

UNIVERZITA PARDUBICE
Fakulta elektrotechniky a informatiky

Moduly pro onkologický systém
Tomáš Pešek

Bakalářská práce
2011

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Tomáš Pešek**
Osobní číslo: **I08135**
Studijní program: **B2646 Informační technologie**
Studijní obor: **Informační technologie**
Název tématu: **Moduly pro onkologický systém**
Zadávající katedra: **Katedra informačních technologií**

Z á s a d y p r o v y p r a c o v á n í :

Teoretická část bakalářské práce bude zaměřena na představení a zhodnocení technologií sloužící k tvorbě webových aplikací s důrazem kladeným na frameworky jazyka PHP. Dále bude proveden úvod do problematiky správy nemocniční agendy, budou přestaveny současné systémy na trhu a bude provedeno jejich kritické zhodnocení.

V aplikační části bakalářské práce budou vytvořeny 3 moduly do stávajícího jádra systému a to: kartotéka pacientů, radiologie a chemoterapie. Modul kartotéka bude základním modulem pro správu pacientů. Modul radioterapie bude zajišťovat plánování pacientů, frakcí, správu ozařečů a běžnou agendu. Modul chemoterapie bude spravovat definici chemo plánů a kontrolu definic u pacientů.

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

1. Gilmore, Jason. **Velká kniha PHP a MySQL 5**. Praha : Zoner Prass, 2007. ISBN: 80-86815-53-6.
2. Schafer, Steven. **HTML, XHTML a CSS**. Praha : Grada, 2009. ISBN: 978-80-247-2850-6.

Vedoucí bakalářské práce:

Ing. Lukáš Čegan, Ph.D.

Katedra informačních technologií

Datum zadání bakalářské práce: **16. prosince 2011**

Termín odevzdání bakalářské práce: **11. května 2012**



prof. Ing. Simeon Karamazov, Dr.
děkan



L.S.



Ing. Lukáš Čegan, Ph.D.
vedoucí katedry

V Pardubicích dne 30. března 2012

Prohlašuji, že jsem tuto práci vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 3. května 2012

Tomáš Pešek

Poděkování

Rád bych poděkoval svému vedoucímu bakalářské práce panu Ing. Lukášovi Čeganovi, Ph.D za vedení této práce, za cenné rady a trpělivost. Dále bych chtěl poděkovat všem, kteří mě podporovali během celého mého studia, zejména mé rodině a přátelům.

Anotace

Tato práce se zabývá problematikou správy nemocniční agendy. Je zaměřena na představení současných nemocničních informačních systémů (s důrazem na použití v onkologických ambulancích) a na představení vývoje nového onkologického IS M2 (nástupce onkologického IS Medoro).

Tato práce se dále zabývá použitými technologiemi, které byly zvoleny pro vývoj. Je to skriptovací jazyk PHP s důrazem na jeho frameworky, databázový systém PostgreSQL a další technologie pro vývoj online aplikací.

Klíčová slova

Onkologický informační systém, nemocniční agenda, Zend framework, PHP, PostgreSQL

Title

Modules for oncological system.

Annotation

This thesis is dealing with issues of management of hospital agenda. It's aimed on presentation of contemporary hospital information systems (with emphasis on utilization in oncological ambulances) and on presentation of development of new oncological IS M2 (successor of oncological IS Medoro).

This thesis is also dealing with used technologies, which were selected for development. It is scripting language PHP with emphasis on its frameworks, database system PostgreSQL and other technologies for development online applications.

Keywords

Oncological information system, hospital agenda, Zend framework, PHP, PostgreSQL

Obsah

1	Úvod a cíl práce	12
1.1	Úvod	12
1.2	Cíl	12
2	Dostupné technologie a systémy	13
2.1	Technologie	13
2.2	Hlavní scriptovací jazyk	13
2.2.1	PHP vs. ASP.NET	13
2.2.2	MVC architektura	14
2.2.3	Frameworky jazyka PHP	15
2.3	Formátování výstupu a skripty na straně klienta	17
2.3.1	XHTML a kaskádové styly	17
2.3.2	SVG	18
2.3.3	Ecmascript, Javascript a AJAX	18
2.4	Databázová vrstva	18
2.4.1	PostgreSQL	18
2.4.2	MySQL	19
2.5	NIS na trhu	19
2.5.1	Produktové řady FONS společnosti STAPRO s.r.o.	19
2.5.2	Medical process assistant společnosti ICZ a.s.	21
2.5.3	Zhodnocení	22
3	Onkologický informační systém M2	23
3.1	Seznámení s OIS M2	23
3.2	Architektura	23
3.2.1	Nová verze	24
3.2.2	Celková koncepce OIS M2	24
3.3	Modul Kartotéka	25
3.3.1	Modelová část	25
3.3.2	Řídící část	27
3.4	Modul Radioterapie	28
3.4.1	Modelová část	28
3.4.2	Řídící část	30
3.5	Modul Chemoterapie	31
3.5.1	Modelová část	31
3.5.2	Řídící část	33
3.6	Instalační příručka	35
3.6.1	HW a SW požadavky na server	35
3.6.2	Postup instalace OIS M2	36

3.6.3	Nutné kroky po konfiguraci	36
3.6.4	Doporučené úpravy po konfiguraci	37
3.7	Uživatelská příručka	37
3.7.1	Příručka kartotéky	38
3.7.2	Příručka radioterapie	38
3.7.3	Příručka chemoterapie	40
4	Závěr	43

PHP	PHP: Hypertext processor
HTML	Hypertext markup language
XHTML	Extensible hypertext markup language
XML	Extensible Markup Language
HTTP	Hypertext Transfer Protocol
SQL	Structured Query Language
MVC	Model - View - Controller
ASP	Active Server Pages
.NET	„dotnet“
JSP	Java server pages
PDF	Portable Document Format
API	Application Programming Interface
SVG	Scalable Vector Graphics
AJAX	Asynchronous JavaScript and XML
HW	Hardware
ZF	Zend framework
MS-DOS	Microsoft - disk operating system
CGI	Common Getway Interface
SW	Software
ER	Entity-relationship
IS	Informační systém
NIS	Nemocniční informační systém
OIS	Onkologický informační systém
HL7	Health Level Seven
PACS	A Photodetector Array Camera and Spectrometer
BMI	Body Mass Index
BSA	Body Surface Area
JVM	Java Virtual Machine
RT	Radioterapie
CHT	Chemoterapie
NRT	Nenádorová radioterapie

Seznam tabulek

3.1	Vývoj PHP 4 a PHP 5.	23
-----	------------------------------	----

Seznam obrázků

2.1	MVC - základní princip.	14
3.1	Grafické znázornění rozdělení modulů.	24
3.2	ER diagram modulu kartotéka.	26
3.3	ER diagram modulu radioterapie.	29
3.4	ER diagram modulu chemoterapie - správa protokolů.	32
3.5	ER diagram modulu chemoterapie - chemoterapeutické pobyty.	34
3.6	Hlavní menu	38
3.7	Procesy základní úrovně	38
3.8	Záložky	38
3.9	Uživatelské procesy v kartotéce.	39
3.10	Uživatelské procesy v radioterapii.	39
3.11	Uživatelské procesy v chemoterapii.	42
3.12	Kalendář aplikací chemoterapie.	42

1 Úvod a cíl práce

1.1 Úvod

Informační systémy (dále jen IS) usnadňují práci milionům lidí denně nejen ve zdravotnictví. Co to ale ve skutečnosti IS je? Podle definice jde o množinu prvků ve vzájemných informačních a procesních vztazích (informační procesy). Množinu prvků si můžeme představit jako soubor lidí a technologických prostředků. Informační a procesní vztahy si můžeme zase představit jako soubor úloh s informacemi, jako například sběr, přenos, zpracování a vyhodnocení. Z definice je patrné to, že když uživatel očekává od IS nějakou práci, musí mu k této práci poskytnout prostředky. Názorně to lze vysvětlit třeba takto: jestliže chceme, aby nám IS vytvořil statistiku toho, jaké byly pacientovi podány léky, IS musí vědět o tom, že pacient P dostal lék L a jestliže chceme tuto statistiku sledovat v čase, IS musí vědět, že pacient P dostal lék L v den D . Jakou formou se o tom IS dozví není podstatné, podstatné je, že bez těchto informací IS statistiku nesestaví.

Důvodů pro nasazení IS může být celá řada. IS za nás neudělá veškerou práci, ale práci, kterou musíme vykonat, nám ulehčí. Kdybychom v dnešní době neměli IS, veškerá data by byla uložena v papírové podobě v centrálních kartotékách, případně v sešitech či diářích jednotlivých zaměstnanců nebo oddělení. Takovéto informace je složité zpětně dohledávat a nejsou (nebo nemohou) být propojeny, takže se ztěžuje i práce s takovými daty. Vyhledávání informací zabere mnoho času a práce s nimi ještě více. Při použití IS máme všechny informace na jednom místě, mezi informacemi existují vazby, takovéto informace máme zálohovány a IS může převzít kus naší práce formou automatických výpočtů, statistik nebo dávkových výkonů.

V nemocniční praxi využijeme možnost snadné zálohy, zamezení vzniku chyb při manuálním přepisu informací, úspora práce a času personálu (automatické výpočty, dávky pro pojišťovny, atd), snadnější a přehlednější přístup k informacím a v neposlední řadě snazší zadávání informací.

1.2 Cíl

Cíl této práce je představení některých nemocniční informačních systému používaných v České republice. Dalším cílem práce je popsat vývoj nových modulů onkologického IS M2 (dále jen OIS M2), zaměřit se na technologie pro vývoj webových aplikací, hlavně scriptovací jazyk PHP s důrazem na jeho frameworky. Cíle práce jsou tedy takové:

1. Kriticky zhodnotit současné NIS na trhu.
2. Popsat technologie použité při vývoji.
3. Popsat vývoj tří stěžejních modulů pro OIS M2.

2 Dostupné technologie a systémy

2.1 Technologie

OIS Medoro, potažmo i nový OIS M2 byl vyvinut (a je dále vyvíjen) na webové platformě. Proč bylo rozhodnuto právě takto? Hlavní výhoda tohoto řešení je v rychlém přístupu k datům pomocí internetového prohlížeče kdekoliv v nemocniční síti (popřípadě ze sítě venkovní). Vyplývá to z definice samotné webové aplikace:

“Webová aplikace je klient/server software, který komunikuje s uživatelem nebo jiným systémem prostřednictvím protokolu HTTP. Jako klient bývá nejčastěji použit webový prohlížeč, jako je například Internet explorer, Firefox, Opera nebo Chrome. ”

Díky všudypřítomným internetovým prohlížečům jsou webové aplikace stále velmi oblíbené a podle všech předpokladů ještě dlouho oblíbené zůstanou.

Dalším argumentem je fakt, že na přelomu roku 2003 a 2004 zažívaly webové aplikace veliký rozmach. V době, kdy NIS disponoval grafickým rozhraním známá z operačního systému MS-DOS a první internetové portály už byly dobře známé a rozšířené ¹, byla vidina práce v takovémto portále velice příjemná. Posoudíme-li fakt, že v této době se PHP 4 těšilo velké oblibě, další argument pro webovou aplikaci je na světě.

Tyto výhody desktopová aplikace nikdy nepřekoná. Sice je pravda, že dnešní doba nabízí velké množství využití desktopových aplikací, ale pro potřeby OIS M2 bohatě postačí (a praxe ukazuje, že postačuje) webová architektura.

2.2 Hlavní scriptovací jazyk

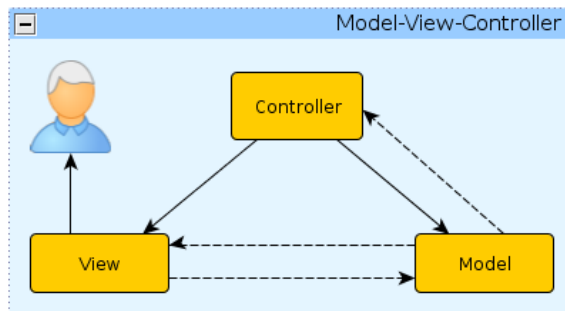
Pro vývoj webových aplikací můžeme dnes sáhnout k několika technologiím. Pomineme-li staříčké CGI, jsou to hlavně technologie PHP, ASP.NET a JSP. PHP a ASP.NET se dnes těší větší oblibě než JSP, proto ve stručnosti v následujícím textu porovnám jen tyto dvě.

2.2.1 PHP vs. ASP.NET

Posuzování výhod jedné technologie a nevýhod druhé je věcí, která bohužel nejde vyřešit v jedné podkapitole této bakalářské práce. Tato problematika by pokryla rozsah samostatné bakalářské nebo diplomové práce. Takováto práce již existuje, ale je z roku 2004, takže dnes je bohužel trochu zastaralá, ale jako zdroj cenných informací slouží dodnes.

Při návrhu nového systému byly kladeny požadavky na osvědčené technologie a postupy. Z předchozích zkušeností bylo známo, že udržovat systém neobjektově a nemodulárně je velice pracné, spíše nereálné. Proto byl požadavek postavit novou aplikaci na robustním základu, čili na frameworku. PHP nabízí frameworků celou řadu (záleží na výběru), ASP.NET je framework sám o sobě. Jak PHP tak ASP.NET dnes nabízejí pohodlnou práci s objektově orientovaným programováním.

¹Seznam.cz byl založen roku 1996 a v roce 1998 spustil e-mailovou službu Seznam mail (*@seznam.cz)



Obrázek 2.1: MVC - základní princip.

Jelikož IS obecně spravují a prezentují velké objemy dat, bylo by vhodné, aby jednotlivé vrstvy byly od sebe odděleny. Toto řešení nabízí softwareová architektura MVC (Model - View - Controller). Většina PHP frameworků tuto architekturu implementuje, ASP.NET v současné době nikoliv. V současné době existuje jen verze 3 Beta frameworku ASP.NET MVC, takže v tomto ASP.NET značně pokulhává za PHP.

Budeme-li vycházet ze závěru Bernardovy bakalářské práce zjistíme, že ve výsledku nemá ani jedna technologie nic navíc oproti druhé. To co se může pro jednoho jevit jako výhoda, může být pro druhého nevýhoda (použití webForms). V roce 2003 bylo rozhodnuto, že započne vývoj na platformě PHP a dnes nemá smysl toto rozhodnutí měnit. Ani ASP.NET, ani JSP nedisponují takovými výhodami, že by kvůli nim stálo za to, měnit osvědčenou platformu.

2.2.2 MVC architektura

MVC architektura rozděluje tři základní části aplikace (datový model, uživatelské rozhraní a řídicí logiku), takže zásah do jedné části bude minimalizovat vliv na ostatní dvě části. Základní princip je znázorněn na obrázku 2.1.

Architektura MVC může být realizována různými způsoby, všeobecně ale platí podobný princip:

1. Aplikace (konkrétně View) čeká na akci uživatele.
2. Controller tuto akci zachytí a rozhodne, co bude dělat dále (upraví Model nebo přímo View).
3. View čeká na další akci a celý proces se opakuje.

Proč je MVC architektura v poslední době tolik oblíbená? V posledních letech zažily informační technologie (a programovací jazyky vůbec) veliký boom. Z těžko dosažitelných funkcionalit se stávají triviality, které zvládne každý, ne každý ji ale zvládne optimálně. Ne že by snad neoptimální aplikace nefungovala, ale je složitá na údržbu. MVC architektura nám pomáhá udržet kód aplikace dobře čitelný a znovupoužitelný. Lépe se v kódu hledají chyby a architektura omezuje duplicity v kódu. V neposlední řadě je jednodušší kód, který je postaven na architektuře MVC, testovat. [1]

Uživatel nijak nezjistí, že je aplikace postavena na architektuře MVC. V drtivé většině totiž ani neví, co to je architektura nebo snad MVC architektura, takže pro něho je přínos nulový, pro vývojáře je to ale cesta, která velice usnadní pozdější správu a rozvoj aplikace.

2.2.3 Frameworky jazyka PHP

Na začátku bychom si měli vyjasnit, co to je framework, k čemu je dobrý, co nám přináší a co po nás chce. Framework je jakási knihovna (software, aplikace), která nám (programátorům) ulehčuje práci a život. Pomocí frameworku zpřehledníme aplikaci, vyvarujeme se duplicitnímu kódu a práci si ulehčíme, protože framework obsahuje nástroje, které bychom jinak museli pracně tvořit. Na oplátku si framework ukousne kus výkonu naší aplikace. U webové platformy jde ale tuto nevýhodu minimalizovat správnou optimalizací webového serveru. Tomu je věnována zvláštní kapitola.

Programovací jazyk PHP disponuje velkým množstvím nejrůznějších frameworků. Opravdový nástup a rozmach PHP frameworků začal s nástupem PHP 5. Do té doby existovaly veřejné depozitáře tříd jako třeba www.phpclasses.com, nebo PEARL. Když si vývojáři všimli toho, jak mocný je framework Ruby on Rails, chopili se příležitosti a začala éra frameworků jazyka PHP. Asi nejznámější framework jazyka PHP je Zend framework (dále jen ZF) od tvůrců samotného PHP. U nás se těší velké oblibě Nette framework od českého autora Davida Grudla. Dalšími známými frameworky jsou například CakePHP, Symfony nebo Yii framework.

Každý známější framework má svého zastávce i svého odpůrce. Je pravdou, že ne každý framework se hodí na všechno. Jaké jsou tedy faktory pro výběr frameworku? Vynechme ekonomické faktory, protože když budeme mít tým pěti programátorů s desetiletou praxí s Ruby on Rails, nebude moc ekonomické všechny přeučit na ZF.

1. Přidaná hodnota frameworku (podle mě nemá smysl použít framework na stránce s jednou fotkou a telefonem).
2. Ulehčení práce (chci použít framework, který mi nebude házet klacky pod nohy).
3. Vystačím si jen s frameworkem (framework by podle mě měl mít nástroje minimálně k odeslání e-mailu, vygenerování PDF a spojit se s databází).
4. Existence použitelné dokumentace a aktivní komunity

Jelikož OIS M2 není webová aplikace, kde se zobrazí pouze jméno, příjmení a telefon, tento fakt přímo lákal k použití některého frameworku. Tím jsme si potvrdili bod jedna. Následně si představíme tři asi nejznámější frameworky jazyka PHP. Všechny frameworky splňují bod dva, tedy implementují architekturu MVC. Další dva faktory si rozebereme u jednotlivých frameworků.

Cake PHP

Vývoj frameworku CakePHP započal v roce 2005 a na konci roku 2005 (konkrétně 20. prosince 2005) [2] byla založena nezisková organizace Cake software foundation, Inc, která má zajišťovat podporu pro rozvoj a vývoj CakePHP (podobné Nette foundation v České republice). CakePHP je jeden z mála frameworků, který podporuje ještě PHP 4. Je to dáno samotnou filosofií tohoto frameworku. Od frameworku se očekává robustní základ, který usnadní správu aplikace a vývoj nové funkcionality tak, aby to bylo vůbec možné a tak, aby to šetřilo čas. CakePHP je přeborník v druhém aspektu. Pomůže nám rychle a celkem efektivně vytvořit nový projekt. Využívá architekturu MVC, takže určitě to je posun kupředu. CakePHP nám ale nedává nic zvláštního navíc. Komponenty pro vytváření PDF dokumentů (včetně čárkových kódů), posílání e-mailů, práci s lokalizací nebo pro vytváření formulářů bychom museli řešit externími knihovnami. Navíc má nepřehledné stránky (zkusme na webu najít licenci) i dokumentaci.

CakePHP je šířen pod licencí MIT ². Samotný fakt, že tato informace je dostupná až ze zdrojových kódů aplikace vypovídá o kvalitě dokumentace.

Nette

Nette framework je z trojice frameworků nejstarší (počítáme-li dobu vývoje), ale podle doby, kdy jej mohou používat i jiní uživatelé než autor, je tento framework nejmladší. Autor se zmiňuje o tom, že vývoj započal roku 2004 a o další čtyři roky později se začal šířit jako open source [3]. Tento fakt se mu stal osudným při výběru frameworku pro OIS M2 a to z toho důvodu, že kvůli svému open sourceovému stáří neměl propracovanou dokumentaci a existovala jen omezená internetová komunita okolo tohoto frameworku. Nette framework je šířen jako svobodný software, to znamená, že si uživatel může vybrat mezi licencí New BSD nebo GPL [4]. Pojdme si říct něco o vlastnostech tohoto frameworku.

Mezi základní vlastnosti tohoto frameworku patří bezpečnost, opravdu povedený ladící nástroj (který může Nette závidět i ZF), velmi dobrý výkon, dnes již celá řada pluginů a rozšíření, aktuálně nejaktivnější komunita a dobrá dokumentace. Nette framework disponuje vlastním šablonovacím systémem ³. Obsahuje velice propracovanou správu formulářů, která umožňuje snadno validovat odeslaná data (na straně serveru i klienta). Jak bylo řečeno disponuje povedeným ladícím systémem. Propracované, obousměrné ⁴ routování.

Nette framework se na první pohled jeví jako ideální volba. Je od českého autora (známého z jeho dřívějších knihoven dibi a Texy!) tudíž by mělo být jednoduší dohledat nejasnosti na fórech nebo v dokumentaci. Bohužel tento fakt platí dnes, kdy Nette disponuje nejaktivnější českou komunitou, ale ještě v roce 2008 byla úroveň dokumentace velice špatná (oproti ZF přímo žalostná) a proto byla dána přednost ZF, který disponuje robustní dokumentací (ať už Reference guide nebo API). Dlužno podotknout, že kdyby se o přepsání OIS Medoro rozhodovalo o pouhý rok déle, Nette framework by nejspíš vyhrál. Další nevýhoda, kterou Nette framework trpěl, bylo to, že existovalo velice omezené množství pluginů (např. přímá podpora generování PDF dokumentů). Jak bylo zmíněno výše, dnes je stav úplně jiný a většina záporných bodů, kterými tento framework trpěl, jsou odstraněny, což z něj dělá velice kvalitní framework, který je schopen nám velice ulehčit a zkvalitnit práci.

Zend Framework

Vývoj ZF započal roku 2005, tedy opět po úspěchu Ruby on Rails. Za vývojem ZF stojí společnost Zend Technologies Ltd., tedy společnost, která zastřešuje celé PHP, jedná se tedy o framework od tvůrců PHP. ZF disponuje rozsáhlou dokumentací, velkým množstvím komponent a celou řadou příkladů na Zend devzone ⁵. Framework je šířen pod licencí New BSD [5].

ZF nám přináší všechno jako Nette framework. Něco lépe, něco hůře. Nejdůležitějšími komponentami, které ulehčují práci, je Zend_Auth a Zend_Acl. Nevýhodu frameworku můžeme minimalizovat vhodným použitím Zend_Cache modulu, přístup do databáze (postgresql, Oracle, MySQL, MSSql) obstarává Zend_Db, s generováním PDF souborů nám pomůže Zend_Pdf (možné propojení s komponentou Zend_Barcode). Zend_View

²<http://www.opensource.org/licenses/mit-license.php>

³Latte - <http://doc.nette.org/cs/templating#toc-latte>

⁴framework vygeneruje URL ze zadané akce presenteru

⁵<http://devzone.zend.com/>

nám umožňuje psát výstupní kódy pomocí prostého PHP, nebo šablonovacího systému SMARTY, který je v Zend_View přímo implementován. ZF disponuje celou řadou komponent, které se hodí více či méně pro určitý projekt. Pro stavbu elektronického obchodu oceníme Zend_Currency nebo Zend_Feed, pro diskusní fórum oceníme modul Zend_Captcha nebo Zend_Mail. Modulů je celá řada a záleží jen na nás, který použijeme.

Jak je patrné z abstraktu této práce, pro vývoj OIS M2 byl nakonec zvolen právě ZF. Největší zásluhu na tom má propracovaná dokumentace. Když porovnáme ZF s jeho největším konkurentem - Nette frameworkem, tak musíme konstatovat, že Nette framework udělal za poslední dva roky opravu mílové kroky a dnes by bylo rozhodování o výběru frameworku složitější. Dost možná by nakonec o výběru nerozhodovaly technické, ale ekonomické a personální parametry.

2.3 Formátování výstupu a scripty na straně klienta

Pojmem formátování výstupu rozumíme procesy a technologie, kterými jsou uživatelům prezentovány data v lidské tj. čitelné podobě.

2.3.1 XHTML a kaskádové styly

XHTML je:

- značkovací jazyk
- jedním z jazyků pro vytváření webových stránek
- zkratkou anglického výrazu eXtensible HyperText Markup Language
- rozšířené HTML tak, aby vyhovovalo XML
- zpětně kompatibilní s HTML

XHTML měl být nástupcem HTML. Verze 1.0 existuje ve třech variantách. Varianta Strict předpokládá, že dokument bude formátován pomocí kaskádových stylů a tak v tomto důsledku zakazuje většinu atributů (width, font, align) a ustupuje i od některých elementů (center, menu). Varianta Translation je vhodná pro vytváření stránek pro starší prohlížeče, protože promíjí zastaralé elementy, jako právě center a menu. Poslední variant Frameset umožňuje používat rámce a stejně jako Translation zastaralé elementy.

Vzhledem k tomu, že dnes nemá moc smysl vyvíjet aplikace pro staré prohlížeče, je jasné, že nejrozšířenější varianta XHTML 1.0 je Strict. Jaké jsou výhody XHTML oproti jeho předchůdci HTML? Důležité je pochopit rozdíl mezi HTML a XHTML. Jestliže se jedná o HTML, stránka by měla být odeslána s MIME type text/html, jestliže se jedná o XHTML, stránka by měla být odeslána s MIME type application/xhtml+xml. Díky tomu internetový prohlížeč pozná, jaký parser má použít a my tak můžeme rozlišit HTML od XHTML. Tato banální věc má však jeden nedostak a to ten, že dominantní prohlížeč na trhu (Internet Explorer) před verzí 9 (vydána v roce 2011) stránku s MIME type application/xhtml+xml nabídne stáhnout, ale nezobrazí jí (což je pro prohlížení trochu nepraktické). Proto se stalo běžnou praxí, že XHTML stránky se posílaly s MIME typem text/html, takže jsme stránku parsovali jako HTML, takže by se dalo říci, že XHTML žádnou výhodu oproti HTML nemá. XHTML mělo být nástupcem HTML, ale jak se ukázalo, HTML vyhrálo. 17. 12. 2010 byla rozpuštěna pracovní skupina, která pracovala na XHTML 2.0 a místo toho pomalu světu představuje HTML5. [6], [7]

2.3.2 SVG

SVG je formát dvourozměrné vektorové grafiky, který lze vytvořit pomocí XML specifikace. Takto vytvořenou grafiku můžeme oživit pomocí Ecmascriptu a kaskádových stylů. V OIS M2 jsou pomocí SVG realizovány grafické výstupy formou 2D grafů. Bohužel nejpoužívanější internetový prohlížeč podporuje XML až od verze 9, takže nižší verze jsou o tuto technologii ošizeny a je nutné grafy převést do klasického obrázkového formátu, který ztrácí výhodu oživení pomocí ecmascriptu a grafická kvalita takového grafu je o trochu nižší.

2.3.3 Ecmascript, Javascript a AJAX

Javascript je nadstavba Ecmascriptu. Tyto technologie provádějí procesy na straně klienta. Porovnejme to s PHP scriptováním. PHP skripty potřebují pro svůj chod webový server s podporou PHP, takže tyto skripty jsou spouštěny na serveru (Klient podá požadavek a server mu odpoví). Mohou přistupovat k databázi a filesystému a jsou spouštěny **před** stažením webové stránky z internetu. Naproti tomu skripty na straně klienta se zpravidla (existují výjimky) spouští až **po** stažení webové stránky z internetu, tedy ke svému chodu nepotřebují webový server a vykonávají se v internetovém prohlížeči. Ecmascript je použit ve formátu SVG, javascript pak v XHTML.

AJAX⁶ je souhrn technologií, které mění část obsahu webové stránky bez nutnosti znovunačtení celé stránky. Z významu slova AJAX zjistíme, že to jsou technologie javascript, xml a XMLHttpRequest. AJAX funguje asi tak, že javascript zachytí událost (např. změna hodnoty v combo boxu), na pozadí pošle HTTP požadavek, na pozadí přijme HTTP odpověď (nejčastěji ve formě XML), zpracuje odpověď a metodami javascriptu změní část stránky.

2.4 Databázová vrstva

Výběr databázové vrstvy byl ve značné míře ovlivněn politikou podniku. Mezi nejznámější komerční databázové servery patří Oracle, Sybase, nebo MSSQL. Mezi open sourcové databázové servery pak PostgreSQL, MySQL a FirebirdSQL. Jelikož stará verze onkologického IS využívala PostgreSQL a protože se tato databáze velice osvědčila, nová verze systému je nativně vyvíjena také na PostgreSQL. Nový systém je ale postaven na architektuře MVC, to znamená, že lze databázový server změnit bez žádných větších komplikací.

Podniková politika upřednostňuje open sourcové řešení (PHP, apache, unix, ...) a proto se výběr nativní databáze zaměřil právě na open sourcové řešení. V první řadě FirebirdSQL si moc nerozumí s unixovým serverem, proto z dalšího porovnávání vypadává.

2.4.1 PostgreSQL

Dlouhou dobu platilo, že databázový systém PostgreSQL je open sourcová varianta pro Oracle. Vysloužil si to hlavně díky svým schopnostem. Již od verze 6⁷ umožňuje tvorbu triggerů a procedur a nejen v jazyce PL/SQL, ale i v C nebo Pearl (mysql umožňuje tvorbu triggerů až od verze 5⁸). PostgreSQL disponuje možností tvorby poddotazů, vytváření

⁶Asynchronous javascript and XML

⁷rok 1997

⁸rok 2005

cizích klíčů a v neposlední řadě i “atomic operations”, což jsou vlastně transakce (buď projde celá posloupnost dotazů, nebo žádný). Co se týče rychlosti základních příkazů SELECT, INSERT a UPDATE, tak PostgreSQL dosahuje horších výsledků než MySQL, disponuje ale daleko větší škálou možností, že se tento fakt stává nepodstatným, protože rozdíly v rychlosti nejsou nijak závratné. [8]

2.4.2 MySQL

MySQL je dnes velice oblíbený databázový systém. Díky tomu, že je součástí základního softwareového vybavení webserverů (LAMP⁹) je nainstalovaný na drtivé většině webhostingů. Vývoj MySQL byl od začátku soustředěn na rychlost. Tento fakt vedl k tomu, že databázový server obsahuje určitá zjednodušení, jako například jen jednoduché způsoby zálohování [9] (pro ukládání citlivých dat podstatný nedostatek). Stejně jako Nette framework udělal v posledních letech databázový server MySQL velký krok kupředu. Od verze 5 umožňuje ukládat procedury, vytvářet trigery a pohledy. Zvláštností MySQL je způsob, kterým supluje chybějící sekvence. Sloupci, který má obsahovat automaticky generovanou posloupnost čísel, je možné přiřadit příznak Auto Increment a tato vlastnost se postará o to, že ve sloupci bude uložena vždy unikátní hodnota. Ve většině známých databázových systémů se tohoto dosahuje právě pomocí sekvencí.

2.5 NIS na trhu

2.5.1 Produktové řady FONS společnosti STAPRO s.r.o.

“STAPRO s.r.o. je významným dodavatelem informačních systémů, diagnostických přístrojů, zdravotnické techniky a zároveň i poskytovatelem služeb v oblasti informačních technologií pro zdravotnictví.”

Produktová řada FONS uceluje portfolio společnosti. Rozsah a obsah produktů společnosti vyžadoval nové uspořádání, které reflektuje požadavky na novou, kompaktnější strukturu produktů. V současné době produktová řada FONS obsahuje tyto systémy:

1. FONS Enterprise - NIS
2. FONS AKORD - NIS
3. FONS Openlins - laboratorní komplement
4. FONS Report - manažerský informační systém
5. FONS Flexi - flexibilní formuláře

Společnost STAPRO donedávna disponovala a nabízela další NIS. Byly to StaproMedea, StaproGF (dříve Grey Fox) a StaproH (dříve HiComp). U instalací těchto systémů je předpoklad postupného upgrade na nové NIS: FONS Akord a FONS Enterprise.

Vzhledem k tématu této práce si podrobněji rozebereme první dva systémy, FONS Enterprise a FONS Akord.

⁹Linux - Apache - MySQL - PHP

FONS Akord

FONS Akord je inovovaná verze staršího systému StaproAkord. FONS Akord umožňuje vedení zdravotnické dokumentace a některé provozní činnosti některých klinických pracovišť pomocí různě propojených modulů. Podle společnosti STAPRO má FONS Akord tyto přínosy a výhody:

1. Zvýšená efektivita práce a standartizace procesů a postupů (přednastavené obrazovky, formuláře a work-flow dle best-practices)
2. Otevřenost systému a vysoká flexibilita umožňuje individuální nastavení odpovídajícím reálným procesům probíhající na daném pracovišti a požadavkům akreditačních standardů
3. Podpora systému při rozhodování, poskytnutí správných informací ve správný čas
4. Omezení chybovosti a zvýšení kvality vykazování pojišťovnam
5. Statistické výstupy
6. Bezpečnost

FONS Akord obsahuje celou řadu modulů, níže jsou uvedené nejzajímavější z nich.

- **CERTIFIKACE** - FONS Akord je pravidelně certifikován na nejnovější operační systémy. Certifikace na WIN 7 je potvrzením korektního fungování systému na nejnovějším operačním systému.
- **ACTIVE DIRECTORY** - Pokud zdravotnické zařízení využívá pro řízení uživatelských účtů službu Active directory, lze ji s FONS Akord propojit a docílit tak snazší správy uživatelů a jejich práv.
- **KOMPLEXNÍ POPIS OPERACE** - operace lze plánovat a následně zapsat do operačních protokolů, které popisují operaci, včetně údajů o operačním týmu, o provedených výkonech, časech, materiálech a použitých přístrojích. Pohled na protokol je ucelený - v jednom formuláři vidíme všechny potřebné údaje včetně vyúčtování.
- **PORODNICE** - umožňuje vést potřebnou dokumentaci k vyšetření a hospitalizaci těhotné ženy, informace o porodu a stav novorozence a matky po porodu.
- **PLÁNOVÁNÍ PROCEDUR** - všechny postupné kroky jsou propojeny - od ordinace procedur lékařem, přes naplánování procedur, záznamy fyzioterapeutů a závěrečný stav pacienta po proceduře.
- **LÉKOVÉ INTERAKCE** - systém ukazuje aktuální údaje o preskripci, upozorňuje na alergie, hlásí lékové interakce a poskytuje informace o alternativních lécích stejné ATC skupiny.
- **KOMPLEXNÍ OBJEDNÁVKOVÝ SYSTÉM** - umožňuje lehce a přehledně objednat pacienta na určité vyšetření.
- **ALERTY** - umožňují online informovat uživatele o různých akcích (požadavek na konzilium a podobně).
- **EVIDENCE PODÁNÍ LÉKŮ** - slouží k online evidenci podaných léků u lůžka pacienta, automaticky zajišťuje vyskladnění léku a vykáže plátci ZULP.

Zdroj: Materiály z konference Inmed, konané v listopadu 2011 v Pardubicích.

FONS Enterprise

“Nemocniční IS FONS Enterprise je komplexní IS pro podporu provozu zdravotnických zařízení různého typu a velikosti. Jde o novou generaci IS, který vznikl na základě měnícího se prostředí zdravotní informatiky.”

Podle společnosti STAPRO má FONS Enterprise tyto přínosy pro uživatele:

1. Rozšíření spektra metrik dle údajů v IS pro hodnocení činnosti zdravotnického zařízení
2. Plánování, sledování a vyhodnocování procesů
3. Rozdělení procesů mezi definované uživatele nebo role - úkolování, TODO list
4. Snížení nákladů na léčbu použitím expertních informací v průběhu léčby
5. Snížení administrativní zátěže uživatelů při práci se systémem
6. Maximální vytíženost evidovaných dat ve formě sestav, statistik a reportů
7. Snížení chybovosti evidovaných dat
8. Efektivní podpora akreditací
9. Legální vedení zdravotnické dokumentace čistě v elektronické podobě
10. Individuální nastavení uživatelského rozhraní
11. Podpora vyšší právní ochrany zdravotnického zařízení díky nástrojům pro řízení kvality péče
12. Plná otevřenost pro komunikaci s jinými systémy
13. Pozitivní přístup uživatelům k aplikaci

FONS Enterprise je rozsáhlý systém, který nabízí celou řadu funkcionalit, níže jsou uvedeny opět nejzajímavější z nich:

- **PODPORA KLINICKÝCH PROCESŮ** - systém obsahově pokrývá všechny klinické procesy řešené ve stávajících NIS.
- **STRUKTUROVANÁ DOKUMENTACE** - je možné nadefinovat konkrétní podobu jakékoliv strukturované dokumentace pro specifické potřeby jednotlivých oddělení.
- **VYSOKÝ STUPEŇ KONFIGUROVATELNOSTI** - umožňuje vyspecifikovat výslednou podobu klinické události přesně dle potřeb jednotlivých pracovišť.
- **DASHBOARD** - díky modernímu pojetí grafické části aplikace umožňuje FONS Enterprise práci s více okny na jedné obrazovce (vhodné pro širokoúhlé monitory).
- **PACIENTSKÝ PANEL** - obsahuje základní údaje o pacientovi, včetně poslední alergie, diagnózy, medikace a jiné informace o pacientovi. Uživatel je na první dojem informován o důležitých faktorech souvisejících s léčbou.
- **TIMELINE** - umožňuje graficky interpretovat výsledky nebo jinak strukturované údaje či události.

Zdroj: Materiály z konference Inmed, konané v listopadu 2011 v Pardubicích.

2.5.2 Medical process assistant společnosti ICZ a.s.

Společnost ICZ a.s. nabízí v rámci regionu střední Evropy integrovaná softwareová a síťová řešení napříč různými obory. Specializuje se tedy, narozdíl od Stapura, nejen na zdravotnictví. Mezi její produkty patří například Regionální informační systém (Geoaplikace roku 2003), systém REP pro Českou poštu, s.p. (IT projekt roku 2003) nebo ePACS (sdílení

obrázkových dat napříč celou republikou).

Medical process assistant (dále jen MPA) je nová generace klinického IS, která na území ČR podporuje všechny typy klinických pracovišť. Podle společnosti se jedná o přední evropské řešení provozované na desítkách pracovištích po střední Evropě, tím pádem je samozřejmostí plná jazyková, legislativní a procesní lokalizace.

Součástí MPA je nativní podpora procesního workflow, které automatizuje řadu procesů, tím pádem plní základní funkci IS - usnadňuje a urychluje práci. Další součástí MPA jsou specifické moduly pro jednotky intenzivní péče, neonatologii a porodnictví, chirurgické obory (včetně elektronických operačních protokolů) a podpora diet včetně komunikace se stravovacím provozem.

Podle společnosti MPA obsahuje tyto unikátní vlastnosti řešení:

1. Implementace procesního přístupu do libovolné hloubky specifikace na úrovni dokumentace, formulářů, uživatelů, činností, úkolů, termínů a informací.
2. Podpora klinického rozhodování a implementace Clinical Pathways¹⁰.
3. Procesní workflow je nástrojem a pomocníkem k certifikaci a akreditaci.
4. Systém je plně konfigurovatelný ve všech potřebných oblastech.
5. Využívá plnohodnotnou integraci prostředí MS Office.
6. Systém plně respektuje všechna národní specifika a normy ČR a veškeré požadavky v Evropské unii.
7. Systém je vytvořen na bázi nejmodernějších technologických poznatků pro vývoj a je vyvíjen na platformě Java.
8. Plně podporuje integrační standardy a technologické požadavky, jako jsou HL7¹¹, SOA¹², Web services.

2.5.3 Zhodnocení

Společnost STAPRO je přední dodavatel IS pro zdravotnické zařízení u nás. S produkty této společnosti se může seznámit budoucí zdravotnický personál již na středních zdravotnických školách. Přes dvacet let zkušeností, dotace z evropských regionálních fondů¹³ a fakt, že společnost dodává svá řešení po celé republice (i na Slovensku), ze společnosti dělá opravdu silného hráče na poli nemocničních IS. Další výhodou této společnosti je fakt, že se nespecializuje pouze na IS, ale dodává i diagnostické přístroje, zdravotnickou techniku a poskytuje další služby pro zdravotnictví. Podle referencí společnosti jsou nemocniční IS produktové řady FONST instalovány u desítek zákazníků, z nichž největší uživatelé jsou Krajská zdravotní a.s. či Krajská nemocnice Tomáše Bati a.s.

Naproti tomu se společnost ICZ specializuje na širší oblast trhu čili nejen na zdravotnictví. Do podvědomí zdravotnického oboru se dostala hlavně díky projektu ePACS, který spojuje více než sto nemocnic napříč Českou republikou se správou ve VFN v Praze. Podle referencí společnosti je MPA implementován v Oblastní nemocnici Jičín Fakultní nemocnice Brno, což je značný nepoměr v počtu implementacích mezi ICZ a Staprem.

¹⁰Mapy péče

¹¹health Level 7

¹²Service-oriented architecture

¹³NOVÁ GENERACE NIS (INOVACE PRODUKTU, INOVACE PROCESU)

3 Onkologický informační systém M2

3.1 Seznámení s OIS M2

Onkologický informační systém M2 (dále jen OIS M2) navazuje na svého předchůdce OIS Medoro, který byl vyvinut v letech 2003 až 2004 firmou Medoro s.r.o. Než se začneme seznamovat s novým systémem, řekněme si, co vůbec vedlo k vývoji starého OIS Medoro.

1. Dokumentace radioterapie a chemoterapie na různých místech
2. Komplikované předávání této dokumentace
3. Důsledkem mnoha informačních zdrojů se dokumentace stává nepřehlednou
4. Pro dokumentaci průběhu léčby je používán příliš obecný NIS nebo jen papír

Zde je nutno podotknout, že cílem OIS není nahradit NIS, ale jen ho vhodně rozšířit. Proto byly ve starém OIS Medoro použity technologie pro komunikaci s NIS a do nového systému byly převedeny také.

Jak se začal starý OIS Medoro nasazovat a používat, vznikaly postupně nové požadavky na jeho funkčnost. V roce 2004 nebyly logicky dostupné dnešní technologie a možnosti a tak se nové funkcionality přidávaly mnohdy krkolomným způsobem, který znamenal odchylky od standardů programování. Typickým způsobem byla snaha vytvořit “něco jako AJAX” na úrovni tehdejšího javascriptu a CCS 1.

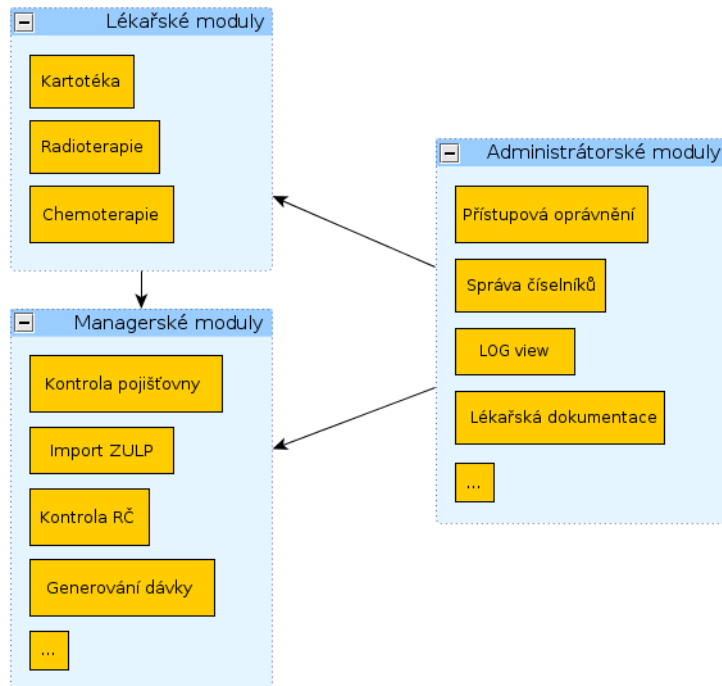
3.2 Architektura

Hlavní důvod pro vývoj nové verze byl ten, že starý OIS Medoro byl vyvinut na technologii PHP 4. V době vývoje sice přišlo na scénu nové PHP 5 s novým přístupem k objektově orientovanému programování (OOP), ale to už vývoj započal a nové PHP 5 přinášelo více otázek než odpovědí. V tabulce 3.1 se podíváme na průběh vydání verzí PHP řady 4 a 5 v období vývoje OIS Medoro.

Takže bylo rozhodnuto, že rozpracovaný projekt nebude přepsán do nového, neznámého PHP 5.0, ale zůstane se u ustáleného a spolehlivého PHP 4. V dnešní době máme k dispozici PHP 5.3 (nejnovější verze PHP 5.3.8, srpen 2011). Spravovat rozsáhlou aplikaci ve staré verzi reálně bylo, ale vyvíjet nové funkcionality už bylo složité. Proto bylo rozhodnuto, že nové moduly OIS budou vyvíjeny na nových technologiích a staré moduly se postupně přepíše.

PHP 4	PHP 5
4.3 - prosinec 2002	5.0 - červenec 2004
4.4 - srpen 2005	5.1 - listopad 2005
4.4.8 - leden 2008	5.2 - listopad 2006
4.4.9 - srpen 2008	5.2.6 - květen 2008
konec řady 4	5.2.8 - prosinec 2008

Tabulka 3.1: Vývoj PHP 4 a PHP 5.



Obrázek 3.1: Grafické znázornění rozdělení modulů.

3.2.1 Nová verze

Nová verze přináší celou řadu vylepšení. Uživatelé ocení implementaci AJAXu a zavedení UI¹ element z knihovny jQuery, která zpříjemňuje práci s celou stránkou ať už je to pomocí widgetů² jako je například UI tab (odpadá znovunačítání stránky), příjemný kalendář nebo jen zpříjemnění grafiky pomocí jQuery efektů. Uživatelé dále ocení jednotný export tiskových sestav do PDF, přímá komunikace s PACS serverem nebo generované grafy pomocí SVG.

To, co uživatel nemůže ocenit, je jádro systému. OIS M2 je postaven na ZF, čili je postaven na architektuře MVC, na principech OOP a na modulární struktuře ZF. OIS M2 je testován unit testy (jeden z kroků pro certifikaci programu), využívá cacheování, vytváří logy, má pokročilejší správu oprávnění, pomocí standartu HL7 může komunikovat se zdravotnickými přístroji jako je například CT nebo rozšiřuje možnosti nemocniční agendy.

3.2.2 Celková koncepce OIS M2

Jak bylo řečeno výše, OIS M2 využívá modulovou strukturu ZF, takže základní dělení jednotlivých funkcionalit je modulové. Z cílů práce je patrné, že stěžejní moduly jsou Kartotéka, Radioterapie a Chemoterapie. V těchto modulech jsou základní funkce pro zpracování lékařských dat o jednotlivých pacientech, zařadíme tyto moduly do skupiny modulů lékařských. Dalšími skupinami modulů jsou managerské a administrátorské. Poslední dvě skupiny obsahují více modulů než skupina lékařská, ale jejich funkcionalita není tak složitá jako moduly lékařské, nehledě na to, že z programátorského hlediska na nich není nic zajímavého. Managerské moduly obhospodařují ekonomické funkce jako je generování sou-

¹User Interface

²"Udělatko", miniaplikace z knihovny jQuery UI

borů dávek pro pojišťovny nebo sledování statistik. Moduly administrátorské zase spravují kompletně celý systém. Starají se o různé číselníky, zobrazování logů a stavu systému nebo spravují uživatelská oprávnění. Na obrázku 3.1 můžeme vidět grafické uspořádání vybraných modulů rozdělených do skupin.

V následujících kapitolách si podrobně rozebereme vybrané tři lékařské moduly.

3.3 Modul Kartotéka

Modul Kartotéka je základní modul pro práci s daty o jednotlivých pacientech. Nelze provádět žádné výkony (ozařování, kontrola, aplikace léčiv) u pacientů, kteří nejsou uvedeni v kartotéce. Tento princip vychází z nemocniční praxe a je zcela logický.

Co do funkcionality tento modul nabízí základní operace jako vytvořit a editovat pacienty. Kromě této jednoduché zprávy pacientů navíc nabízí možnost k pacientům přikládat různé druhy dokumentace. Nejjednodušší je informace o změně váhy (s tím spojená změna BMI a BSA) a tlaku, libovolné množství různých souborů (kategorizace obrázky, dokumenty, laboratorní výsledky, ...), libovolné množství anamnéz (specifikovaný formulář) a v neposlední řadě libovolné množství zpráv (taktéž specifikovaný formulář).

Specifické formuláře anamnéz a zpráv jsou prvními faktory, které začínají OIS M2 oddělovat od obecných NIS. To, co různé NIS řeší textovým polem obdobné MS Word, je v OIS vysoce specifikováno a proto tyto data přinášejí širší možnost zpracování. Nejjednodušší příklad této výhody je vyfiltrování pacientů s určitou diagnózou (to ale umí i většina NIS). To, co ale NIS neumožňují je filtrace podle specifických onkologických kritérií (např TNM³ specifikace s rozšířeními).

Veškeré tiskové výstupy nejen v modulu Kartotéka probíhají tak, že data jsou exportována do PDF a potažmo vytisknuta. Tímto způsobem se vyhneme různým nástrahám tisku webové stránky se specifickým souborem kaskádových stylů. Mimo tisků uložených zpráv a anamnéz tento modul nabízí tisk strvrzenky o úhradě regulačního poplatku, žádost o příspěvek na úhradu tohoto poplatku, štítky se jménem a diagnózou, žádanky na vyšetření CT, MR či PET a různé jiné žádanky.

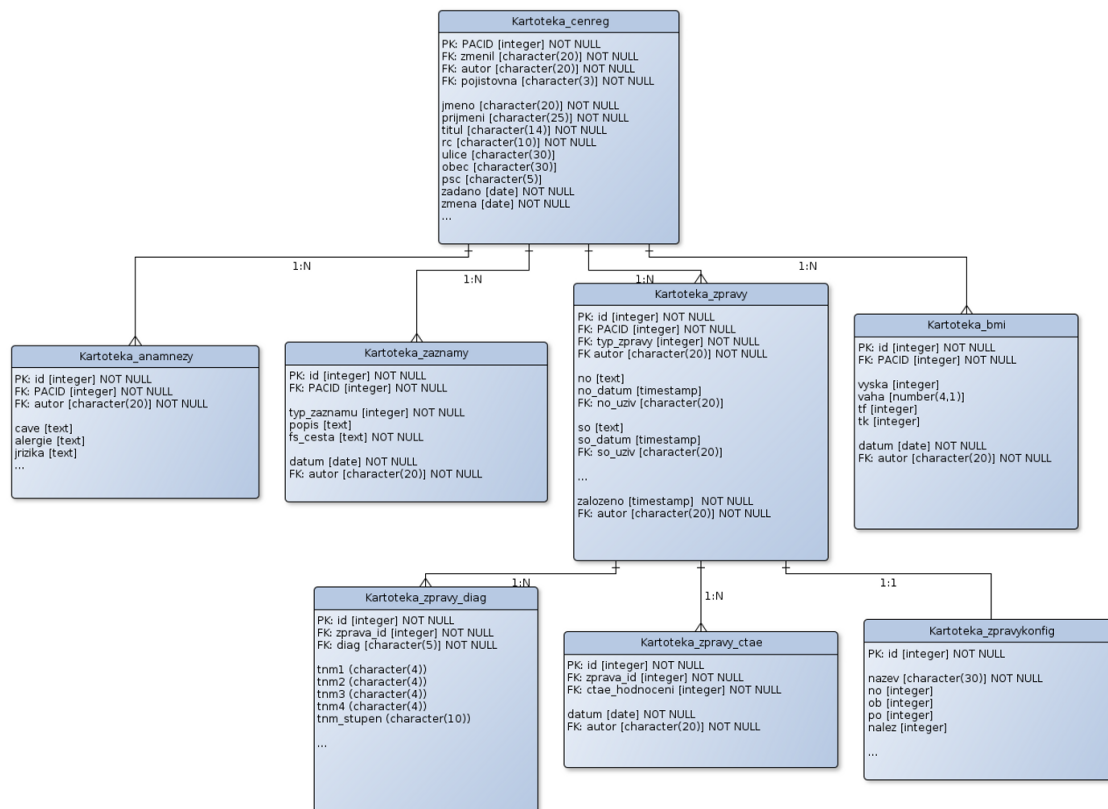
3.3.1 Modelová část

Nyní se podíváme na modelovou stránku tohoto modulu. ER diagram si můžeme prohlédnout na obrázku 3.2. V ER diagramu nejsou u tabulek vypsané všechny atributy, tento krok je kvůli přehlednosti diagramu. Například tabulky `Kartoteka_anamnezy` a `Kartoteka_zpravy` mají přes sedmdesát atributů a z hlediska pochopení struktury jsou vynechané atributy nezájímavé. Jinak řečeno, zajímavé atributy jsou v ER diagramu znázorněny, nezájímavé jsou naznačeny a doplněny třemi tečkami. Na ER diagramu vidíme, že modul je po modelové stránce jednoduchý, skládá se z osmi tabulek spojenými mezi sebou vazbou 1:N (až na jednu výjimku). Tento návrh splňuje 3 NF, čili každý atribut obsahuje atomickou hodnotu, každý neklíčový atribut je závislý na primárním klíči (PK) a všechny neklíčové atributy jsou vzájemně nezávislé. Jelikož se ve všech tabulkách primární klíč skládá jen z jednoho atributu, dalo by se říci, že tento model splňuje BCNF, ale při návrhu s touto formou nebylo počítáno, takže se jedná o jistý bonus.

Pojďme si nyní probrat jednotlivé tabulky a jejich význam v modulu.

Kartoteka_cenreg Jedná se o tabulku se základními daty o pacientech. Za zmínku zde stojí primární klíč PACID. Zde bych si dovolil obhájit unikátní identifikátor z číselné řady

³Tumor - Nodus - Metastáze



Obrázek 3.2: ER diagram modulu kartotéka.

oproti rodnému číslu. V první řadě se mnozí lidé domnívají, že rodné číslo je samo o sobě unikátní identifikátor osob narozených na území České republiky (popřípadě Slovenské republiky). V rodných číslech ale existují duplicity.

Přibližme si princip rodných čísel. Prvních šest číslic nese informaci o datu narození, kde u žen se k měsíci přičte padesát. Zbytek rodného čísla (tři až čtyři místný) odděluje dvě osoby narozené ve stejný den, zpravidla se tyto číslice oddělují lomítkem. Od roku 2004⁴ zákon umožňuje přiřadit alternativní číslo, jestliže jsou v jeden den vyčerpána rodná čísla. Alternativní číslo má formát takový, že se k měsíci přičte u mužů číslo dvacet a u žen sedmdesát.

V jiných zemích jsou použity například číselné řady, čísla sociálního pojištění, fiskální číslo (číslo daňové evidence) nebo jiná kombinace, která může obsahovat datum narození (například finská rodná čísla obsahují datum narození).

V použití v informatice má ale rodné číslo jisté vady. Jedna vada může být ta, že v rodných číslech existují chyby. Existuje nejméně jeden případ toho, že dvěma občanům bylo přiděleno stejné rodné číslo. Další vada je ta, že s rodným číslem je vhodné zacházet jako s řetězcem (character) a ne jako s celým číslem (integer popřípadě number), takže 4B (integer) proti 11B (character, 10 znaků + informace o velikosti). Třetí, poslední a nejvýraznější vada je ta, že kdyby rodné číslo figurovalo jako primární klíč, přibude nám jedna velká starost se situací, kdy uživatel-lékař zadá pacientovy špatné rodné číslo (i přes algoritmus ověřující správnost rodného čísla, zkrátka osoba X bude mít přiděleno rodné číslo osoby Y, nebo rodné číslo, které nebylo vydáno).

Vypadá to jako banalita, ale zkusme změnit primární klíč tabulky, na kterou odkazuje

⁴citovat <http://aplikace.mvcr.cz/archiv2008/sbirka/2004/sb098-04.pdf>

60% všech tabulek. Někdo si řekne vždyť stačí napsat pouze `UPDATE Kartoteka_cenreg SET rc = nove_rc WHERE rc = stare_rc`; jenže v databázi jsou nastavené relace, takže takhle jednoduše to bohužel nejde. Musel by se vytvořit nový pacient s novým (možná správným) rodným číslem, musely by se zeditovat všechny tabulky, které mají vazbu s tabulkou `Kartoteka_cenreg` a nakonec smazat pacienta se špatným rodným číslem.

Proto bylo rozhodnuto, že rodné číslo nebude primárním klíčem tabulky `Kartoteka_cenreg`.

Kartoteka_anamnezy Tato tabulka obsahuje data anamnéz. Tabulka obsahuje 74 různých entit představujících žádoucí informace jako například demografie, rodinná anamnéza, toxikologická anamnéza, sociální anamnéza nebo informace i předchozí aplikaci chemoterapie.

Kartoteka_zpravy Tato tabulka obsahuje data zpráv. Tabulka obsahuje 79 různých entit. Od tabulky `Kartoteka_anamnezy` se liší v tom, že jednotlivé úseky zpráv může zadávat různý lékař a je třeba uchovávat informaci o tom, kdo a kdy zprávy zapsal. Takže zatímco tabulka `Kartoteka_anamnezy` si vystačila s jednou relací na tabulku uživatelů, tabulka `Kartoteka_zpravy` uchovává relací více.

Kartoteka_zpravy_diag Tato tabulka obsahuje informace o diagnózách. Každá zpráva může obsahovat kolonku **primárního onemocnění** a kolonky vedlejšího onemocnění. Primární onemocnění je jen jedno, zato vedlejších onemocnění může být více a informace o primárním a vedlejších onemocněních jsou totožná. Proto vazba 1:N. V této tabulce mohou být uloženy specifikace TNM a dalších onkologických dat.

Kartoteka_zpravy_ctae Tato tabulka obsahuje informace hodnocení toxicity. U jedné zprávy může být libovolný počet hodnocení, proto vazba 1:N.

Kartoteka_zpravykonfig Tato tabulka je editována z administrátorských modulů a definuje typy zpráv. V současné době je v systému nakonfigurováno 12 druhů zpráv od telefonické konzultace přes ambulanci zprávu ke konziliu.

Kartoteka_bmi V této tabulce jsou uloženy hodnoty BMI dat (váha a váška) plus hodnoty tlaku. Výstupy z této tabulky lze sledovat v grafickém výstupu (knihovna generující SVG grafy je popsána v kategorii Použité technologie).

Kartoteka_zaznamy Tabulka obsahuje informace o zdravotnické dokumentaci v různých formátech (např. pdf, txt, doc, jpg, png, ...). Zdravotnické dokumenty jsou kategorizovány. Momentálně jsou vytvořeny čtyři kategorie (obrázková dokumentace, psaná dokumentace, ostatní dokumentace a laboratorní výsledky).

3.3.2 Řídící část

Modul obsahuje celkem tři controllery:

1. AjaxController
2. IndexController
3. PacientController

Veškeré AJAXové operace v celém systému M2 jsou směřovány do speciálního controlleru vždy se stejným jménem - AjaxController, takže jestliže se v kterémkoliv modulu nachází **AjaxController**, tento modul využívá AJAX. Nejprve nás asi zděsí třicet šest různých metod. Toto číslo je opravdu velké (nejvíc ajaxových metod v celém M2). Tento počet je dán tím, že v tomto modulu jsou implementovány zprávy a tisky žádank. Tyto dvě operace jsou kvůli úspoře času řešeny kompletně celé AJAXově. Když vezmeme v potaz, že máme sedmnáct různých polí pro zprávu, z toho osm jich je mezi sebou

tak odlišných, že vyžadují svou metodu a u každého pole máme možnost načíst data z předchozí zprávy, máme to 16 metod jen pro obhospodařování zpráv.

IndexController má na starosti základní úkoly se všemi pacienty. Je to hlavní výpis pacientů, filtrování pacientů, řazení pacientů, nastavení počtu zobrazených pacientů a listování ve výpisu.

PacientController má na starosti veškeré funkcionality ohledně informací o pacientovi v tomto modulu. Stará se o ukládání, editaci a mazání dat z databáze (popřípadě z file systému) (anamnézy, záznamy, zprávy). Umožňuje tisk žádanek a štítků a zobrazuje grafický výstup z hodnot BMI.

3.4 Modul Radioterapie

V tomto modulu jsou soustředěny informace o radioterapeutické léčbě. Dále je tento modul členěn na nádorovou a nenádorovou radioterapii. V tomto modulu je možné nadefinovat radioterapeutický plán ozařování, který nese veškeré informace o plánované léčbě. Nejdůležitějšími informacemi je celková dávka radiace, která má být cílená na postiženou tkáň a informace o tom, do kolika frakcí má být tato dávka rozdělená. Na základě těchto údajů poskytne OIS M2 seznam aktuálních pacientů, kteří v daný den čekají na ozáření. OIS M2 se po potvrzení aktuálního ozáření postará o zaznamenání patřičných výkonů, ze kterých se generují dávky pro pojišťovny. V části potvrzení aktuálního odzáření nechybí možnost zaškrtnout případně portál s lékařem. V tomto případě se na dalším místě OIS M2 zobrazí seznam pacientů, které tento portál čeká.

Při importu plánu pro nádorovou radioterapii systém na základě dat z DVH⁵ systém sestaví a nabídne uživateli graf, ve kterém je možné pozorovat vývoj dávky ve sledovaných orgánech.

U nenádorové radioterapeutické léčby dochází k ozařování cesiovým ozařovačem. OIS M2 poskytuje u aktuální frakce nenádorové radioterapie údaj o tom, jak dlouho se má inkrimované místo v daný den ozařovat. Výpočet vychází z rovnice rozpadu cesia a pro personál tak odpadá nutnost tento údaj dohledávat v tabulce, která se musí pravidelně aktualizovat.

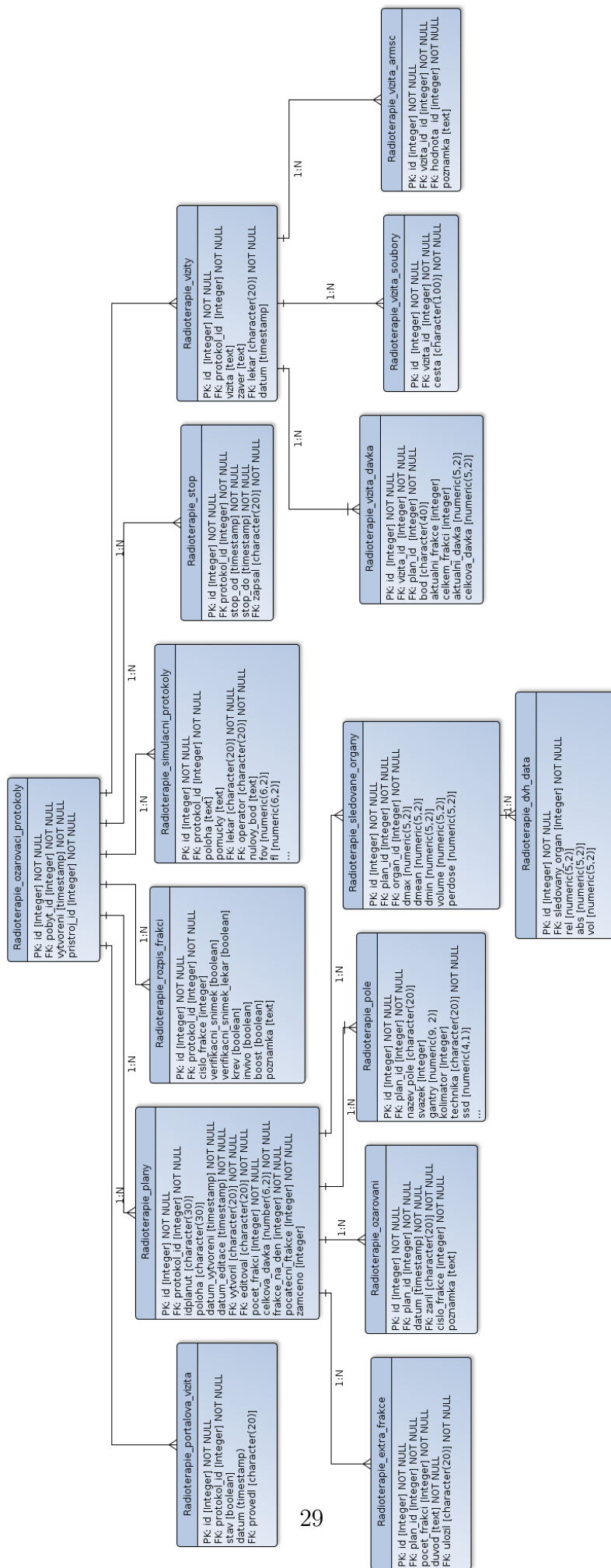
V tomto modulu je dále možné vytvářet vizity k léčbě a evidovat simulační protokoly léčby. Všechny tyto procesy jsou opět navázány na pojišťovnu, takže opět odpadá povinnost evidovat výkony jinde.

3.4.1 Modelová část

Tento modul se skládá z patnácti různých databázových tabulek. Tabulky jsou ve vazbě 1:N a stejně jako u modulu Kartotéka splňují 3 NF a stejně jako v Kartotéce se u všech tabulek primární klíč skládá jen z jednoho atributu, takže návrh splňuje i BCNF a stejně jako u Kartotéky s tímto faktem nebylo počítáno, takže je to opět jistý bonus. ER diagram tabulek si můžeme prohlédnout na obrázku 3.3 a v následující části je stručně nastíněno, k čemu jednotlivé tabulky slouží.

Radioterapie_ozarovaci_protokol - tato tabulka obsahuje informace o ozařovacích protokolech. Je to klíčová tabulka modulu, protože jak je vidět, vážou se na ní všechny ostatní tabulky. Samotná tabulka je pak pomocí entity **pobyt_id** vázaná na seznam pobytů, které jsou pak vázány na konkrétního pacienta.

⁵Dose-volume histogram



Obrázek 3.3: ER diagram modulu radioterapie.

Radioterapie_plany - tato tabulka obsahuje radiologické plány. Jak bylo řečeno výše, plán obsahuje základní informace o ozařování (dávku na frakci, počet frakcí, počet frakcí na den, adt.). Na tuto tabulku se vážou tabulky, které nějakým způsobem informace o plánu rozšiřují, nebo upravují. Jsou to tabulky:

1. **Radioterapie_extra_frakce** - pomocí dat v této tabulce lze upravit výsledný počet frakcí. Řešení přes pomocnou tabulku bylo nutné z toho důvodu, že každá takováto odchylka od plánu musí být odůvodněná a zpětně dohledatelná.
2. **Radioterapie_ozarovani** - pomocí této tabulky systém uchovává informace o odzářených frakcích.
3. **Radioterapie_pole** - v této tabulce jsou uloženy informace o ozařovacích polích, které jsou uloženy ke konkrétnímu plánu. každý plán musí obsahovat alespoň jedno pole.
4. **Radioterapie_sledovane_organy** - v této tabulce jsou uložena data o sledovaných orgánech (jedná se o statistické údaje dávek do orgánů v okolí ložiska tumoru). Na tuto tabulku se dále váže **Radioterapie_dvh_data**, ve které je uložen podrobný vývoj dávky ve sledovaných orgánech.

Radioterapie_rozpis_frakci - v této tabulce jsou zaznamenány informace o plánování frakcí. V systému můžeme zaznamenat například to, že po každé desáté frakci musí pacient absolvovat portálovou vizitu s lékařem nebo to, že při poslední frakci se má pacientovi odebrat krev na rozbor. Tyto informace jsou uloženy právě v této tabulce.

Radioterapie_simulacni_protokoly - v této tabulce jsou uloženy informace o simulačních protkolech.

Radioterapie_stop - v této tabulce jsou uloženy data o přerušení soustavného ozařování. V nádorové RT je počítáno s tím, že pacient většinou chodí na ozařování každý den. V této tabulce jsou uloženy dny, kdy pacient ozařován nebyl. Rozlišují se tři druhy přerušení: absence, stop a porucha.

Radioterapie_vizity - v této tabulce jsou uchovány informace o vizitách. Vizita musí obsahovat zprávu vizity a informace o aktuální dávce ozáření. K vizitě lékař může přiložit hodnocení ARMSC ⁶ nebo obrázkovou dokumentaci. Tyto informace jsou uloženy v tabulkách **Radioterapie_vizita_armsc** a **Radioterapie_vizita_soubory**. Informace o dávkách jsou uloženy v **Radioterapie_vizita_davky**.

3.4.2 Řídící část

Modul Radioterapie má celkem čtyři controllery.

1. AjaxController
2. IndexController
3. OzarovnyController
4. ProtokolController

Stejně jako u modulu Kartotéka (a v celém systému) má **AjaxController** na starosti všechny operace spojené s ajaxovou technologií. Jedná se převážně o zobrazování modálních oken a interaktivní rozšíření prvků formuláře.

IndexController má na starosti úvodní stránku modulu a s tím spojené filtrování a řazení radioterapeutických pobytů.

⁶Acute Radiation Morbidity Scoring Criteria

OzarovnyController má na starosti řízení procesů v seznamu pacientů u jednotlivých přístrojů. Seznam pacientů lze různými způsoby filtrovat a řadit a navíc tento controller nabízí uživatelům jednoduchou denní statistiku.

ProtokolController má na starosti všechny ostatní procesy v modulu Radioterapie. Stará se o procesy okolo plánů, vizit a simulací. Generuje rozpis frakcí a stará se o vše okolo frakcí, ať už je to odzářením frakce (a poukládání výkonů do pojišťovny) nebo vyhlášení STOP, absence nebo poruchy.

3.5 Modul Chemoterapie

Modul Chemoterapie se stará o správu pacientů, kteří mají naordinovanou chemoterapeutickou léčbu. Základním pilířem tohoto modulu jsou protokoly chemoterapie, které obsahují podrobné informace o léčbě. Protokoly se skládají ze tří částí. Jsou to nosiče hydratace, přípravky premedikace a samotné přípravky chemoterapie. Organizace léčby je pak následující: *pobyt CHT - linie CHT - cyklus CHT*. Při vytváření cyklů se vybere právě jeden protokol, který bude po dobu léčby (vyspecifikovanou v protokolu) aplikován.

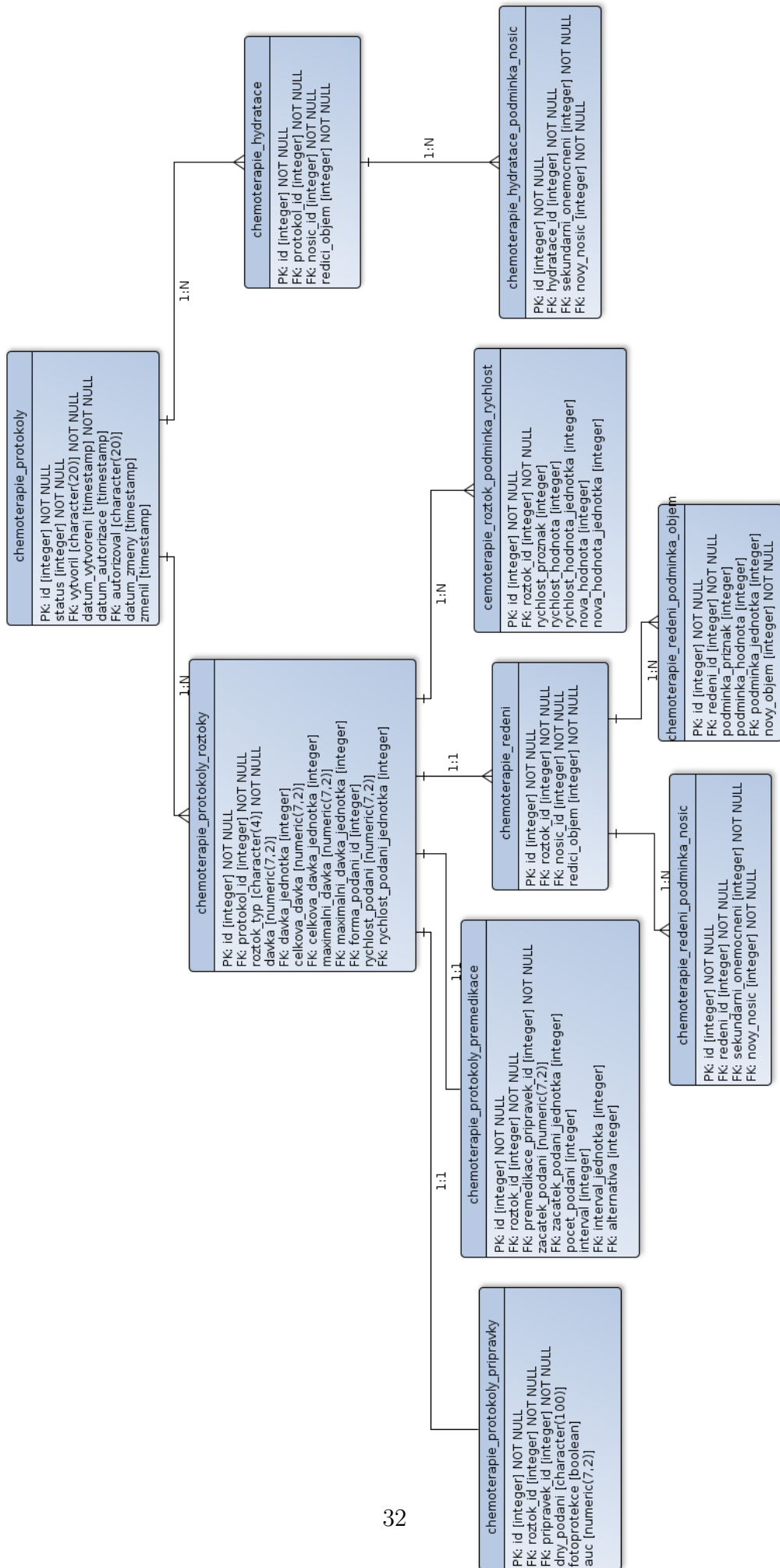
V protokolu je uvedeno, které dny (od zahájení léčby) se má daný přípravek aplikovat. Obsahuje i alterantivy pro výpočet doby aplikace nebo typu ředícího nosiče (některé typy nosičů nelze aplikovat pacientům s výskytem určitého druhu alergie a podobně). Systém při vytváření cyklu CHT vytvoří přehled aplikačních listů, ve kterých je uvedeno datum aplikace a přes systém nejruznějších kontrol vypočte aplikovanou dávku, když je to vyžadováno, vypočte i aplikovaný čas. Po tomto vytvoření aplikačních listů může zdravotnický personál přiřadit pacienta na inkriminované datum na lůžko a systém vykreslí do grafu údaj o délce aplikace (slouží k přehledu obsazenosti lůžek). V den a aplikace přípravku musí pacient absolvovat prohlídku lékařem. Lékař potvrdí aplikaci přípravku (může provést korekci dávky) a po potvrzení systém odešle informaci do ředění o požadavku na přípravu přípravku k aplikaci.

3.5.1 Modelová část

Databázové tabulky v tomto modulu jsou rozděleny na dvě skupiny. Tabulky s prefixem **chemoterapie** obsahují informace o protokolech chemoterapie. Tabulky s prefixem **Chemo** obsahují data o léčbě.

Nejprve se zaměřím na tabulky s informacemi o protokolech. ER diagram si lze prohlédnout na obrázku 3.4. Na první pohled je patrné, že se tato část skládá z deseti tabulek. Hlavní tabulka je pojmenována **chemoterapie_protokoly** a jsou v ní obsaženy základní informace o protokolu, jako například název protokolu, datum vytvoření, kým byl vytvořen, informace o revizích a v neposlední řadě informace o autorizaci protokolu. Lékař může pacientovi předepsat jen autorizovaný protokol, z toho vyplývá, že vytvořit protokol může libovolný pracovník, ale autorizovat jen určitá skupina pracovníků (z praxe to bývá primář a jeho zástupce). Jak bylo řečeno výše, protokol se může skládat ze tří složek. Nosiče hydratace jsou uchovány v tabulce **chemoterapie_hydratace** a na tuto tabulku se váže ještě další, **chemoterapie_hydratace_podminky_nosic**, ve které jsou uchovány informace o nosičích, které mají být aplikovány s určitým typem alergie nebo vedlejší diagnózy.

Informace o přípravku premedikace a samotného přípravku aplikace chemoterapie jsou uloženy v tabulkách **chemoterapie_protokoly_premedikace** a **chemoterapie_protokoly_pripravky**. Na schématu můžeme vidět, že tyto tabulky jsou vázány



Obrázek 3.4: ER diagram modulu chemoterapie - správa protokolů.

na hlavní tabulku přes tabulku **chemoterapie_protokoly_roztoky**. Toto optaření bylo použito z hlediska opětovného použití, protože když se podrobněji podíváme na informace o samotném přípravku a přípravku premedikace zjistíme, že mají mnoho společného. Hlavní rozdíly jsou v tom, že léčba se řídí podle samotných přípravků (obsahují informaci o dnech podání), takže přípravek premedikace je svázán se samotným přípravkem chemoterapie. Na druhou stranu, přípravek premedikace nese další informace, jako například dobu podání vztaženou na aplikaci samotného přípravku. Vytknutím společných prvků do tabulky **chemoterapie_roztoky** navíc odpadla nutnost duplikovat zbývající tabulky, které rozšiřují informaci o ředění (**chemoterapie_redeni**) a podmínkách pro stanovení rychlosti podání (**chemoterapie_roztok_podminka_rychlost**). Na tabulku s informacemi o ředění se dále vážou tabulky **chemoterapie_redeni_podminka_nosic** (obdobný důvod jako u **chemoterapie_hydratace_podminka_nosic**) a **chemoterapie_redeni_podminka_objem** (obdobné restriktce pro stanovení objemu ředění, nejčastěji vázána na bsa).

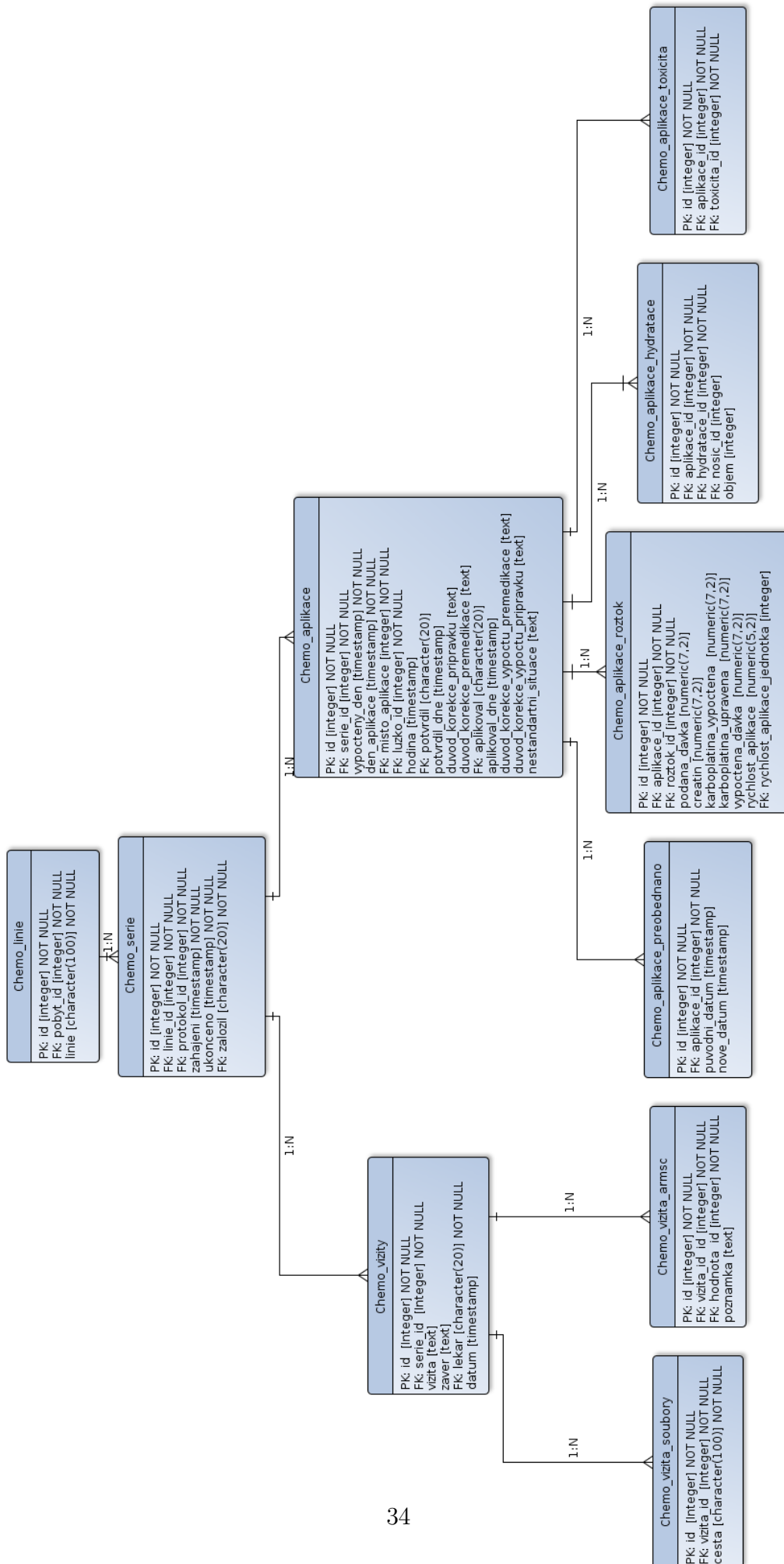
Druhá část tohoto modulu je tvořena informacemi o samotné léčbě. Jak bylo řečeno výše, organizace léčby chemoterapie vychází z modelu *pobyt CHT - linie CHT - cyklus CHT*. Tuto strukturu můžeme vyzorovat na ER diagramu této části modulu, který je znázorněn na obrázku 3.5.

Základní tabulkou je **Chemo.linie**, ve které jsou uloženy informace o liniích léčby. Tatu tabulka se váže (obdobně jako v modulu radioterapie) na tabulku **Pobyty**, která je dále vázána na konkrétního pacienta. Na tabulku s liniemi se váže tabulka **Chemo.serie**. Když se podrobně zaměříme na tabulku s liniemi, zjistíme, že je vytvořena vlastně jen kvůli přehlednosti. Prvotní návrh s touto tabulkou vůbec nepočítal. Jako základní léčebný cyklus je brána jedna série, takže veškeré vztažené údaje, jako datum vytvoření, autor a podobně jsou obsaženy v tabulce s informacemi o sérii, naproti tomu tyto informace nejsou obsaženy v tabulce s liniemi, protože to není podstatné. Veškerá důležitá data jsou obsažena v sérii, tato tabulka se váže na jeden konkrétní protokol chemoterapie. K sérii chemoterapie lze připojit lékařskou vizitu, která je totožná s vizitou radioterapie, pouze neobsahuje informace o aktuálním ozařování. Tyto vizity jsou uloženy v tabulce **Chemo.vizity**, kterou rozšiřují tabulky **Chemo.vizita_soubor** a **Chemo.vizita_armsc**. Zde je dlužno vysvětlit, proč jsou zduplikovány tři totožné tabulky. Vysvětlení je jednoduché. Zde se jedná o špatný návrh, který nerespektuje opětovnou použitelnost. Původní záměr byl takový, že Chemoterapie a Radioterapie budou dva navzájem nezávislé moduly. Tento záměr je splněn, avšak chyba s vizitou by šla řešit elegantněji tím způsobem, že by se mohl vytvořit modul **Vizita**, který by byl spojen jak s modulem **Chemoterapie**, tak s modulem **Radioterapie**.

Další tabulka vázaná na sérii chemoterapie je **Chemo.aplikace**. Bylo řečeno výše, že při zakládání série chemoterapie systém vytvoří aplikační listy. Tyto listy jsou právě uloženy v této tabulce. Na tuto tabulku se váže **Chemo.aplikace_preobednani**, kde jsou uloženy informace o změně data oproti plánu (nedodržení plánu se projeví na výsledné účinnosti série), dále se na ní váže tabulka **Chemo.aplikace_roztok** a **Chemo.aplikace_hydratace**, kde jsou uloženy informace o podaných látkách (hlavně aplikované dávky, čas aplikace a podobně). Poslední tabulka, která se váže na aplikační listy, je **Chemo.aplikace_toxicita**, kde jsou uloženy informace o výsledcích toxicity pro konkrétní aplikační list.

3.5.2 Řídící část

Tento modul obsahuje celkem pět controllerů.



Obrázek 3.5: ER diagram modulu chemoterapie - chemoterapeutické pobyty.

1. AdminController
2. AjaxController
3. IndexController
4. LuzkaController
5. PacientController

AdminController obhospodařuje procesy spojené s vytvářením, editací a mazáním protokolů CHT léčby.

AjaxController má opět na starosti veškeré procesy související s AJAXovou technologií.

IndexController má opět na starosti úvodní stránku, čili zobrazuje, filtruje a řadí pacienty s CHT pobytem.

LuzkaController obhospodařuje procesy s lůžky. Pro snazší orientaci byla pro zdravotnický personál, který se stará o pacienty s CHT pobytem, vyvinuta grafická podpora zobrazování obsazenosti lůžek. V této části lze objednávat a přeobjednávat pacienty na jednotlivá lůžka a všechny tyto procesy má na starosti tento controller.

PacientController má na starosti prezentaci dat u jednotlivých pacientů. Stará se o výpočty množství jednotlivých roztoků na základě jednotek (ml/m³, ml/kg, ...), vybírá druhy přípravků hydratace a ředění na základě alergií a kontroluje další vlastnosti, které je možné zadat v protokolu CHT. Tento controller se dále stará o potvrzování taktó vypočtených dávek a o prezentaci léčby v podobě kalendáře.

3.6 Instalační příručka

3.6.1 HW a SW požadavky na server

Pro instalaci onkologického IS M2 je nutné mít samostatný server, který má vyřešené zálohování a alternativní napájení. Jelikož výše bylo řečeno, že podniková politika upřednostňuje open sourceové řešení, proto je onkologický IS M2 dodáván se serverem s operačním systémem na bázi Unixu.

Jelikož je onkologický IS M2 postaven na webové platformě, je nutné na tomto serveru mít nainstalovaný webový server (ideálně Apache), databázový server (ideálně PostgreSQL) a PHP (minimálně verze 5.2, nižší verze nejsou podporovány). Pro webový server Apache je nutné povolit direktivu *mod_rewrite*, *mod_deflate* a *mod_expires* a dále je nutné mít povolený konfigurační soubor *.htaccess*.

Na začátku této práce bylo řečeno, že daň za použití frameworku je rychlost, pro minimalizování této daně je navíc nutné, aby byl na serveru nainstalován Zend Guard/Optimizer a eAccelerator ⁷.

Další nutností pro instalaci je mít k dispozici databázový server (pro snadnější popis vybereme PostgreSQL) s databází ois-m2 a uživatelem ois-m2 se všemi právy k databázi ois-m2.

Poslední požadavek na plné využití OIS M2 je nutnost mít na serveru nainstalovaný interpret Javy - JVM. Bez JVM by nebylo možné realizovat komunikaci s PACS serverem a nebylo by možné využít software pro automatické nahrávání výsledků laboratorního vyšetření.

⁷Více informací o rychlosti frameworků na adrese <http://www.root.cz/clanky/velky-test-php-frameworku-zend-nette-php-a-ror/>

Shrnutí HW a SW požadavků

1. potřebujeme server s OS na bázi debian (debian, ubuntu)
2. potřebujeme mít zprovozněný web server Apache
 - nutné mít povolený konfigurační soubor .htaccess
 - nutné mít povolený modul mod_rewrite
 - doporučené moduly mod_deflate a mod_expires
3. nutné nainstalovat PHP (optimálně 5.2 a novější)
 - optimální je mít zprovozněný Zend_Guard a eAccelerator
4. Nutné nainstalovat postgresQL s databází ois-m2 a uživatelem ois-m2 se všemi právy k databázi ois-m2
5. Pro správné připojení k PACS serveru a načítání laboratorních výsledků je nutné mít nainstalováno JVM

3.6.2 Postup instalace OIS M2

Jelikož je OIS M2 vysoce specifický SW, který není určený pro libovolné použití, nemá smysl vytvářet automatický instalátor. Základní instalace se skládá z pěti jednotlivých kroků a z toho poslední tři kroky jsou konfigurace aplikace.

V prvním kroku nahrajeme na databázový server strukturu databáze. Použijeme-li databázový server postgresQL a čistou zálohu databáze ois-m2.back, postačí nám použít služeb programu pg_restore, který je součástí instalace databázového serveru postgresQL. Kdyby byl OIS M2 instalován na jiný databázový server, museli bychom použít služby konkrétního databázového serveru, nebo spustit transakci samotného SQL dotazu. Bylo by ale nutné SQL dotaz přizpůsobit konkrétnímu databázovému serveru.

Druhý krok je nahrání zdrojových kódů na server. Doporučené umístění je `/var/www/ois-m2`.

Třetí, čtvrtý a pátý krok je konfigurace nahraného systému.

Shrnutí instalace OIS M2

1. nahrání databáze (ideálně pomocí pg_restore, pro jiný databázový server nutno spustit sql dotaz, který vytvoří základní databázovou strukturu)
2. nahrání zdrojových kódů na server (ideálně do `/var/www/ois-m2`)
3. změnit v `./app` konfigurační soubor `cfg.ini` (hlavně přístup do databáze a základní cesty)
4. upravit konfiguraci modulů v tabulce `configModules` (hlavně nastavení adres Pacs serverů a podobně)
5. nastavit správnou licenci v tabulce `licence`

3.6.3 Nutné kroky po konfiguraci

Aby systém mohl fungovat, je nutné po instalaci ještě provést pár kroků, které se týkají samotného běhu systému a bezpečnosti systému.

1. nastavit správné oprávnění pro specifické složky.
 - `./cache`
 - `./temp`
 - `./log`
 - složku pro potřeby eAcceleratoru, pokud není nastaveno na `/temp`

2. prověřit existenci symbolických odkazů ve složkách modulů v `./app/modules/*`
3. prověřit existenci souboru `.htaccess` ve složce `./app`

Obzvláště poslední krok je z hlediska bezpečnosti důležitý. Kdyby se nedopatřením stalo, že by nedošlo k nahrání souboru `.htaccess` do této složky, mělo by to nepříjemný dopad na bezpečnost.

3.6.4 Doporučené úpravy po konfiguraci

Posledním krokem instalace jsou doporučené úpravy. Tyto úpravy nemají přímý vliv na funkci samotného systému, je však vhodné se těmito úpravami zabývat.

1. Vytvoření uživatelských účtů
2. Vytvoření lokalit pro přístup
3. nahrát logo zdravotnického zařízení

Bez realizace prvního bodu by sice systém fungoval, ale k čemu by to bylo dobré, když by do něj nikdo nemohl přistupovat?

Druhý bod se týká opět bezpečnosti. Když se totiž někdo snaží přistoupit do systému z jiné než uložené lokality, systém bude vyžadovat přísnější kontrolu autentizace.

Třetí, poslední, doporučený krok je nahrání loga zdravotnického zařízení. Je to poslední úprava celé konfigurace, protože nemá vůbec žádný vliv na fungování systému, jen upravuje vizuální podobu tiskových sestav.

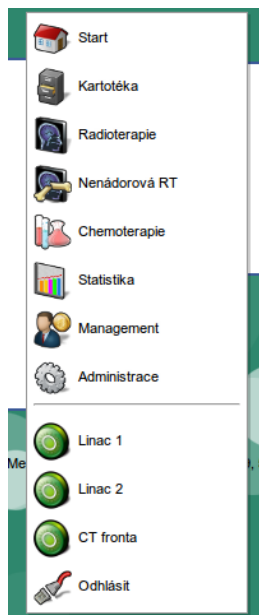
3.7 Uživatelská příručka

Hlavním rozcestníkem celého systému je hlavní menu. Na obrázku 3.6 je znázorněné menu `superusera`⁸. Menu lze vyvolat pravým kliknutím myši kamkoliv v prohlížeči a do jisté míry lze ovlivnit i jeho položky. Restrikce položek se provádí na základě oprávnění (můžeme vidět na obrázku 3.7) a expanzi může ovlivnit každý uživatel zvlášť a takovéto položky mají ikonu zeleného vypínače. Na obrázku menu 3.6 jsou to položky `Linac 1`, `Linac 2` a `CT fronta`. Takto volitelné menu usnadňuje přístup do částí systému, které uživatel využívá nejčastěji.

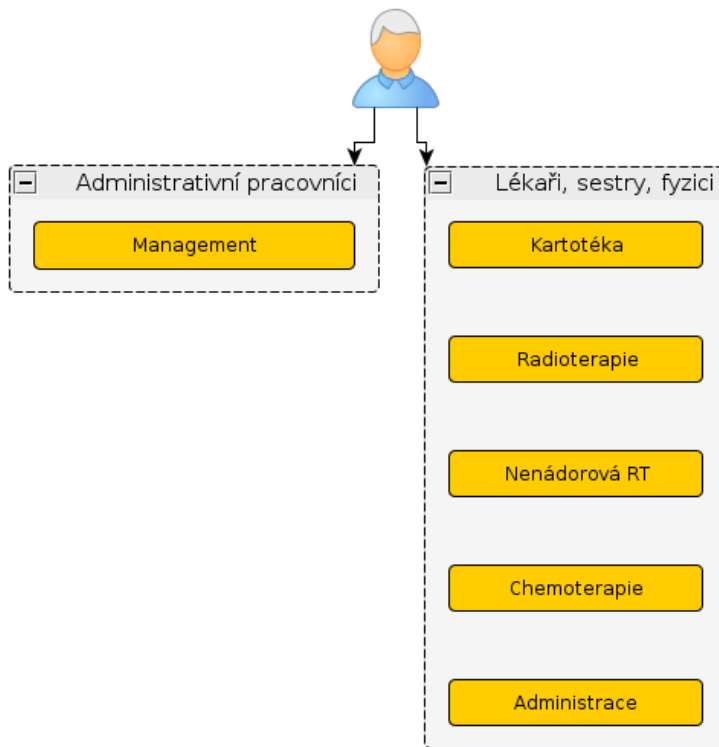
vedlejším rozcestníkem jsou záložky, jejichž podobu si můžeme prohlédnout na obrázku 3.8. Obsah záložek je ovlivněn stránkou, na které se momentálně v systému nacházíme. Záložka z obrázku 3.8 pochází z centrální karty pacienta. Výhoda těchto záložek spočívá v tom, že přepínání mezi záložkami nevyžaduje obnovu stránky, takže toto řešení do značné míry zrychluje systém. Kliknutím na záložku totiž akorát přepneme viditelnost daných bloků. Navíc jsou záložky nastaveny tak, že jsou plně kompatibilní s tlačítkem `Zpět` v internetových prohlížečích, takže využíváním záložek nemusíme měnit styl zaběhnuté práce.

Posledním rozcestníkem v systému jsou odkazy mezi navazujícími moduly. Slouží k tomu, abychom mohli rychle a jednoduše přeskakovat mezi moduly. Příklad využití může být takovýto: Prohlédneme si ozařovací protokol RT léčby a po té budeme chtít vytisknout pacientovi žádanku na CT vyšetření. To znamená nutnost dostat se z modulu `Radioterapie` do modulu `Kartotéka`. K tomuto účelu nám právě slouží toto rozcestí. Nalezneme ho vždy na konci prohlížené stránky. Konkrétně v ozařovacím protokolu bychom tam našli odkazy do centrální karty a do karty pobytu. Obsah tohoto rozcestníku je také závislý na stránce,

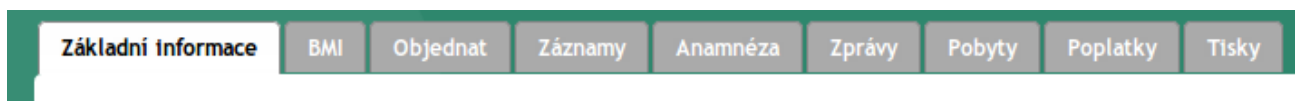
⁸role `superuser` je nejvýše v hierarchii oprávnění



Obrázek 3.6:
Hlavní menu



Obrázek 3.7: Procesy základní úrovně



Obrázek 3.8: Záložky

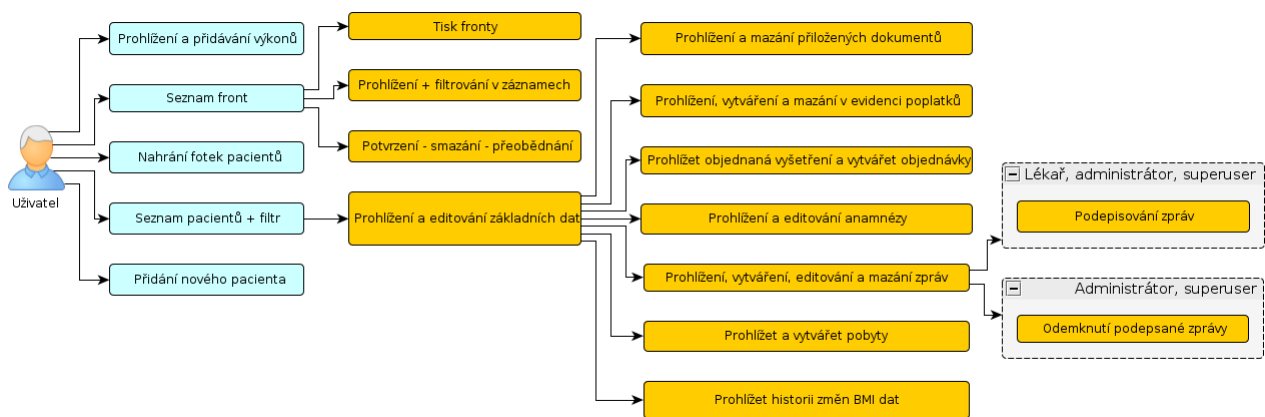
kteřou právě prohlížíme, ale nejčastěji toto rozcestí tvoří kolečko *CENTRÁLNÍ KARTA - KARTA POBYTU - RADIOTERAPIE* (popř. *CHEMOTERAPIE*).

3.7.1 Příručka kartotéky

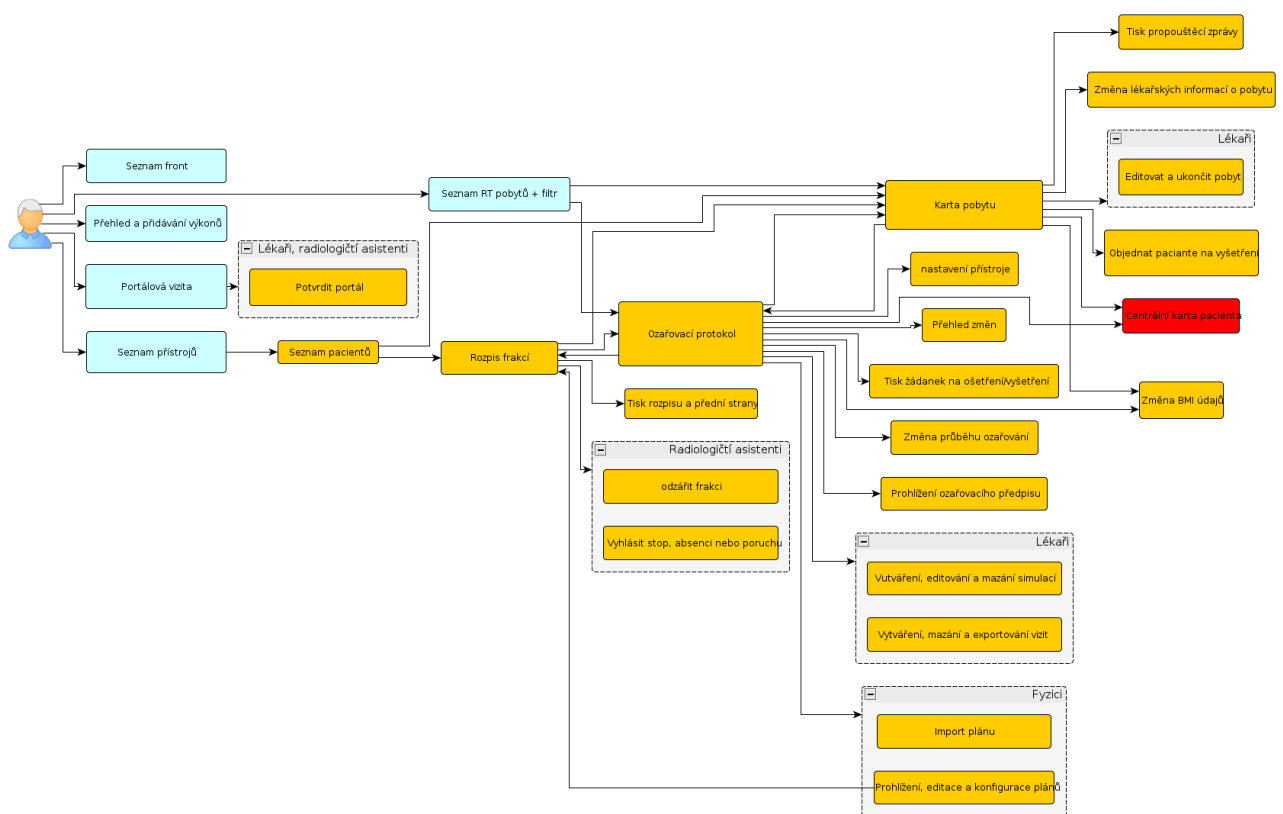
V hlavním modulu celého OIS M2 jsou soustředěny základní operace s pacienty bez ohledu na to, jakou podstupují léčbu. Na obrázku 3.9 můžeme vidět znázorněné procesy, které můžeme v tomto modulu provést. Při podrobném zkoumání obrázku si lze všimnout toho, že v tomto modulu nejsou nijak složitě řešené restriktce oprávnění. Jediné procesy, které jsou limitovány uživatelskou rolí, jsou podepisování zpráv (zprávu smí podepsat jen lékař, v případě problémů administrátor či superuser) a odemknutí podepsaných zpráv (jen administrátor a superuser). Důvod nízkého stupně restrikcí je dán tím, že základní operace s pacienty v systému mají na starosti sestry, které představují (vyjma pomocného personálu) nejnižší stupeň zdravotnického personálu.

3.7.2 Příručka radioterapie

Jak již bylo řečeno, modul radioterapie obhospodařuje veškeré procesy pacientů, kteří mají naordinovanou RT léčbu. Na obrázku 3.10 jsou znázorněné procesy, které je možné



Obrázek 3.9: Uživatelské procesy v kartotéce.



Obrázek 3.10: Uživatelské procesy v radioterapii.

v modulu vyvolat. Lze si všimnout, že restriktce oprávnění je tu vyšší než v modulu kartotéka.

Základní jednotkou RT léčby v OIS M2 je **ozařovací protokol**. K ozařovacímu protokolu se vážou veškeré informace o léčbě. V OIS M2 je možné mít k jednomu RT pobytu více ozařovacích protokolů, ale této možnosti není nevyužíváno. Obecně tedy platí, že jeden RT pobyt obsahuje jeden ozařovací protokol.

K ozařovacímu protokolu může fyzik nahrát RT plán (popřípadě plány), který, jak bylo řečeno výše, nese informace o léčbě. Důležité údaje (z hlediska dalších procesů) v RT plánu jsou počet frakcí celkem a počet frakcí na den. Z těchto údajů se totiž dále generuje rozpis frakcí a v tomto rozpisu mohou radiologičtí asistenti potvrzovat odzářené frakce. Systém potom nabízí přehled, kolik frakcí je odzářeno a jaké je množství aplikované dávky na cílové místo. Kromě těchto údajů se systém dále postará na základě údajů z plánu, o to, jaké má poukládat výkony do databáze. Zdravotnickému personálu tak odpadá nutnost evidovat výkony související s ozařováním ručně. K nahranému plánu lze také pomocí časování vyznačit, které úkoly se v který den ozařování mají provést. Takovéto informace se zobrazí v rozpisu frakcí a eliminuje to tak možnost některý úkol vynechat. Jedná se o již zmiňovanou portálovou vizitu (s lékařem nebo bez lékaře), invivo, boost a vizita.

OIS M2 nabízí nahrání libovolného počtu plánu a jejich konfiguraci. Lze tedy bez větších problémů nastavit to, aby běžely dva plány současně nebo třeba to, aby druhý plán začal až den po té, co bude ukončen první plán. Po odzáření poslední frakce v plánu dojde automaticky k uzamčení daného plánu.

Dalšími informacemi, které se vážou na ozařovací protokol, jsou lékařské vizity a simulace včetně tisků. Vizity a simulace může v OIS M2 vytvářet, editovat a mazat pouze lékař. Prohlížet a tisknout tyto údaje pak může kterýkoliv uživatel.

Poslední procesy v tomto modulu jsou připojeny z modulu Kartotéka a jedná se o tisk žádanek na nejruznější vyšetření/ošetření. Tyto procesy byly přidány na žádost uživatelů.

3.7.3 Příručka chemoterapie

Modul chemoterapie je jediný z modulů, který momentálně není používán, protože je pořád ve stádiu vývoje. Je to zapříčiněno tím, že organizace léčby je velice složitá a je složité vyvinout modul tak, aby vyhovoval veškerému personálu. V současné době je v testování již třetí verze tohoto modulu a s ostrým nasazením se počítá ve čtvrtém kvartálu roku 2012. S ohledem na tuto skutečnost lze očekávat, že v tomto modulu dojde ještě k některým dílčími změnám, které by ale měly mít minimální vliv na hlavní funkcionalitu modulu.

Modul chemoterapie je také jediný modul, který je nutné před použitím nakonfigurovat. Jedná se o konfiguraci léčebných protokolů chemoterapie. V současné verzi je možné k jednomu protokolu nadefinovat libovolné množství hydratačních prostředků s definicí výjimek nosičů, lze nadefinovat libovolné množství prostředků premedikace s celou škálou restrikcí pro různé typy pacientů, ať už se jedná o podmínky sekundárního onemocnění, nebo přizpůsobení dávky na základě tělesných proporcí pacientů. Systém nabízí u léčebných prostředků zadat údaj o ředění, tento údaj by měl vyplnit uživatel s rolí lékárníka. Proto by měl proces schvalování vypadat následovně:

Lékař vytvoří protokol - lékárník zreviduje ředěné přípravky - primář schválí protokol

Vytváření těchto protokolů je velice pracné a zdlouhavé, bohužel ale zatím neznáme způsob, jak práci ulehčit.

Současná verze modulu zatím umožňuje vyvolat procesy, které jsou znázorněny na obrázku 3.11 (na tomto obrázku nejsou znázorněny vazby mezi procesy u správy protokolů CHT, ty se v podstatě omezují jen na přidání a odebrání určitého typu přípravku). Při zakládání cyklu CHT systém automaticky vygeneruje 1. léčebnou linii a lékaři poté mohou k cyklu přidat schválené protokoly. Po přidání 1. protokolu systém vygeneruje kalendář, kde můžeme vizuálně sledovat rozložení léčby. Kalendář je znázorněn na obrázku 3.12. Podobný kalendář se zobrazuje u jednotlivého detailu protokolu CHT s tím rozdílem, že v něm jsou vyznačeny pouze položky daného protokolu.

Po rozkliknutí detailu protokolu můžeme na první záložce vidět detaily protokolu. Jedná se o seznam veškerých aplikovaných přípravků (hydratace, premedikace, přípravky CHT). Na záložce druhé jsou pak rozepsány jednotlivé aplikační listy (aplikační list = jedna aplikace).

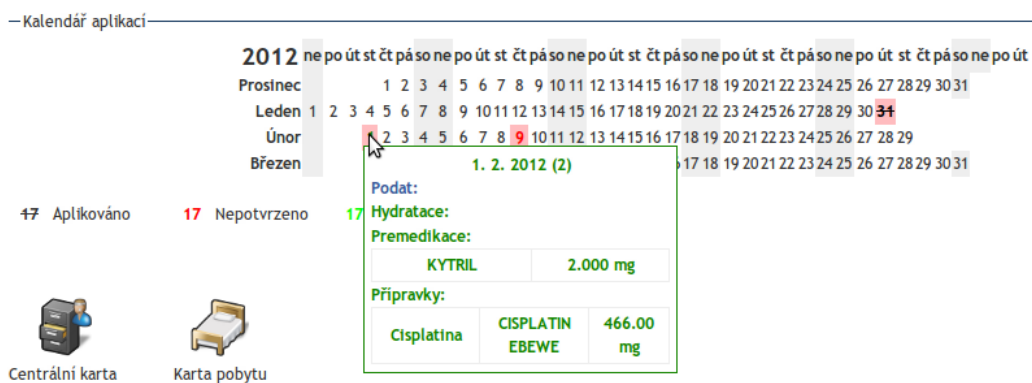
U jednotlivých aplikačních listů můžeme vyvolat celkem pět procesů a ideálně v tomto pořadí:

1. Potvrzení aplikačního listu (kontrola vypočtené dávky, popřípadě 1. korekce)
2. Výběr místa aplikace (zařízení, lůžko, hodina, popřípadě změna data aplikace)
3. Zapsání neočekávané události
4. Potvrzení aplikace léčebného přípravku (popřípadě 2. korekce)
5. Zapsání sesterské dokumentace

Pro realizaci druhého bodu lze využít přehled obsazenosti, kde můžeme opět graficky pozorovat přehled obsazenosti jednotlivých lůžek. Na první pohled je tedy vidět například to, že čtvrté lůžko je mezi 11:00 až 13:00 hodinou volné a pomocí rychlých nástrojů můžeme pacienta na toto lůžko umístit. Systém pak z doby aplikace léčebného přípravku vyznačí na grafu dobu, po kterou bude pacient na lůžku. Na tomto místě je možné v rychlosti zařadit pacienta, který nemá založený CHT pobyt a ani nemá založený cyklus CHT. Tato funkce byla přidána na základě praxe. Funguje to tak, že sestra vyhledá pacienta podle rodného čísla (s funkčním našeptávačem), vybere protokol (který se dozví na základě telefonického kontaktu s lékařem), vybere hodinu a místo aplikace a systém sám na základě dostupných údajů uloží veškeré informace, které chybí, tedy může založit pobyt, cyklus i linii. Tento přehled je možné si prohlížet v závislosti na dnech a do budoucna se počítá s tím, že se přehled rozšíří o jednotlivé ambulance.



Obrázek 3.11: Uživatelské procesy v chemoterapii.



Obrázek 3.12: Kalendář aplikací chemoterapie.

4 Závěr

Tato práce měla stanovené tři cíle, které byly zpracovány ve dvou hlavních kategoriích této práce. První dva cíle (zhodnocení současné NIS na trhu a popis technologií použitých při vývoji) jsou popsány v kapitole Dostupné technologie a systémy. Poslední cíl (popis vývoje modulů OIS M2) je rozebrán v kapitole Onkologický informační systém M2.

V první části první kapitoly jsem se zaměřil na dostupné technologie určené pro vývoj online aplikací. Provedl jsem krátké porovnání technologie ASP.NET a skriptovacího jazyka PHP. U PHP jsem se zaměřil na frameworky toho jazyka, konkrétně na CakePHP, Nette a Zend framework. V této části jsem provedl základní popis jednotlivých frameworků i s pohledem do historie. Podle mě nelze v dohledné době očekávat nějakou výraznou změnu na poli frameworků jazyka PHP. Možná by stálo za úvahu do srovnání přidat i framework Symfony, o kterém se můžeme dočíst na různých diskusních fórech, myslím si ale, že na území České republiky se volba frameworků bude zaměřovat nejčastěji na Zend framework a Nette. Nette foundation uvedlo na začátku roku 2012 verzi Nette 2.0, stejný krok se v dohledné době očekává i od Zend Technologies se Zend frameworkem 2.0. Nette do značné míry těží z oblíbenosti jeho autora a podle mě je to framework vhodný takřka na cokoliv, jak se lze přesvědčit z referencí projektů, které jsou postaveny na Nette. Na závěr této části jsem provedl krátké zhodnocení dvou open source databázových serverů a to MySQL a PostgreSQL. Stejně jako u frameworků jsem provedl krátké seznámení s oběma servery i s krátkým pohledem do historie, podle kterého lze sledovat postupné dorovnávání obou serverů. PostgreSQL disponuje většími možnostmi, naproti tomu MySQL je na základní operace rychlejší, nabízí však menší množství nastavení zálohování. MySQL také těží z toho, že je součástí základního softwareového vybavení webserveru, tím pádem není (a nikdy nebyl) velký problém tento databázový server nainstalovat a proto je ve větším povědomí programátorů webových aplikací.

V druhé části první kapitoly jsem představil dvě známé společnosti na poli zdravotnictví, které se zabývají tvorbou nemocničních informačních systémů. Byla to společnost Stapro s.r.o. s produkty FONS Akord a FONS Enterprise a společnost ICZ a.s. s produktem Medical Process Asistant (MPA). Jak bylo řečeno v závěru této části, myslím si, že společnost Stapro s.r.o. má daleko větší potenciál na trhu s nemocničními informačními systémy. K tomuto závěru mě vede to, že společnost na sebe nabalila různými fúzí většinu společností, které u nás nabízely své systémy (AKORD SOFTWARE, MLAB software, HiComp a IT divize GreyFox ve společnosti Medicon) a tím upevnila v průběhu let 2005 až 2010 své postavení na trhu. Naproti tomu se společnost ICZ a.s. nezaměřuje pouze na zdravotnický trh. Z referencí nebylo jasné, zda společnost dodává produkt MPA i do zemí evropské unie, ale s ohledem na rozložení sil na českém trhu a na fakt, že MPA disponuje možností plné lokalizace, je podle mě osud toho NIS svázán spíše se zahraničními instalacemi.

V druhé kapitole se věnuji popisu vývoje základních modulů onkologického informačního systému M2. Na začátku této kapitoly se věnuji seznámení se systémem taktéž s pohledem do historie. Důležitý poznatek této části je ten, že OIS M2 nemá za úkol nahradit NIS. Má za úkol NIS vhodně doplnit a nabídnout možnost komunikace mezi OIS M2 a jakýmkoliv NIS. Po seznámení s koncepcí OIS M2 popisují vývoj základních modulů to-

hoto systému. Jelikož se na vývoji podílím už více než dva roky, mohu s klidným svědomím říci, že jsem se zaměřil na nejzajímavější poznatky tohoto projektu. U jednotlivých modulů kladu důraz na představení modelové části modulu a to z toho důvodu, že je podle mého názoru nejzajímavější. Okrajově se u modulů věnuji řídicí část a zobrazovací část jsem z popisu vypustil úplně, z důvodu nízké relevance.

V závěru této kapitoly popisují instalaci OIS M2 a základní principi práce se systémem. Je zde vysvětleno, proč OIS M2 nedisponuje instalačním souborem a dále jsou popsány jednotlivé kroky při nasazování OIS M2. Popis principů práce se systémem je rozdělen podle modulů a součástí každé této části je graf, na kterém lze sledovat procesy (a jejich vazby), které je možné v modulu vyvolat.

Na tomto místě bych se chtěl ještě zamyslet nad dalším vývojovým potenciálem OIS M2. V práci je řečeno, že modul Chemoterapie je jediný z modulů, který není v současné době používán. Proto se ve výhledové době vývojový potenciál bude zaměřovat na tento modul. Dalším cílem je sestavit pro jednotlivého pacienta časovou osu jeho léčby. Lékař by pak měl možnost v přehledné grafické podobě pozorovat vývoj nemoci od ranné fáze až do léčby lineárním urychlovačem či aplikace přípravků chemoterapie. Vzhledem k použitým technologiím není v dohledné době vývojový potenciál technologicky omezen.

Literatura

- [1] Zend Technologies Ltd.: *ZF Zend Framework* [online]. c2011 [cit. 2011-11-30]. Zend Framework & MVC Introduction. Dostupné z WWW:
<http://framework.zend.com/manual/en/learning.quickstart.intro.html#learning.quickstart.intro.mvc>
- [2] Cake Software Foundation, Inc.: *Cake Software Foundation* [online]. c2011 [cit. 2011-11-30]. About the Foundation. Dostupné z WWW:
<http://cakefoundation.org/pages/about>
- [3] Nette Foundation: *Nette framework* [online]. c2011 [cit. 2011-11-30]. Tvůrci. Dostupné z WWW:
<http://doc.nette.org/cs/credits>
- [4] Nette Foundation: *Nette framework* [online]. c2011 [cit. 2011-11-30]. Licenční politika. Dostupné z WWW:
<http://doc.nette.org/cs/licence>
- [5] Zend Technologies USA, Inc.: *ZF Zend framework* [online]. c2011 [cit. 2011-11-30]. New BSD License. Dostupné z WWW:
<http://framework.zend.com/license>
- [6] Jiří Kosek: *Zdroják.cz* [online]. 24. 10. 2011 [cit. 2011-11-30]. Polyglot aneb webovým kóděrem pod obojí. Dostupné z WWW:
<http://zdrojak.root.cz/clanky/polyglot-aneb-webovym-koderem-pod-oboji/>
- [7] W3C.org: *XHTML2 Working Group Home Page* [online]. 2010/12/17 [cit. 2011-11-30]. Mission of the XHTML2 Working Group. Dostupné z WWW:
<http://www.w3.org/MarkUp/>
- [8] Ján Šuňavec: *ROOT.CZ* [online]. 25.10.2005 [cit. 2012-02-25]. MySQL vs PostgreSQL vs Firebird. Dostupné z WWW:
<http://www.root.cz/clanky/mysql-vs-postgresql-vs-firebird/>
- [9] OpenACS: *OpenACS* [online]. 2000 [cit. 2012-02-25]. Why not MySQL?. Dostupné z WWW:
<http://openacs.org/philosophy/why-not-mysql.html/>