rozkliknutí možnosti denní program, se lékaři zobrazí seznam naplánovaných schůzek na daný den. Seznam je řazen podle času a obsahuje ID pacienta, který má na schůzku přijít, jeho kartu a důvod domluvené schůzky. Po zobrazení karty, může lékař vkládat nové zápisy a editovat stávající. Po vyřízení prohlídky se schůzka z denního programu odstraní.

2. Seznam pacientů

Zde jsou zobrazeny jednotlivé záznamy o pacientech, kteří pravidelně dochází do ordinace. Lékař zde má možnost zobrazit si podrobnosti o jednotlivých pacientech, jejich zdravotní karty. Dále je zde také možnost editovat údaje, přijímat nové pacienty a vyřazovat stávající.

3. Fakturace

V záložce fakturace jsou zobrazeny jednotlivé faktury, vydané nebo přijaté, které byly v interakci s ordinací. V záložce jsou jednotlivé záznamy řazeny podle data splatnosti a lékař si u každého záznamu může zobrazit jeho detail, nebo zobrazit požadovanou fakturu ve formátu pdf, kterou si může následně uložit nebo vytisknout.

4. Pojišťovny

Zde jsou záznamy o jednotlivých pojišťovnách, které spolupracují s ordinací a jsou jim fakturovány poskytnuté zákroky. Lékař může přidat nově spolupracující pojišťovnu.

5. Přehled zásob

V záložce zásoby jsou zobrazeny jednotlivé položky, se kterými lékaři v ordinaci pracují, nebo je jinak využívají. U každého záznamu je možné si zobrazit jeho detail, nebo objednat další kusy. V případě objednání je lékař přesměrován na vytvoření nové objednávky a kromě doobjednání dalších kusů může lékař přijmout i úplně nové zásoby nebo nástroje.

6. Dodavatelé

Zde jsou zobrazeni jednotliví dodavatelé, kteří spolupracují s ordinací, lékař si může zobrazit podrobnosti o jednotlivém dodavateli nebo přidávat nové.

7. Objednávky

Záložka objednávky funguje na podobném principu jako fakturace, lékařovi jsou zde zobrazeny jednotlivé objednávky zásob. U každého záznamu si může zobrazit její detail, jaké položky daná objednávka obsahuje a jakému dodavateli bude odesílána. Dále si také lékař může objednávku zobrazit ve formátu pdf, kterou může rovněž jako fakturu uložit nebo vytisknout.

7.3.2 Pacient

Jak již bylo výše zmíněno, pacient se nemůže sám registrovat do systému a přihlašuje se prostřednictvím vygenerovaných přihlašovacích údajů, které získá na úvodní prohlídce. Jeho možnosti v systému jsou následující:

1. Osobní údaje

V záložce osobní údaje jsou pacientovi zobrazeny údaje, které si o něm lékař v systému eviduje. Patří sem například, přihlašovací údaje, osobní údaje a kontakt. Pacient může jednotlivé údaje libovolně měnit.

2. Karta pacienta

Zde si může pacient prohlížet, jaké operace mu byly vykonány a jaký typ léčby a prostředky lékař při zákroku zvolil.

3. Objednat se

Prostřednictvím záložky objednání si může pacient domluvit s lékařem schůzku. Pacient zde vyplní datum a čas, v který by chtěl přijít na prohlídku a stručně popsat důvod návštěvy. Každé objednání se zobrazí lékaři v nabídce denní program.

4. Kontaktní formulář

Jedná se podobný formulář, který je k dispozici všem návštěvníkům stránek ordinace. Pacient zde může napsat lékaři jakýkoli dotaz nebo problém. Prostřednictvím formuláře vyplní své jméno, emailovou adresu a poté může popsat svůj problém. Jednotlivé zprávy jsou prostřednictvím formuláře odesílány na vytvořený email *j.kaz@centrum.cz*.

8 Diskuse

8.1 Návrh HW zařízení

Pro pohodlné a kvalitní fungování informačního systému a webových stránek zubní ordinace bych doporučil pronájem vzdáleného serveru. V dnešní době existuje mnoho společností, které se zabývají pronájmem a správou serverů, pro využití ke své práci bych doporučil společnosti Master internet nebo VS Hosting, které mimo jiné nabízejí i virtuální servery (VPS), které jsou cenově dostupnou variantou oproti klasickému cloud hostingu.

Při práci v informačním systému nepředpokládám vysoký počet souběžně pracujících uživatelů, proto by pro potřeby zubní ordinace stačila základní varianta **VPS Alpha+** za 298 Kč/měsíc. Tato varianta neobsahuje možnost zálohování, proto bych dále doporučil službu **Managed backup** za 300Kč/ měsíc, která garantuje pravidelné zálohy a správu dat.

Hardwarové vybavení ordinace bych doporučil základní kancelářskou PC sestavu a router pro zajištění připojení k internetu. Cenovou relaci u těchto zařízení neudávám, je to čistě na preferenci uživatelů v jednotlivých ordinacích, kolik jsou ochotni do vybavení své ordinace investovat.

8.2 Ekonomické zhodnocení

Cílem mé práce bylo kromě návržení vlastního informačního systému i jeho ekonomická analýza, ve které budou zahrnuty náklady na vývoj systému a jeho provoz. Mezi nefunkčními požadavky bylo vytvořit systém s co možná nejnižšími náklady a získat tak komparativní výhodu oproti konkurenčním společnostem a jejich produktům. Již ve čtvrté kapitole, IS ve zdravotnictví byli vybrány tři systémy používané ve zdravotnictví, se kterými budu vlastní systém porovnávat. Jedná se o systémy MEDICUS Stomatolog a DENTIST+, které jsou speciálně vytvořené pro stomatologické ordinace a systém AIS – Ambulantní Informační Systém, který nachází uplatnění spíše pro obecné nebo jinak specializované ordinace.

Náklady spojené s vývojem vlastního systému nebudou nijak vysoké, protože, projekt byl vytvořen v rámci studentské práce a budeme tedy počítat pouze čas strávený nad jeho tvorbou, kde budeme uvažovat hodinový tarif pro studenta 100 Kč za hodinu a čistý čas strávený nad tvorbou aplikace zhruba 80 hodin. Náklady na provoz a údržbu budou odvozeny z předchozí kapitoly, kde jsme zjistili ceny, které si účtují externí firmy za pronájem a správu serveru.

V následujících tabulce si porovnáme ceny produktů od již předem zvolených společností vždy v základní variantě, poplatky za instalaci a měsíční paušální příspěvky.

Tab. 4 Tabulka 4 Cenové porovnání konkurenčních produktů

Produkt	Cena licence (Kč)	Instalace (Kč)	Měsíční příspěvky (Kč)
DENTIST+ Základní systém	19 250	2 500	590
Rozšíření o další licenci	9 570	1 000	160
MEDICUS Stomatolog Verze START	6 900	2500	330
Rozšíření o další licenci	3 400	1 000	115
AIS Komfort První licence	8 000	Zdarma	425
Další licence	5 000	Zdarma	134
Vlastní IS Základní verze	8 000	Zdarma	598
Rozšíření o další licence	Zdarma	Zdarma	Zdarma

8.3 Možnosti rozšíření

Současná verze informačního systému je zaměřena na zobrazení požadovaných informací a jejich zpracování. Prozatím je implementována pouze část procesů, které v opravdové ordinaci probíhají. Nabídku operací, které bude systém schopen provádět, bych rozšířil například o možnost vyhodnocení nashromážděných údajů do podrobných přehledů. Lékař tak bude moci evidovat, jaké zákroky se provádějí nejčastěji, jaká věková skupina chodí nejvíce na prohlídky a podobné statistické informace.

Současný návrh aplikace nepodporuje responsivní design, což ovlivňuje zejména pacienty, kteří by chtěli jednotlivé operace v informačním systému provádět prostřednictvím svých tabletů nebo smartphonů. Proto by dalším krokem bylo pro tuto skupinu uživatelů vytvořit mobilní aplikaci, která by tyto nedostatky zobrazení vyřešila.

9 Závěr

V této práci byl vytvořen prototyp informačního systému, který má své využití v zubních ordinacích. Pro tvorbu systému bylo nutné seznámit se současnou situací na trhu s informačními systémy používanými v medicíně, dále bylo nutné definovat požadavky na nový systém a to jak ze strany pacientů, tak lékařů. Na základě výsledků z jednotlivých analýz byl za pomoci UML nástrojů vytvořen návrh řešení a poté jeho implementace. Součástí práce bylo také ekonomické zhodnocení nákladů na tvorbu systému a srovnání s existujícími řešeními.

V teoretické části práce byly popsány základy informační systémů a jejich architektura, jednotlivé pohledy na ní a architektonické vzory. Dále byl popsán jednotlivé modely životního cyklu jejich vývoj a výhody. Nakonec byly představeny a popsány techniky které byly použity během vývoje aplikace.

Aplikace byla v průběhu vývoje konzultována se zubními lékaři z praxe a tak bylo možné upravit jednotlivé funkce přesně podle jejich potřeb. Výsledný prototyp aplikace byl naplněn testovacími daty a vyzkoušen v reálném prostředí zubní ordinace. Na základě tohoto otestování v praxi je možné prohlásit, že definované cíle z úvodní kapitoly byly splněny. Výsledný prototyp je vhodný k použití v malých ordinacích, kde lékaři potřebují evidovat informace o svých pacientech, používaných nástrojích a v případě potřeby vytvářet faktury nebo objednávky pro dodavatele. Z pozice pacientů byli požadavky také splněny, je možné mít přehled o své zdravotní kartě a domlouvat se s lékařem na pravidelných prohlídkách.

10 Literatura

- ARLOW, JIM A ILA NEUSTADT. UML 2 A UNIFIKOVANÝ PROCES VÝVOJE APLIKACÍ: OBJEKTIVĚ ORIENTOVANÁ ANALÝZA A NÁVRH PRAKTICKY. 2., AKTUALIZ. A DOPL. VYD. BRNO: COMPUTER PRESS, 2007. ISBN 978-80-251-1503-9.
- BUCHALCEVOVÁ, ALENA. METODIKY VÝVOJE A ÚDRŽBY INFORMAČNÍCH SYSTÉMŮ: KATEGORIZACE, AGILNÍ METODIKY, VZORY PRO NÁVRH METODIKY. 1. VYD. PRAHA: GRADA, 2005. MANAGEMENT V INFORMAČNÍ SPOLEČNOSTI. ISBN 80-247-1075-7.
- CTMOS. AIS - AMBULANTNÍ INFORMAČNÍ SYSTÉM [ONLINE]. CMS MADE SIMPLE, 2004 [CIT. 2016-10-26]. DOSTUPNÉ Z: <http://www.ctmos.cz/cm/>
- DENTIST +. O PROGRAMU DENTIST + [ONLINE]. COMPUGROUP MEDICAL ČESKÁ REPUBLIKA S.R.O.: SHERWOOD, 2016 [CIT. 2016-10-26]. DOSTUPNÉ Z: [HTTP://WWW.DENTIST.CZ/](http://www.dentist.cz/)
- CHLAPEK, DUŠAN, VÁCLAV ŘEPA A IVA STANOVSKÁ. ANALÝZA A NÁVRH INFORMAČNÍCH SYSTÉMŮ. 1. VYD. PRAHA: OECONOMICA, 2011. ISBN 978-80-245-1782-7.
- IT NETWORK: HTML & CSS. IT NETWORK [ONLINE]. 2016 [CIT. 2016-12-15]. DOSTUPNÉ Z: <http://www.itnetwork.cz>
- IT NETWORK: PHP. IT NETWORK [ONLINE]. 2016 [CIT. 2016-12-15]. DOSTUPNÉ Z: <http://www.itnetwork.cz>
- JAROSLAV ŽÁČEK. JAROSLAV ŽÁČEK - VÝUKA [ONLINE]. OSTRAVSKÁ UNIVERZITA, PŘÍRODOVĚDECKÁ FAKULTA [CIT. 2016-10-13]. DOSTUPNÉ Z: <http://www1.osu.cz/~zacek/>
- KANISOVÁ, HANA A MIROSLAV MÜLLER. UML SROZUMITELNĚ. 2., AKTUALIZ. VYD. BRNO: COMPUTER PRESS, 2006. ISBN 80-251-1083-4.
- KRUTCHEN, PHILIPPE. THE RATIONAL UNIFIED PROCESS: AN INTRODUCTION. BOSTON: ADDISON-WESLEY, c2000. ADDISON-WESLEY OBJECT TECHNOLOGY SERIES. ISBN 0-201-70710-1.
- KRUTCHEN, P. (1995). „THE 4+1 VIEW MODEL OF SOFTWARE ARCHITECTURE“ (MODEL SOFTWARE ARCHITECTURY S POHLEDEM 4+1). IEEE SOFTWARE, 12(6), 42-50.
- MEDICUS. MEDICUS STOMATOLOG [ONLINE]. COMPUGROUP MEDICAL ČESKÁ REPUBLIKA S.R.O.: SHERWOOD, 2016 [CIT. 2016-10-26]. DOSTUPNÉ Z: [HTTP://WWW.MEDICUS.CZ/](http://www.medicus.cz/)
- MINISTERSTVO VNITRA ČESKÉ REPUBLIKY: EGOVERNMENT. EGOVERNMENT: CESTY K EL. ZDRAVOTNICTVÍ [ONLINE]. MVCR, 2016 [CIT. 2016-12-28]. DOSTUPNÉ Z: [HTTP://WWW.MVCR.CZ/CLANEK/EGOVERNMENT-CESTY-K-ELEKTRONICKEMU-ZDRAVOTNICTVI.ASPX](http://www.mvcr.cz/clanek/egoovernment-cesty-k-elektronicemu-zdravotnictvi.aspx)
- PERROW, CHARLES. NORMAL ACCIDENTS: LIVING WITH HIGH-RISK TECHNOLOGIES. PRINCETON, N.J.: PRINCETON UNIVERSITY PRESS, c1999. ISBN 0691004129.
- PFLIEGER, CHARLES P. A SHARI LAWRENCE. PFLIEGER. SECURITY IN COMPUTING. 4TH ED. UPPER SADDLE RIVER, NJ: PRENTICE HALL, c2007. ISBN 9780132390774.

SOMMERVILLE, IAN. SOFTWARE INŽENÝRSTVÍ. 1. VYD. BRNO: COMPUTER PRESS, 2013. ISBN 978-80-251-3826-7.

VOŘÍŠEK JIŘÍ. STRATEGICKÉ ŘÍZENÍ INFORMAČNÍHO SYSTÉMU A SYSTÉMOVÁ INTEGRACE. VYD. 1. PRAHA: MANAGEMENT PRESS, 1997. ISBN 80-85943-40-9.

ZDRAVOTNICTVÍ MEDICÍNA [ONLINE] EHEALTH A TELEMEDICÍNA: NEMOCNIČNÍ INFORMAČNÍ SYSTÉMY. ZDRAVOTNICTVÍ MEDICÍNA [ONLINE]. [CIT. 2015-12-20]. DOSTUPNÉ Z: [HTTP://ZDRAVI.E15.CZ/CLANEK/MLADA-FRONTA-ZDRAVOTNICKE-NOVINY-DN/EHEALTHA-TELEMEDICINA-NEMOCNICNI-INFORMACNI-SYSTEMY-13-DIL-472740](http://zdravi.e15.cz/clanek/mlada-fronta-zdravotnicke-noviny-dn/ehealtha-telemedicina-nemocnicni-informacni-systemy-13-dil-472740).

Přílohy

A Obsah přiloženého CD

Součástí práce je přiložené CD obsahující:

- Vlastní práci v elektronické podobě
- Zdrojové kódy k aplikaci
-

B Ukázky výsledné práce

KONTAKTNÍ FORMULÁŘ

KONTAKTUJTE SVÉHO LÉKAŘE

Vaše jméno

Váš email

Aktuální rok

Jiří Fiala - PEF Mendelu
BP - Analýza a návrh IS zubní ordinace

MUDR. JIŘÍ KAZ ZUBNÍ ORDINACE

PŘIHLÁŠENÍ DO IS

PŘIHLAŠTE KE SVÉMU ÚČTU - ZAMĚSTNANCI

Login:

Heslo:

Jiří Fiala - PEF Mendelu
BP - Analýza a návrh IS zubní ordinace

Objednávka číslo: 001

Dodavatel

Obchodní název: Hartmann, s. r. o.

Adresa: Staroměstská 14, PRAHA 1

IČO: 123456798

Odběratel

Obchodní název: MUDr. Jiří Kaz

Adresa: Hronova 844, 395 01 PACOV

IČO: 66666666

Datum vystavení

01.01.2017

Datum splatnosti

12.01.2017

Položky:

Množství	Název	Cena
15 x	Vrtací hlavice	890,- CZK

Celkem 890,- CZK