



**FAKULTA
INFORMAČNÍCH
TECHNOLOGIÍ
ČVUT V PRAZE**

ZADÁNÍ DIPLOMOVÉ PRÁCE

Název:	Návrh databáze a systému pro evidenci pacientů s akutní myeloidní leukémií
Student:	Bc. Tomáš Pastorek
Vedoucí:	Ing. Zdeněk Rybala, Ph.D.
Studijní program:	Informatika
Studijní obor:	Webové a softwarové inženýrství
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	Do konce letního semestru 2018/19

Pokyny pro vypracování

Navrhněte relační databázi a informační systém pro evidenci pacientů s akutní myeloidní leukémií pro účely Ústavu hematologie a krevní transfuze sídlící na Karlově náměstí v Praze.

Díličí cíle práce:

1. Obecně popište problematiku léčby leukémie a specifika akutní myeloidní leukémie.
2. Analyzujte aktuální způsob práce s daty pacientů s akutní myeloidní leukémií v ÚHKT a typické případy užití těchto dat.
3. Popište vliv nařízení GDPR o ochraně osobních údajů na evidované informace a nově vznikající systém.
4. Na základě analýzy v předchozích bodech a s ohledem na způsob využití dat navrhněte strukturu relační databáze pro uložení dat o pacientech. Navržený databázový model patřičně zdokumentujte a implementujte ve formě SQL skriptů.
5. S ohledem na způsob využití dat navrhněte uživatelské rozhraní pro práci s daty v databázi a vytvořte interaktivní mock-up model.

Seznam odborné literatury

Dodá vedoucí práce.

Ing. Michal Valenta, Ph.D.
vedoucí katedry

doc. RNDr. Ing. Marcel Jiřina, Ph.D.
děkan

V Praze dne 23. ledna 2018



**FAKULTA
INFORMAČNÍCH
TECHNOLGIÍ
ČVUT V PRAZE**

Diplomová práce

Návrh databáze a systému pro evidenci pacientů s akutní myeloidní leukémií

Bc. Tomáš Pastorek

Katedra softwarového inženýrství

Vedoucí práce: Ing. Zdeněk Rybala, Ph.D.

5. května 2018

Poděkování

Velké díky patří MUDr. Mgr. Cyrilu Šálkovi, Ph.D. za jeho čas a trpělivost při konzultování této práce. Dále chci také poděkovat vedoucímu práce, který mě vždy dovedl nasměrovat správným směrem.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 5. května 2018

.....

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2018 Tomáš Pastorek. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí a nad rámec oprávnění uvedených v Prohlášení na předchozí straně, je nezbytný souhlas autora.

Odkaz na tuto práci

Pastorek, Tomáš. *Návrh databáze a systému pro evidenci pacientů s akutní myeloidní leukémií*. Diplomová práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2018.

Abstrakt

Tato práce se zabývá datovou analýzou procesu diagnostiky a léčby akutní myeloidní leukémie v Ústavu hematologie a krevní transfuze. Práce popisuje současný stav evidence dat ve formě excelových tabulek, jeho problémy a možnost zlepšení procesu vytvořením relační databáze a návrhem společného systému pro usnadnění každodenní práce vědců, datových manažerů a lékařů. Výstupem práce je analýza současného stavu, návrh relační databáze a návrh uživatelského rozhraní systému, společně s wireframe prototypem.

Cílem práce není systém implementovat. Práce se zaměřuje na analytickou a návrhovou část.

Klíčová slova leukémie, datová analýza, relační databáze, uživatelské rozhraní

Abstract

This work deals with data analysis of the process of diagnosis and treatment of acute myeloid leukemia in the Institute of Hematology and Blood Transfusion. The thesis describes the current state of data recording in form of the excel tables, its problems and the possibility of improving the process by creating a relational database as well as proposing a common system to facilitate the

daily work of scientists, data managers and doctors. The output of the thesis is analysis of the current status, the relational database design and the design of the user interface of the system including the wireframe prototype.

The aim of the thesis is not to implement the system. The thesis focuses on the analytical and design elements.

Keywords leukemia, data analysis, relational database, user interface

Obsah

Úvod	1
Cíl práce	1
Struktura práce	2
1 Analýza	3
1.1 Akutní myeloidní leukémie	3
1.2 Formát evidence dat	8
1.3 Proces evidence dat	8
1.4 Problémy současného stavu	12
1.5 Vliv GDPR	14
1.6 Funkční požadavky na systém	18
1.7 Nefunkční požadavky na systém	23
1.8 Případy užití	24
1.9 Doménový model	27
2 Návrh relační databáze	29
2.1 Požadavky na databázi a transformaci dat	29
2.2 Konvence modelování	31
2.3 Rozdělení dat do tabulek databáze	32
2.4 Rozbor datových typů	34
2.5 Databázové datové typy	36
2.6 Modelování dat tabulky pacientů	36
2.7 Výjimky ze systematického modelování	42
2.8 Modelování další funkcionality	43
2.9 Postup převod dat	47
3 Návrh uživatelského rozhraní	49
3.1 Postup návrhu	49
3.2 Správnost návrhu	52
3.3 Rozbor obdobných systémů	55

3.4	Konceptuální návrh	62
3.5	Prototyp	65
Závěr		77
	Realizace a budoucí vývoj	77
	Automatizace návrhu databáze	78
	Dosažení cílů	78
Literatura		81
A Seznam použitých zkratk		83
B Obsah příloženého CD		85

Seznam obrázků

1.1	Mikroskopický pohled na leukemické buňky	4
1.2	Projevy AML: zbytnělé dásně	5
1.3	Projevy AML: krvácení kůže	5
1.4	Odběr kostní dřeně z prsní kosti	6
1.5	Graf přežití pacientů AML	7
1.6	Výřez z tabulky pacientů	9
1.7	Tabulka měření reziduální nemoci	10
1.8	Cytogenetické údaje	11
1.9	Nákres implementace grafu přežití	20
1.10	Nákres implementace grafu měření reziduální nemoci	20
1.11	Návrh datového zpracování cytogenetické metody FISH	21
1.12	Diagram případů užití	24
1.13	Diagram průchodu scénářem: Úprava dat pacienta	25
1.14	Diagram průchodu scénářem: Vykreslení grafu přežití	26
1.15	Diagram průchodu scénářem: Založení uživatelského účtu	27
1.16	Doménový model	28
2.1	Ukázka číselníkové tabulky	32
2.2	Zjednodušený databázový model	33
2.3	Ukázka výstupu analytického programu	35
2.4	Datová analýza sloupce <i>Extramedulární postižení</i>	38
2.5	Datová analýza sloupce <i>Předchozí chemo- nebo radioterapie</i>	39
2.6	Databázová implementace sloupce <i>Předchozí chemo- nebo radioterapie</i>	39
2.7	Databázová implementace sloupce <i>Indukce</i>	41
2.8	Databázová tabulka <i>obdecny_udaj</i>	41
2.9	Databázový model mutací	42
2.10	Databázový model léčby	44
2.11	Výřez databázového modelu	45
2.12	Nekompletní tabulka měření reziduální nemoci	46

3.1	Proces návrhu uživatelského rozhraní	49
3.2	PC Doktor - kartotéka pacientů	56
3.3	PC Doktor - historie léčby	57
3.4	WinMed 2 - detail pacienta	58
3.5	WinMed 2 - historie léčby	59
3.6	WinMed 2 - filtrování pacientů	60
3.7	Medicus - kartotéka pacientů	60
3.8	Medicus - detail pacienta	61
3.9	Medicus - historie léčby	61
3.10	Task graph	65
3.11	Obecné rozvržení obrazovky	66
3.12	Prototyp obrazovky nastavení číselníkových hodnot	67
3.13	Prototyp obrazovky s grafy přežití	68
3.14	Prototyp obrazovky správy uživatelů	69
3.15	Prototyp obrazovky logu systému	70
3.16	Prototyp dialogu s detailem změny	71
3.17	Prototyp obrazovky karty pacienta	72
3.18	Prototyp obrazovky reziduální nemoci	72
3.19	Prototyp formuláře zadání odběru	73
3.20	Pozicování popisků formulářového pole	74
3.21	Pozicování zástupné hodnoty formulářového pole	75
3.22	Značení povinných polí formuláře	75
3.23	Označení chyby formulářového pole	75

Seznam tabulek

2.1	Tabulka převodních konstant	38
2.2	Tabulka převodních konstant pro typ datum	39
2.3	Tabulka konstant operací logu	46
3.1	Rozdíl úrovní prototypů UI	51

Úvod

Ústav hematologie a krevní transfuze (dále také zkratkou ÚHKF), sídlící na Karlově náměstí, se již více jak 60 let zabývá mimo jiné výzkumem leukémie a možnostmi její léčby. Jednou z hlavních diagnóz ústavu je akutní myeloidní leukémie. Ústav eviduje data o bezmála 1500 pacientů, jejich osobní údaje, diagnostické údaje, informace o průběhu léčby a výsledku jejich léčení.

V současné době jsou data skladována v nepřehledných excelových tabulkách. Samotné soubory se vyskytují v různých odděleních a v různých verzích, přičemž ústav vynakládá nemalé prostředky na jejich sjednocení a normalizaci pro analytické potřeby.

V této práci se zabývám návrhem na zlepšení procesu evidence lékařských dat. Navrhuji konkrétní databázový model, který umožní společný přístup ke vždy aktuálním datům a nabídne normalizaci dat pro lepší přehlednost. Také se věnuji návrhu uživatelského rozhraní aplikace, která nabídne novou funkcionalitu usnadňující každodenní práci vědců, lékařů a datových analytiků pracujících v ústavu. Zároveň také zkoumám vliv nového nařízení Evropské Unie o ochraně osobních údajů GDPR na navrhovanou databázi a systém.

Cíl práce

Cílem práce je ve srozumitelné formě pro laické čtenáře představit problematiku diagnózy a léčby leukémie. Následně analyzovat a popsat postupy evidence dat spojených s léčbou akutní myeloidní leukémie v ÚHKF.

Na základě předchozí analýzy současného stavu, je mým dalším cílem navrhnout zlepšení, které usnadní práci odborníkům ústavu. Součástí návrhu je zohlednit také požadavky nařízení GDPR. Výstupem návrhu je relační databázový model pacientů a návrh uživatelského rozhraní systému pro lepší práci s daty, včetně wireframe modelu.

Práce bude sloužit jako návod a doporučení postupů pro tým programátorů, kteří budou systém implementovat.

Cílem práce není implementovat systém nebo jakékoli jeho součásti, cílem práce také není převést data pacientů do navržené databáze.

Struktura práce

V první kapitole *Analýza* postupně popisují problematiku diagnostiky a léčby leukémie. Následně se zaměřím na konkrétní proces evidence dat pracovníky ÚHKM, zkoumám formát evidence a problémy současného stavu. Následně zkoumám praktické požadavky na systém z hlediska legislativy a to především GDPR. Na základě analýzy specifikuji funkční a nefunkční požadavky na systém. Dále v kapitole představím koncept systému na diagramu případů užití a blíže se věnuji popisu některých scénářů. Závěrem stanovím obecný doménový model systému.

Druhá kapitola *Návrh relační databáze* se zabývá návrhem databázové podoby excelových souborů. Začátkem kapitoly popisují požadavky na databázi stanovené konvence modelování. Návrh následně začínám rozdělením dat do tabulek, pokračuji detailní analýzou datových typů a dále modeluji kompletní tabulky a relace mezi nimi. Konec kapitoly věnuji stanovení požadavků na převod dat.

Třetí kapitola *Návrh uživatelského rozhraní* popisuje postupný návrh uživatelského rozhraní pro navrhovaný systém. V úvodu se věnuji stanovení postupu návrhu, následně analyzuji znaky správného rozhraní a zkoumám implementaci v obdobných systémech. Samotný postup návrhu začínám konceptuálním návrhem a postupně se dopracuji až k tvorbě prototypu.

V závěru práce popisují budoucí implementaci systému a možné vylepšení mého návrhu.

Analýza

První kapitola mé práce se věnuje analýze problémové domény. V této kapitole popisují obecnou problematiku diagnózy a léčby akutní myeloidní leukémie, z datového pohledu zkoumám konkrétní procesy v ÚHKT a analyzuji současný stav evidence dat. Dále zkoumám požadavky na informační systém, který nahradí v současné době používané excelové tabulky.

1.1 Akutní myeloidní leukémie

Následující kapitola se věnuje vysvětlení základních pojmů a principů v léčbě leukémie. Předmětem této práce není zabývat se lékařským aspektem problematiky a proto je následující kapitola sepsána co nejsrozumitelněji pro laické čtenáře.

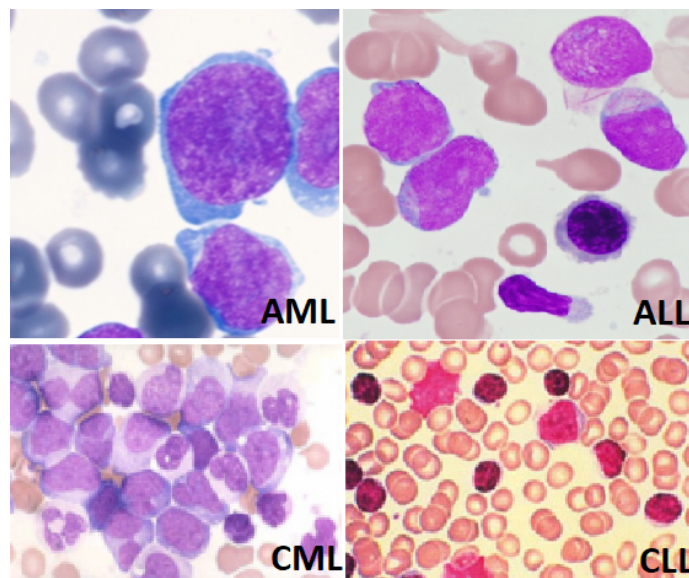
1.1.1 Leukémie

Leukémie je označení pro řadu chorob, které se dají vnímat jako rakovina krve. Rakovina je nádorové onemocnění, které spočívá v tom, že se skupina buněk v organismu vymkne kontrole a začne se nekontrolovatelně množit.

Nádorový proces v tomto případě postihuje bílé krvinky (leukocyty). Ty se v našem těle během celého života tvoří v kostní dřeni a některých dalších tkáních. Za určitých okolností se stane, že dojde k mutaci některé nezralé bílé krvinky a ta se začne nekontrolovatelně dělit. Není-li imunitním systémem včas rozpoznána a zlikvidována, začnou se její dceřiné buňky (klony) rozšiřovat krví po těle. Příčina mutace není mnohdy známá [1].

1.1.2 Akutní a chronické typy

Leukémie se dělí do dvou základních typů dle rychlosti nástupu nemoci. A to na chronický typ (leukémie s pomalým nástupem a průběhem nemoci) a akutní



Obrázek 1.1: Miskroskopický pohled na leukemické buňky [3] [5]

typ (rychlý nástup a průběh). Dále dělíme leukémie dle typu bílé krvinky, ze které nemoc vznikla na lymfoblastické a myeloidní.

Celkově tedy rozlišujeme 4 základní druhy leukémie:

AML Akutní myeloidní leukémie,

ALL Akutní lymfoblastová leukémie,

CML Chronická myeloidní leukémie,

CLL Chronická lymfoblastová leukémie

Obrázek 1.1 znázorňuje nádorové buňky zmíněných typů leukémie. Tato práce se nadále zabývá akutní myeloidní leukémií (označovanou také zkratkou AML).

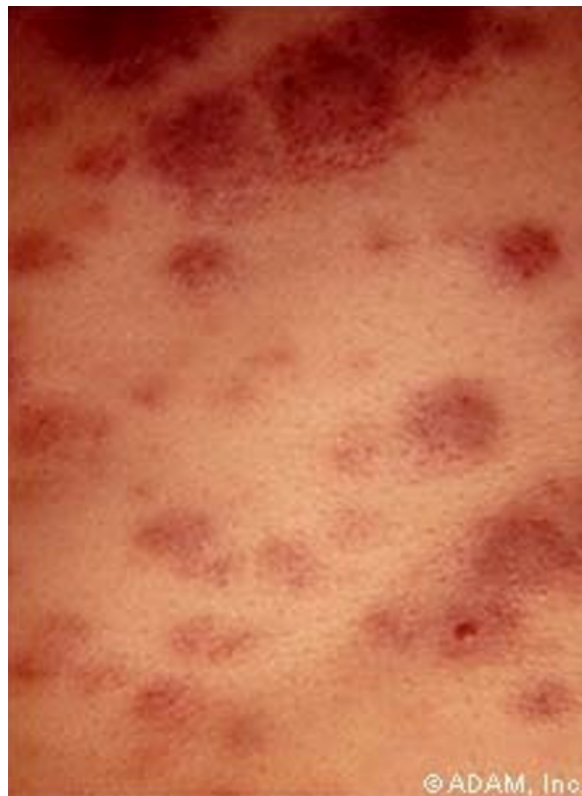
1.1.3 Projevy

Projevy odpovídají obecným znakům leukémie. Nemocný má zvýšený počet bílých krvinek, které se usazují v kostní dřeni, v lymfatických uzlinách a tělesných orgánech. Zásadní problémy tohoto procesu jsou dva. Nádorové bílé krvinky jsou nezralé a neumí plnit obrannou imunitní funkci. A za druhé utlačí nádorové leukocyty tvorbu ostatních krevních buněk (normální bílé krvinky, červené krvinky) a krevních destiček [2].

Příznaky, se kterými nemocný navštíví lékaře jsou nespecifické, časté jsou únava a snížený fyzický výkon. Příznaky mohou být také vnější, jako zbytnění dásní (viz obr. 1.2), nebo krvácení v kůži (viz obr. 1.3).



Obrázek 1.2: Zbytnělé dásně [3]



Obrázek 1.3: Krvácení kůže [3]



Obrázek 1.4: Odběr kostní dřeně z prsní kosti [3]

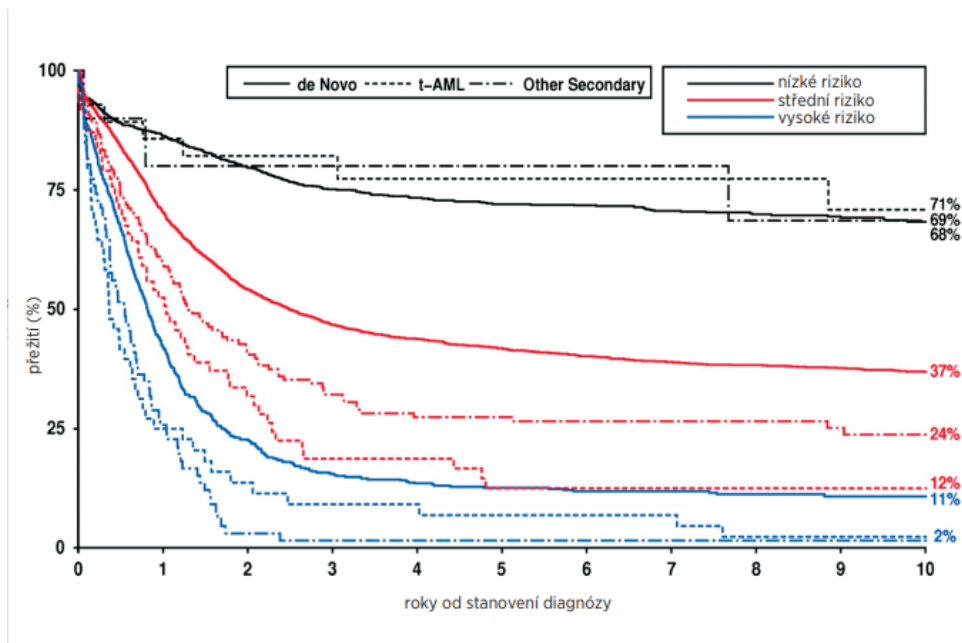
Pokud není nemoc včasně diagnostikována a léčena, může AML usmrtit pacienta během několika týdnů.

1.1.4 Diagnostika

Základním krokem k diagnóze akutní leukémie je vyšetření krevního obrazu, při kterém se stanoví diferenciální počet nezralých bílých krvinek (blastů).

Nález nemoci v krvi si vyžádá podrobné hematologické vyšetření, jehož součástí je vyšetření kostní dřeně. Tento vzorek se získává z prsní kosti pacienta (viz obr. 1.4). Vzorek se následně zašle několika laboratorům k vyhodnocení. Cytogenetická laboratoř zkoumá počet blastů. Cytochemická laboratoř zkoumá chemické složení buněk a pomáhá určit typ akutní leukémie. Typ leukémie dále vyhodnotí imunofenotypizační vyšetření. Cytogenetické vyšetření zkoumá strukturu a odchylky v chromozomech nemocného pacienta. A metody molekulární genetiky dále zkoumají fúzní geny a reziduální nemoci (o této metodice více v kapitole 1.3.3) [4].

Součástí diagnostiky je také klasifikace pacienta do rizikových (prognostických) skupin. Jedním z faktorů může být i věk pacienta.



Obrázek 1.5: Graf přežití pacientů AML

Dalšímu vysvětlení diagnostiky se věnuji později při analýze datových procesů.

1.1.5 Léčba

Základní metodou léčby je chemoterapie, která má za cíl postupně zničit veškeré nádorové buňky. Další možností je léčba transplantací kostní dřeně. Tento proces začíná zničením celé kostní dřeně pacienta a následnou transplantací zdravé kostní dřeně dárce. Při tomto procesu doufáme v to, že zdravé krvetvorné buňky se rozšíří a nahradí záměrně zničenou populaci. Tento zákrok je vysoce rizikový. Pokud nové buňky nejsou schopny fungovat, nebo dojde k infekci, často zákrok končí smrtí pacienta [2].

1.1.6 Prognóza léčby

Celkové přežití se u mladších pacientů pohybuje kolem 50%, u starších maximálně 15%. Po provedení transplantace přežívá po třech letech 40 až 60% pacientů [4]. Graf 1.5 znázorňuje přežití pacientů v závislosti na rizikové skupině [3].

1.2 Formát evidence dat

V předchozí sekci jsem představil základní problematiku diagnostiky a léčby AML, v této a následujících sekcích se zaměřím již na konkrétní způsob práce s daty v ÚHKKT.

Před tím, než začnu popisovat postup, jakým ústav pracuje s daty, představím jejich formát, ke kterému se budu v pozdější kapitole odvolávat.

V současné době ústav eviduje data ve formě tabulkových souborů pro program Microsoft Excel. Nejdůležitější souhrnné informace o pacientech obsahuje soubor *tabulka pacientů*. Další data obsahují *soubory reziduálních nemocí*. K významu dat se vyjádřím později, nyní chci představit pouze formát.

1.2.1 Tabulka pacientů

Tato komplexní tabulka eviduje informace o 1017 (prosinec 2017) pacientech s diagnózou, nebo podezřením na diagnózu AML. Formát tabulky (viz obr. 1.6) je následující. Na každém řádku je evidován jeden pacient a dále v necelých 200 sloupcích jsou informace o pacientovi. Pro zjednodušení budu tento soubor dále v textu označovat jako *tabulka pacientů*. Pokud nebude specifikováno jinak, tabulkou pacientů se myslí vždy excelová podoba dat.

1.2.2 Tabulky reziduálních nemocí

Zatímco tabulka pacientů obsahuje souhrnné informace pro celkový přehled, excelové soubory reziduálních nemocí obsahují historická měření každého pacienta (viz obr. 1.7). Zde každý jeden soubor obsahuje výsledky pro jednoho pacienta.

Každý takový soubor (tabulka reziduální nemoci) je pojmenován dle příjmení pacienta, což je zároveň jediné praktické propojení s tabulkou pacientů. Každý řádek tabulky reziduální nemoci obsahuje jedno měření. Soubory jsou rozděleny do složek dle typu reziduální nemoci měřené u pacienta.

1.3 Proces evidence dat

V této sekci navážu na dříve představený formát dat a vysvětlím, jaký význam data nesou a jaké procesy jsou v praxi implementovány pro evidenci těchto dat.

Při analýze dat se odkazují na datové sloupce tabulky pacientů, která je ve své digitální, omezené a anonymizované podobě přiložena na CD.

1.3.1 Diagnostické a osobní údaje

Celý proces začíná přijetím vzorku krve nebo kostní dřeně pacienta podezřelého na leukémii. Tento vzorek putuje do několika laboratoří zároveň. Každá

1.3. Proces evidence dat

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	
1	Číslo (z databáze ÚHK)	Nemocnice	RNA	Příjmení a jméno	Rodné číslo	Datum narození	Věk při dg.	Věková skupina	Datum dg.	Mužské pohlaví	Bydl. okres (místo)	Typ AML (FAB)	FAB (kód)	Klasifikace WHO	Sekundární AML	Předchozí hematol. onk	Předchozí onkol. onem	Předchozí chemo- nebo radioterapie	Dny od příznaků k dg.	Perform st WHO/ECOG	Extramedulární postiže	Leuko- [10 ⁹ /l]	WBC > 100, u APL > 5	Blasty (+pmy) v perif. [%]	Hemoglobin [g/l]	Trombo- [10 ⁹ /l]	Blasty (+ pmy) dřeň	Auerovy tyče	Faggot cells	Dysplázie	
501	500 ÚHK	1134					7365	5	12.11.07	1		AML M2	2.0	M/D	0					2	0	79.1	0	19	9.0	127	36	0	NA	1	
502	501 ÚHK	1154					2771	2	15.11.07	0		AML M3	3.0	IL s t15:	0					1	0	0	2.7	0	0	115	222	65	1	1	0
503	502 ÚHK	1172					6690	5	20.11.07	1		AML M2	2.0	M2bss	0					2	0	41.5	0	88	8.4	19	75	1	NA	0	
504	503 ÚHK	1232					6592	5	03.12.07	0		AML M2	2.0	M/D	0					0	0	6.1	0	15	9.6	83	34	0	NA	1	
505	504 ÚHK	1299					6466	5	14.12.07	1		AML M1	1.0	M1	0					2	0	94.7	0	95	7.7	15	89	0	NA	0	
506	505 ÚHK	1344					5703	4	21.12.07	1		RAEB-T	8.0	2.AML s 0							1	0	3.4	0	40	84.0	60	19	0	NA	1
507	506 ÚHK	1404					4551	4	03.01.08	0		s-AML	9.0	f-AML	1					1	1	9.2	0	29	11.2	37	36	0	NA	1	
508	507 ÚHK	1404					8051	5	16.01.08	1		AML M2	8.0	f-AML	1					1	0	19.2	0	27	10.0	121	24	0	NA	1	
509	508 ÚHK	1411					5861	4	21.01.08	1		RAEB-T	8.0	2.AML s 1	IDSMP					7	3	0	4.4	0	3	94.0	13	12	0	NA	1
510	509 ÚHK						6226	5	05.02.08	1		M2	2.0	4.AML s 0	NA				35	1	0	27.5	0	40	94.0	29	ND	ND	NA	ND	
511	510 ÚHK	1514					8003	5	12.02.08	1		s-AML	9.0	f-AML	1	CLL		FC, FCR		2	0	2.3	0	0	11.1	140	60	0	NA	0	
512	511 ÚHK	1520					3686	3	13.02.08	0		AML M1	1.0	M1	0					0	1	15.1	0	86	11.9	157	80	1	NA	1	
513	512 ÚHK	1530					5793	4	14.02.08	1		s-AML	9.0	2.AML s 1	CMML-2						1	0	43.4	0	50	120.0	46	56	0	NA	1
514	513 ÚHK	1588					4021	3	26.02.08	0		AML M3	3.0	IL s t15:	0					0	0	0.7	0	2	8.3	72	71	1	1	0	
515	514 ÚHK	1598					5903	4	04.03.08	1		RAEB-T	8.0	M/D	0					0	0	98.0	0	13	9.9	95	25	1	NA	1	
516	515 ÚHK	1620					5734	4	06.03.08	1		RAEB-T	8.0	M/LMDS	0					0	0	2.0	0	0	124	122	24	0	NA	1	
517	516 ÚHK	1453					5091	4	06.03.08	0		M5a	5.1	2.AML s 1	CMML							34.7	0	3	77.0	114	10	0	NA	0	
518	517 ÚHK	1629					6559	5	10.03.08	0		AML M2	2.0	M/D	0						0	0	2.6	0	22	11.1	196	40	1	NA	1

Obrázek 1.6: Výřez z tabulky pacientů (anonimizováno)

1. ANALÝZA

	A	B	C	D	E	F	G	H	I	J	K	L	M
	Č. izol.	Datum odběru	Den č.	$2(Ct\text{ abl} - Ct\text{ SG})$	Datum měření	$C_t\text{ SG}$ PB hodn 1	$C_t\text{ SG}$ PB hodn 2	$C_t\text{ SG}$ PB hodn 3	$C_t\text{ SG}$ PB prům	$C_t\text{ abl}$ PB hodn 1	$C_t\text{ abl}$ PB hodn 2	$C_t\text{ abl}$ PB prům	$C_t\text{ abl} - C_t\text{ SG}$
1													
2	O345	17.10.11	0	0,5932313	03.11.11	32,32	32,48	32,44	32,413	31,61	31,71	31,66	-0,75
3	O422	31.10.11	14	0,5796819	03.11.11	31,55	31,87	31,55	31,657	30,7	31,04	30,87	-0,79
4	O475	14.11.11	28	0,0099345	15.11.11	37,73	37,66	37,51	37,633	31,02	30,94	30,98	-6,65
5	O613	15.12.11	59	#####	20.12.11	neg.			#####			#####	#####
6	O680	10.01.12	85	#####	11.01.12	neg.			#####			#####	#####
7	P191	17.05.12	213	#####	21.05.12	neg.			#####			#####	#####
8	P910	28.11.12	408	#####	30.11.12	neg.			#####			#####	#####
9	R334	20.03.13	520	#####	25.03.13	neg.			#####			#####	#####
10				#####					#####			#####	#####

Obrázek 1.7: Tabulka měření reziduální nemoci

laboratoř vyhodnotí svoje měření nad daným vzorkem a následně dojde k celkovému vyhodnocení pacienta.

První na řadě s testováním je morfologická laboratoř, která produkuje prvotní diagnózu. V této laboratoři se počítají nezralé buňky (blasty). Jednou z podmínek pro diagnózu akutní leukémie je jejich počet převyšující 20% z celkového množství zkoumaných buněk. Vedoucí laborant založí pacientovi řádek v tabulce pacientů a vyplní osobní údaje, klasifikaci AML dle morfologického obrazu (toho, jak buňky vypadají) a údaje o předchozím onemocnění pacienta. Datový rozsah těchto informací v tabulce pacientů jsou sloupce A až U (dále A-U) (viz obr. 1.6).

Další z laboratoří diagnostikujících AML je cytogenetická laboratoř, která obdrží vzorek kostní dřeni a zkoumá chromozomy a jejich odchylky od zdravého člověka. Tato laboratoř následně vyplní datové sloupce AW a AX (viz obr. 1.8).

Laboratoř molekulární genetiky zkoumá vybrané fúzní geny a jejich mutace odpovídající datovému rozsahu sloupců AZ až DA.

Každé tři týdny se schází zástupci laboratoří na společné schůzi, kde společně vyhodnotí své nálezy. Na základě měření ze všech laboratoří se vyhodnotí u pacienta prognostická skupina, do které je pacient zařazen (BE-BG) a která určuje pravděpodobnost úspěchu léčby. Na této schůzce se dále rozhodne, zdali budou provedena další genetická vyšetření (BX-DP). Po schůzce má na starost doplnění dalších údajů datový manažer, který sbírá informace z výsledků laboratorních zpráv a propouštěcích nebo hospitalizačních zpráv.

1.3.2 Údaje spojené s léčbou

Po dokončení diagnózy je na řadě začít s léčbou, jejímž cílem je vyvolat stav indukce (stav bez leukémie). Takového stavu se dosáhne pomocí indukční léčby.

AW	AX
Cytogenetika	FISH
22/22: 46,XY	neg. MLL, 5,7,8,9
13/13 46,XY	neg. MLL, 5,7,8,9
NT	
3/17 47,XY,+11	3/21: add. MLL genu a v 9% jader. Neg. 5,7,8,9
22/22,46,XY	neg. MLL, 5,7,8,9
0 mit.	Monosomie 9 73% jader. neg. MLL, 5,7,8
22/22,46,XX	neg. MLL, 5,7,8,9
3/22 47,XY,+X	neg. MLL, 5,7,8,9,+X (jen 4,4%)
22/22 46,XY	neg. MLL, 5,7,8,9
21/21 46,XX	neg. MLL, 5,7,8,9,16,20
5/5 46,XY	PML/RARa

Obrázek 1.8: Cytogenetické údaje

Pokud první taková léčba neuspěje, provádí se druhá. Stav indukce se určuje především dle počtu blastů, který by měl být menší než 5% celkově napočítaných buněk, normálního krevního obrazu a nezávislosti pacienta na transfuzích (DS-EF).

Jakmile je dosažen stav indukce, je třeba tento stav udržet za pomoci udržovací konsolidační léčby v cyklech, většinou ve formě chemoterapie. Zpravidla je těchto léčeb tři nebo čtyři. Pokud byla ale nemoc vyhodnocena jako riziková, tak součástí konsolidační léčby může být pouze jeden udržovací cyklus následovaný transplantací (EG-EU).

U pacientů vyléčených z aktuální nemoci může nastat událost zvaná relaps, tedy návrat choroby. V pravidelných intervalech jednou za rok, nebo každých 6 měsíců, prochází datamanažeři ústavu všechny pacienty evidované jako žijící a zjišťují jejich stav. Především, zdali stále žijí a zdali se jim choroba vrátila. Pokud ano, zanesou tyto informace do tabulky pacientů. Pokud by pacient v úseku od poslední kontroly zemřel, zjišťují se údaje okolnosti smrti. Například souvislost s nemocí (EV-FJ).

Další datové sloupce počínající FK jsou již nadbytečná a do nového systému na žádost zadavatele nebudou uváděna.

1.3.3 Reziduální nemoci

Reziduální nemoc je vyjádřena jako četnost choroby na molekulárně genetické úrovni přítomností genetické změny, která není normální u zdravého člověka. Pro lepší představu vysvětlím problematiku na příkladu fúzního genu *PML-RARa*. U zdravého člověka je gen PML na chromozomu 15 a gen RARa na chromozomu 17. Při vzniku leukémie se chromozomy ulomí a spojí, buňky se množí a vzniká leukémie. Při měření se tedy zkoumá četnost tohoto fúzního genu *PML-RARa* v zachycených buňkách.

Z takového nálezu vyhodnocujeme molekulární remisi. Dokud existuje měřitelný výskyt fúzního genu, tak nemůžeme prohlásit remisi za molekulární a hrozí návrat nemoci (nemoc je přítomná na molekulární úrovni). Měření probíhá v intervalech po třech měsících po dobu dvou let od ukončení léčby.

Ve své práci se zabývám třemi typy reziduální nemoci a to *PML-RARa*, *AML1-ETO* a *inv.16*. Rozdíly mezi nimi zanedbávám, neboť jsou z datového hlediska zpracovány stejně.

Samotné zpracování popisuje již zmíněný obrázek 1.7. V datech se zaznamená identifikační číslo vzorku (sloupec A), datum odběru materiálu (B) a den od diagnózy nemoci (C). Poté datum měření (E) a následně se provedou tři měření zkoumaného genu (F,G,H) a výsledek se aritmetickým průměrem zapíše do sloupce I. Dále se také provede dvakrát měření referenčního genu *abl* a zapíše průměr (J,K,L). Nakonec se provedou matematické operace pro přepočítání výsledků (M, následně D) a ve sloupci D vidíme normalizovanou hodnotu měření. Validní výstup je také hodnota *neg.* u měření fúzního genu, která znamená, že nemoc nebyla nalezena.

1.4 Problémy současného stavu

V této části popisuji nejzávažnější problémy, které shledávám v současném systému evidence dat.

1.4.1 Přístup k datům

Jedním z největších problémů aktuálního stavu je přístup k aktuálním datům. Soubory existují v několika verzích a jsou spravovány řadou lidí. Pro to, aby došlo ke sjednocení dat, je třeba vynaložit nemalé prostředky a úsilí. Zároveň také lékaři nemusí mít přístup k aktuálním datům a musí dohledávat nejnovější informace o stavu pacienta, což vede na zpomalení jejich práce. Řešením tohoto problému je vytvoření databáze a informačního systému, který umožní přístup k aktuálním datům pro všechny příslušné pracovníky.

1.4.2 Pozvolný úpadek konzistence dat

Před několika lety, kdy vznikal popisovaný systém práce s daty, měl tabulku pacientů pod kontrolou jediný lékař, který udržoval data v konzistentním stavu. Postupem času bylo však potřeba umožnit zápis a modifikaci dat různým laboratořím, což znamená přístup mnoha lidí.

S příchodem dalších editorů souborů a nedodržováním stanovených konvencí zaznamenávání dat vznikla spousta unikátních hodnot, jejichž význam zná pouze autor. Tato nekonzistence opět vede k nutnosti objasňovat význam těchto specifických záznamů.

Tento problém nelze bez dlouhodobé spolupráce s pracovníky ústavu vyřešit. Navrhovaná databáze a systém však mohou zamezit dalšímu rozpadu konzistence dat a to tím, že stanoví přísná pravidla pro zadávání nových hodnot a umožní nápravu záznamů porušující konzistenci.

1.4.3 Omezená funkcionalita

Kromě samotné evidence dat lékaři a výzkumníci potřebují přidanou funkcionalitu, kterou současný systém evidence buď nenabízí, nebo je realizace za pomoci aktuálních prostředků uživatelsky nepřívětivá a komplikovaná.

1.4.3.1 Selektivní přístup a zabezpečení

Kontrola přístupu je fakticky neexistující. Každá osoba vlastní excelové soubory může editovat libovolná data a také dále šířit přístup k datům.

Myšlenka pro zlepšení bezpečnosti je omezit přístup zaměstnanců k těmto datům. Zároveň různé laboratoře editují různé datové rozsahy a fakticky je pro ně zbytek dat z hlediska změny nezajímavý. Proto je vhodné omezit editační přístup daným laboratořím na rozsah pro ně nezbytný. Současně ale musí být zachována možnost pro určité uživatele editovat celý datový rozsah, například vedoucí výzkumné pracovníky.

1.4.3.2 Vizualizace dat

Veškerá data jsou uložena v textové (případně číselné) podobě. Neexistuje v současnosti jednoduchý způsob, jak například vykreslit křivku průběhu nemoci (reziduální nemoci). Taková funkcionalita by mohla napomoci rychlejší analýze dat a tedy nepřímo i diagnóze.

1.4.3.3 Vyhledávání

Jedním z nejčastějších úkonů všech pracovníků přistupujících k datům je vyhledání dat konkrétního pacienta. Data jsou sice seřazena dle identifikačního čísla, nicméně je stále třeba najít konkrétní řádek. Větší problém nastane, když přistupující uživatel nezná identifikační číslo a hledá pacienta dle jména nebo

rodného čísla, což vede na výrazně pomalejší proces identifikace správného řádku.

Kromě vyhledávání specifického pacienta je mnohem větším problémem vyhledat skupinu pacientů dle vlastností jejich léčby nebo diagnózy. Například najít všechny pacienty starší padesáti let s rizikovou prognózou je v současném stavu téměř nemožný úkol.

Navrhovaný systém by měl umožnit rychlé vyhledávání pacientů dle jejich osobních údajů a identifikačního čísla. Celkově tedy zkrátit dobu potřebnou k tomu, aby uživatel našel data konkrétního pacienta. Zároveň také zavést možnost vyhledat podmnožinu pacientů dle zadaných údajů.

1.4.3.4 Statistické údaje

Kromě vyhledávání a editace dat konkrétních pacientů je občasným problémem generovat statistická data nad sledovanými pacienty. Tou nejdůležitější statistikou je křivka přežití pro danou skupinu pacientů.

Statistika *Overall survival* je křivka přežití sledována od události diagnózy do poslední návštěvy pacienta. Přičemž pokles křivky značí úmrtí pacienta (viz obr. 1.5).

Druhou potřebnou statistikou je *Progression free survival*, což je obdobná křivka přežití s tím rozdílem, že začátek křivky je datován od události remise po událost relapsu nebo smrti pacienta.

Obě tyto křivky je vhodné vykreslit nad vyhledanou podmnožinou pacientů, například porovnat křivky přežití různých rizikových, nebo věkových skupin pacientů.

1.4.4 Oddělená data reziduálních nemocí

Dalším z výrazných problémů současného stavu je oddělená evidence tabulky pacientů a tabulek měření reziduálních nemocí. Soubory jsou také zpravovány jinými lidmi ve výrazně odlišném formátu. Zatímco tabulka pacientů zobrazuje souhrnný přehled, tabulky reziduálních nemocí dodávají datům pomyslný třetí rozměr a to vývoj nemoci v čase léčby pacienta.

Požadavkem zadavatele je sloučit tato data takovým způsobem, aby byly společně snadno dosažitelná a zobrazitelná uživateli.

1.5 Vliv GDPR

Obecné nařízení na ochranu osobních údajů neboli GDPR (General Data Protection Regulation) je nařízení Evropského parlamentu a Rady (EU) 2016/679 ze dne 27. dubna 2016, o ochraně fyzických osob v souvislosti se zpracováním osobních údajů a o volném pohybu těchto údajů a o zrušení směrnice 95/46/ES. Účinnost tohoto nařízení začíná platit jednotně na celém území EU 25. května 2018 [6].

Nařízení cílí na firmy, instituce a jednotlivce, kteří zpracovávají nebo shromažďují osobní informace občanů EU. Zároveň také na subjekty mimo EU působící na evropském trhu.

Navrhovaný systém zpracovává velké množství osobních informací pacientů a taktéž dlouhou historii jejich léčby. Proto se v této kapitole zaměřím na to, jaké požadavky přináší nové nařízení na navrhovaný systém.

1.5.1 Záběr této práce

GDPR je rozsáhlé procesní doporučení cílíc především na instituci jako takovou, většina požadavků směřuje na vedení ústavu a úpravu vnitřních procesů organizace. V současné době také stále existuje řada výkladových nejasností. Není smyslem práce detailně popisovat legislativní problematiku v celé své šíři.

Tímto se chci vymezit vůči požadavkům procesním, jako je získání souhlasu se zpracováním osobních údajů, případně oznamovací povinnost bezpečnostního incidentu. Tyto povinnosti padají na provozovatele systému, což já ani implementátoři systému nejsme, tudíž se zaměřím pouze na technické požadavky, které ovlivní návrh systému.

Samotná implementace systému spadá pod přísný dohled orgánů ÚHKT zabývajících se splnění požadavků GDPR. Veškerý text této kapitoly je osobním vyjádřením mé osoby jako autora práce a není schválen zadavatelem. Veškerá doporučení jsou pouze orientační a finální požadavky na implementaci bude určovat zadavatel.

1.5.2 Osobní údaj dle GDPR

Osobní údaje jsou veškeré informace o subjektu údajů, ze kterých lze přímo či nepřímo identifikovat subjekt údajů. Fakticky neexistuje výčet všech osobních údajů, neboť neustále vznikají nové způsoby a technologie jak identifikovat osobu.

Mezi obecné osobní údaje řadíme jméno, pohlaví, věk a datum narození, osobní stav, ale také IP adresu a fotografický záznam. Vzhledem k tomu, že se GDPR vztahuje i na podnikající fyzické osoby, řadíme mezi osobní údaje i tzv. organizační údaje, kterými jsou například e-mailová adresa, telefonní číslo či různé identifikační údaje vydané státem [7].

1.5.3 Citlivé a lékařské údaje

Citlivé osobní údaje jsou speciální kategorií údajů, mezi které mimo jiné patří především informace o zdravotním stavu subjektu. Podléhají přísnějším podmínkám zpracování. Pro jejich zpracování potřebujeme oprávněný zájem. V případě této práce dle zákonné výjimky: *zpracování nezbytné pro účely preventivního nebo pracovního lékařství, pro posouzení pracovních schopností za-*

městnance, lékařské diagnostiky, poskytování zdravotní nebo sociální péče atd [8].

V této práci považuji za citlivé údaje veškeré údaje tabulky pacientů (kromě osobních údajů) a data měření reziduálních nemocí.

1.5.4 Údaje uživatelů a pacientů

V databázi systému jsou uloženy osobní údaje dvou druhů subjektů. Uživatelé systému, tedy zaměstnanci nemocnice, a údaje samotných pacientů. Údaje těchto dvou subjektů jsou zpracovány odlišně za jiným účelem a z hlediska GDPR (a další rozšiřující legislativy) se s nimi nakládá jinak.

Uživatelé mohou bez větších obtíží systém „opustit“, zatímco data pacientů nemohou systém z lékařských důvodů opustit. Přesto však pacienti mají svá práva dle GDPR, kterým musí systém vyhovět.

1.5.5 Faktické požadavky na systém

V tuto chvíli neexistuje žádný praktický zdroj technických požadavků. Pro zpracování požadavků jsem tedy zvolil jeden z článků vykládajících požadavky na vývojáře. Následující text se opírá o výklad nařízení dle článku „GDPR: A Practical Guide For Developers“ [9].

1.5.5.1 Právo být zapomenut - článek 17 GDPR

Jedním ze základních práv uživatelů dle GDPR je *Právo být zapomenut*. Teoreticky to znamená, že systém musí být připraven odstranit veškerá osobní data uživatele na jeho žádost.

Z hlediska dat uživatelů (systému) to znamená požadavek na odstranění osobních údajů uživatele. Jednoduše by tak stačilo implementovat funkcionální odstranění účtu a souběžného vymazání všech údajů o uživateli. Problémem jsou ale reference na uživatelský účet, které vnikly v rámci využívání systému. Konkrétně se jedná o potřebu logovat historické změny v datech. Podrobněji se tomu věnuji v části 1.5.5.4, kde vysvětlím řešení zmíněného problému.

V případě dat pacientů si nemůžeme dovolit vymazat data. Data obsažená v systému jsou relevantní ke zdravotnímu stavu pacienta a mají vysokou výzkumnou hodnotu i desítky let po skončení léčby (případně smrti pacienta). V praxi tedy i na žádost pacienta nelze (z procesně právní stránky věci) vymazat jeho citlivá data, neboť máme oprávněný zájem tato data sledovat. Tento fakt pramení z Vyhlášky č. 98/2012 Sb. Vyhláška o zdravotnické dokumentaci, která se v tomto případě aplikuje namísto GDPR. Tedy přísnější a specifitější zákon je nadřízen GDPR.

Prakticky tedy systém nemusí implementovat mechanismus na odstranění dat pacientů.

1.5.5.2 Právo na přenositelnost dat - článek 20 GDPR

Další právo s přímým vlivem na návrh systému je *Právo na přenositelnost dat*. Právo klade povinnost na systém umožnit všem subjektům dat v systému nahlédnout na jejich zpracovávaná data ve strojově čitelné podobě (formát není dále specifikován). Tento požadavek postihuje stejnou mírou data pacientů i uživatelů.

Z praktického pohledu se jedná o požadavek na provozovatele systému jako instituci celkovou. Tedy může nastat scénář, kdy subjekt dat (v tomto případě uživatel systému i pacient) přijde s požadavkem: „Chci kopii veškerých (o mě sledovaných) dat.“ V takovém případě musí mít organizace nastavené procesy tak, aby neprodleně požadavku vyhověla. To znamená získat data ze všech interních systémů a vše předat najednou.

Z toho plyne požadavek umožnit tuto funkcionalitu jako součást systému, který na žádost „nadríženého systému“ (není v tuto chvíli specifikován), odešle veškerá data vyžádaného subjektu.

1.5.5.3 Právo na nápravu dat - článek 16 GDPR

Právo na nápravu dat znamená povinnost systému umožnit subjektům opravit svá data. Případně s přihlédnutím na způsob zpracování dat, bez zbytečného odkladu obdržet formou dodatečného prohlášení informaci o změně dat.

Pro uživatele systému to znamená jednoduše připravit funkcionalitu na změnu osobních údajů, což bývá dnes běžnou praxí u řady systémů.

U dat pacientů je třeba jiný postup a to manuální oprava dat a následné vydání prohlášení. To vše z důvodu, že pacienti nemají přístup do systému a nemohou tak tuto akci provést. Prakticky tedy pacient zažádá instituci o změnu dat, kterou provede uživatel systému a ten v souladu s vnitřními procesy organizace vygeneruje prohlášení. Dle mého názoru není třeba specifikovat speciální požadavky na systém.

1.5.5.4 Logování přístupu k datům

Poslední praktikou, kterou chci zmínit, je logování přístupu k datům, konkrétněji logování čtecích a editačních operací. Nejedná se sice o konkrétní požadavek GDPR, ale o jakousi doporučovanou nejlepší praxi sledování zodpovědnosti uživatelů při manipulaci s daty. Stejně tak tento požadavek řeší odpovědnost za integritu dat.

Z důvodu implementace logování nemůžu plně vyhovět požadavku uživatelů na vymazání osobních dat ze systému, stanoveného výše 1.5.5.1. Představím situaci na příkladu, kdy uživatel opustí svoje zaměstnání. Jeho přístup do systému bude odstraněn společně s jeho daty a veškeré záznamy logu, kde je evidován jako autor nějaké změny, budou anonymizovány či odstraněny. Po uplynutí několika let vznesení pacient evidovaný v systému požadavek na přezkoumání jeho léčby. V tuto chvíli je nutné dohledat, kdo a kdy změnil určitá

data, a to i změny provedené uživatelem, který již dlouhou dobu nepracuje v ÚHK. V tomto případě potřebuji identifikovat „vymazaného uživatele“, tedy musím si uchovat osobní data.

1.6 Funkční požadavky na systém

V předchozím textu kapitoly jsem popsal současný stav práce s daty a problémy, které přináší. Na základě mé předchozí analýzy a dalších požadavků zadavatele vznikla následující funkční specifikace požadavků na systém.

1.6.1 Autentizace a registrace uživatelů

- Každý uživatel systému se před prvním přihlášením registruje. Registrace bude po uživateli požadovat celé jméno (pro účely identifikace ostatními uživateli), kontaktní (pracovní) email, uživatelské jméno a heslo.
 - Heslo musí být alespoň 8 znaků dlouhé, obsahovat alespoň jedno velké písmeno, jedno malé písmeno a jednu číslici.
- Uživatel se před vstupem do systému přihlásí svým uživatelským jménem a heslem.
- Systém umožní každému registrovanému uživateli upravit jeho údaje.
- Uživatel bude pozván do systému unikátním odkazem obsahujícím registrační token, obdrženým formou emailu. Platnost této pozvánky do systému bude nastavena na 30 dní. Poté bude pozvaný uživatel informován o skončení platnosti a znehodnocení odkazu.
- Kdykoliv v průběhu doby platnosti může administrátor zrušit platnost odkazu.

1.6.2 Zobrazení dat

- Systém umožní zobrazit data každého pacienta aktuálně evidovaného v tabulce pacientů. Oddělí vizuálně data různých laboratoří a umožní editaci dle příslušných omezení uživatelských rolí.
- Systém dále zobrazí data měření reziduálních nemocí formou tabulky. Dojde tedy ke spojení informací z tabulky pacienta a tabulky měření reziduálních nemocí.
- V systému bude možno příslušným uživatelům zobrazit seznam ostatních uživatelů a možnost pozvat nové uživatele do systému.

1.6.3 Uživatelské role

- Čtenář
 - Může nahlížet do celého systému kromě správy uživatelů.
 - Nemůže upravovat jakákoliv data.
 - Uživatel laboratoře
 - Přebírá veškerá práva role „Čtenář“.
 - Může navíc editovat data pacientů příslušné laboratoře.
- Specifické datové rozsahy laboratoří z tabulky pacientů:

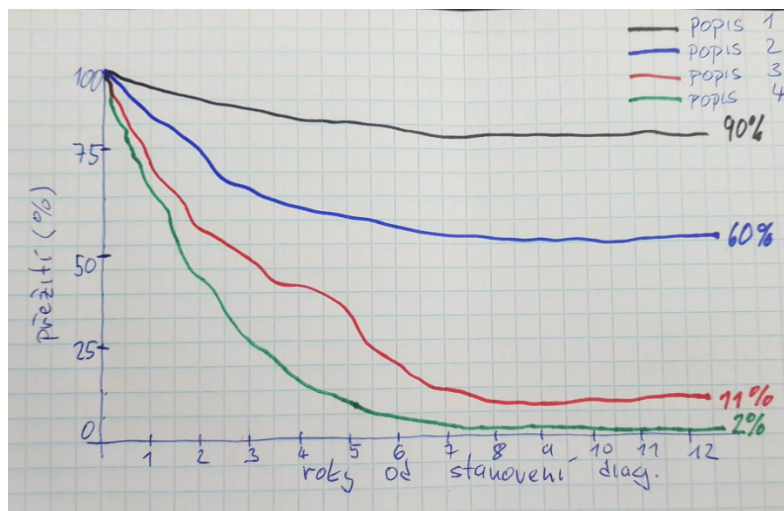
Název rozsahu (nebo laboratoře)	od	do
Obecné údaje	A	U
Krevní obraz a kostní dřeň	V	AD
Průtoková cytometrie	AH	AV
Cytogenetická laboratoř	AW	AX
1. molekulární laboratoř	AZ	DA
2. molekulární laboratoř	DB	DP
Klinické sledování	DT	konec

- Uživatel (globální)
 - Přebírá veškerá práva role „Uživatel laboratoře“.
 - Může navíc editovat veškerá data pacientů v systému.
- Administrátor
 - Přebírá veškerá práva role „Uživatel“.
 - Může zobrazit seznam aktuálně registrovaných uživatelů
 - Může editovat (případně odebrat) přístup uživatelům nižší role.
 - Může pozvat nového uživatele do systému.
 - Může zrušit platnost vystavené pozvánky (registračního tokenu), tedy zneplatnit odkaz na registraci

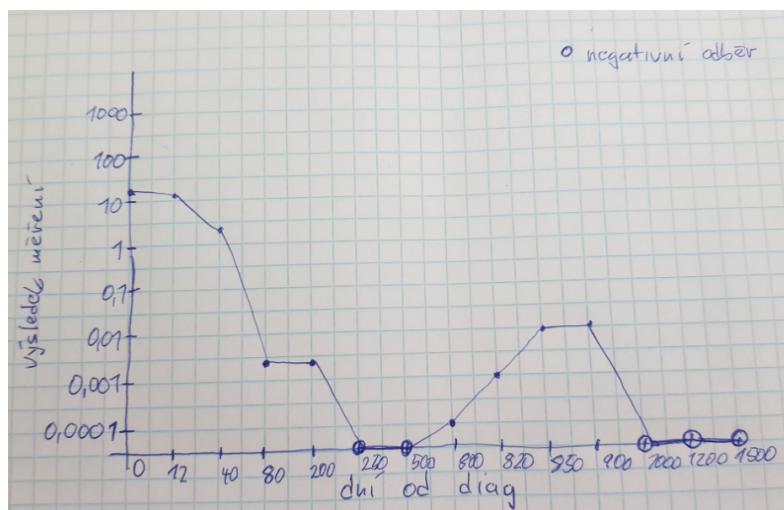
1.6.4 Vyhledávání

- Systém umožní uživatelům vyhledat pacienta dle identifikačního čísla, jména, rodného čísla, data narození a data diagnózy. Následně vybráním zobrazit jeho data.
- Systém umožní vyhledat podmnožinu pacientů dle věku (zadaného intervalu čísel), FAB kódu (typu AML), počtu leukocytů (interval čísel) a dle prognózy výběrem jedné nebo více možností. Systém umožní libovolně kombinovat tyto parametry vyhledávání.

1. ANALÝZA



Obrázek 1.9: Nákres implementace grafu přežití

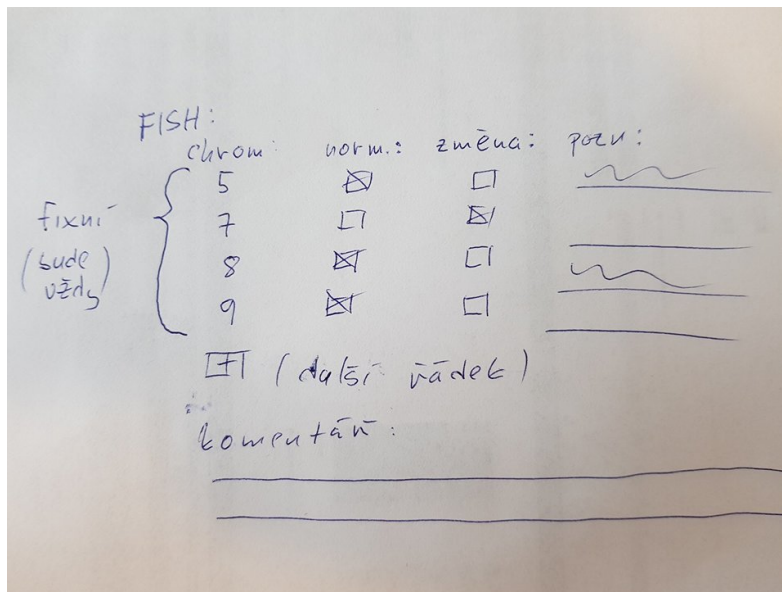


Obrázek 1.10: Nákres implementace grafu měření reziduální nemoci

- Systém umožní exportovat seznam vyhledaných pacientů (identifikačního čísla, jména, rodného čísla, data narození, data diagnózy) do Excelové tabulky.

1.6.5 Vizualizace dat

- Systém umožní zobrazit data měření reziduálních nemocí v jednoduchém grafu. Horizontální osa grafu bude vyobrazovat dny od diagnózy, vertikální osa bude zobrazovat naměřenou hodnotu počtu fúzních genů.



Obrázek 1.11: Návrh datového zpracování cytogenetické metody FISH

Výsledek negativního měření bude odlišen. Vizuálně bude graf odpovídat nákrese na obrázku 1.10.

- Systém dále umožní vykreslit statistické grafy přežití *Overall survive* a *Progression free survive* libovolné vyhledatelné podmnožiny pacientů. Současně také do jednoho grafu vykreslit až 4 různé křivky různých skupin pacientů odlišené barvou a popisem křivky. Vizuální podoba grafu bude odpovídat nákrese 1.9. Detailní popis grafů je vysvětlený v sekci Statistické údaje 1.4.3.4. Pokud není stanoveno jinak, grafem přežití se myslí oba grafy přežití ve zbytku práce.

1.6.6 Další požadavky zadavatele

- Změnit evidenci datové položky FISH (sloupec AX viz obr. 1.8). V současném stavu uživatel vypíše informace o zkoumaných chromozomech do textového řádku. Požadavek zadavatele spočívá v tom, rozdělit tento údaj na 4 samostatné záznamy jednotlivých chromozomů, u každého možnost zaškrtnout políčko s informací, zdali je chromozom normální či změněný. Současně ke každému z chromozomů napsat textovou poznámku. Dále je také požadovaná možnost zadat další chromozom stejného datového tvaru. V případě abnormálního nálezu ponechat pouze textové pole pro přesnější popis. Návrh domluvený se zadavatelem popisuje obrázek 1.11.

- Zadavatel dále požaduje funkcionalitu „potvrzení pacienta“ v podobě jednoduchého tlačítka, kterým uživatel s právy "Uživatel (globální)" a vyšší označí daného pacienta za zkontrolovaného. Tímto si lékaři ověří správnost převedených dat požadovaného pacienta do nového systému. Pacient zůstane označen za zkontrolovaného. Potvrzení lze odvolat libovolným uživatelem systému pro případ, že byla nalezena chyba, kterou potvrzující uživatel přehlédl.
- Vyhledávání nezkontrolovaných pacientů.

1.6.7 Požadavky dle GDPR

- Na základě požadavku uživatele systému a dalšího souhlasu nadřízené osoby (proces souhlasu není součástí specifikace), systém musí umožnit uživatelům role Administrátor vyvolat částečné odstranění uživatelského účtu uživatele.
 - Uživatelé budou odstraněny přístupová práva do systému.
 - Uživatel nebude zobrazen v přehledu uživatelů.
 - Veškeré reference na tento účet budou zachovány z důvodu potřeby archivace změn (potřeba logování)
- Systém umožní uživatelům skrytou funkcionalitu exportování osobních dat uživatelů a veškerých citlivých (včetně osobních) dat pacientů. V současné době není specifikován formát předání dat. Systém bude umožňovat programově vyhledat uživatele dle pracovní emailové adresy a pacienta dle rodného čísla a získat veškerá data o něm uložena. Výstup nebude součástí uživatelského rozhraní, bude sloužit jiným systémům pro sloučení veškerých vyhledávání.

1.6.8 Požadavky dle dalších legislativních norem

- Systém zaznamená v tabulce změn (logu) přístupným uživatelům následující události:
 - Přístup k osobním a citlivým datům pacienta (čtení)
 - Úprava osobních a citlivých údajů pacienta a jeho léčby (tvorba, editace, mazání)
- Systém bude evidovat záznamy o změnách v následujícím formátu:
 - Identifikace uživatele provádějícího změny
 - Identifikace pacienta, jehož data podléhají změně
 - Časová známka úpravy
 - Druh operace s daty (vytvoření, změna, smazání)

- Stará hodnota dat
- Nová hodnota dat
- Identifikace databázové tabulky a datového sloupce, nad kterými změna proběhla

1.7 Nefunkční požadavky na systém

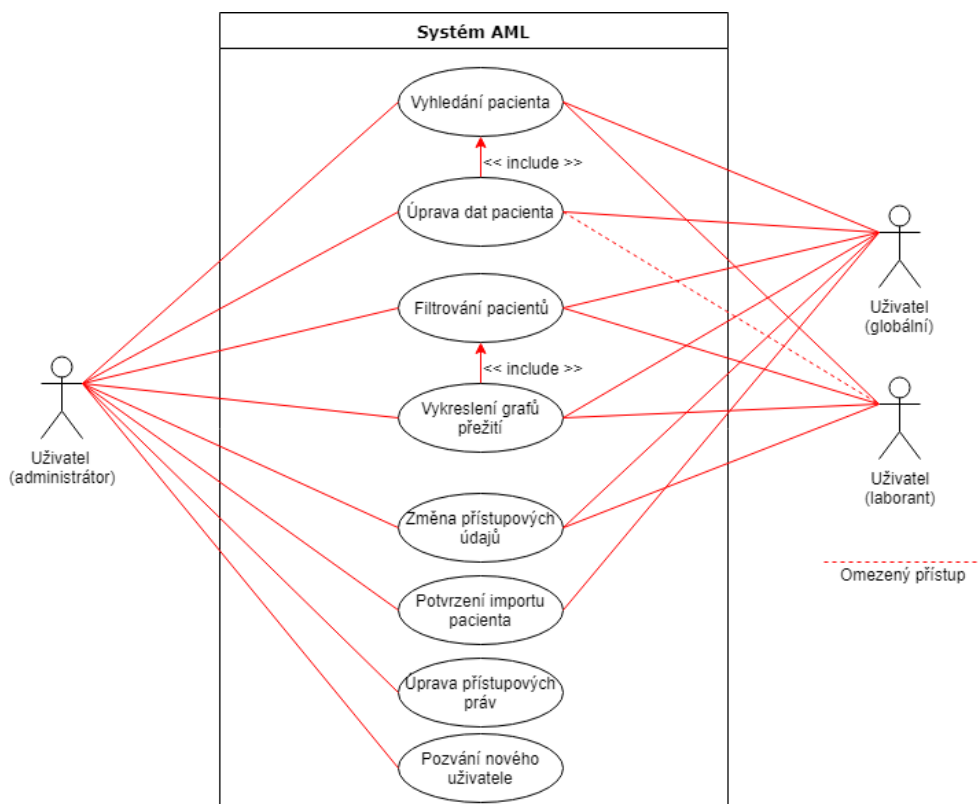
Nefunkční požadavky jsou doplněním funkčních požadavků, popisující vlastnosti systému vzhledem k prostředí, ve kterém bude systém provozován a užíván. V této sekci popisují základní nefunkční požadavky na dostupnost, zabezpečení, výkon a škálovatelnost. Současně podotýkám, že já ani vývojový tým nejsme provozovatelé systému. Tato zodpovědnost padá na IT oddělení ÚHKT.

1.7.1 Dostupnost

- Systém bude dostupný jako webová aplikace. Bude podporovat aktuální verzi prohlížeče Google Chrome. Podpora jiných prohlížečů není zadavatelem vyžadována (v případě problémů bude nainstalován prohlížeč Chrome v aktuální verzi).
- Systém bude podporovat responzivní návrh uživatelského rozhraní pro stolní počítače s rozlišením 1024x768px, zobrazené na uhlopříčce obrazovky 15 palců a větší. Podpora mobilních zařízení (tabletů a telefonů) není požadována.
- Systém bude dostupný shodně s dostupností vnitřní sítě ÚHKT, za jejíž běh odpovídá IT oddělení ústavu. Výjimkou jsou období odstávky z důvodu přechodu na novější verzi systému (případně databáze), kdy systém bude nedostupný maximálně 24h.

1.7.2 Zabezpečení

- Systém bude přístupný pouze zaměstnancům ÚHKT k tomu pověřených. Pověření spravuje zadavatel (MUDr. Mgr. Cyril Šálek, Ph.D., nebo jím pověřená osoba).
- Systém nesmí být viditelný mimo interní síť ÚHKT. Přístup do interní sítě je spravovaný IT oddělením ústavu.
- Systém musí podporovat přístup přes virtuální privátní síť (VPN) ÚHKT. Jinými slovy, uživatelé pracující z domova se vzdáleným přístupem do nemocniční sítě přes VPN musí být schopni naplno využívat systém.



Obrázek 1.12: Diagram případů užití

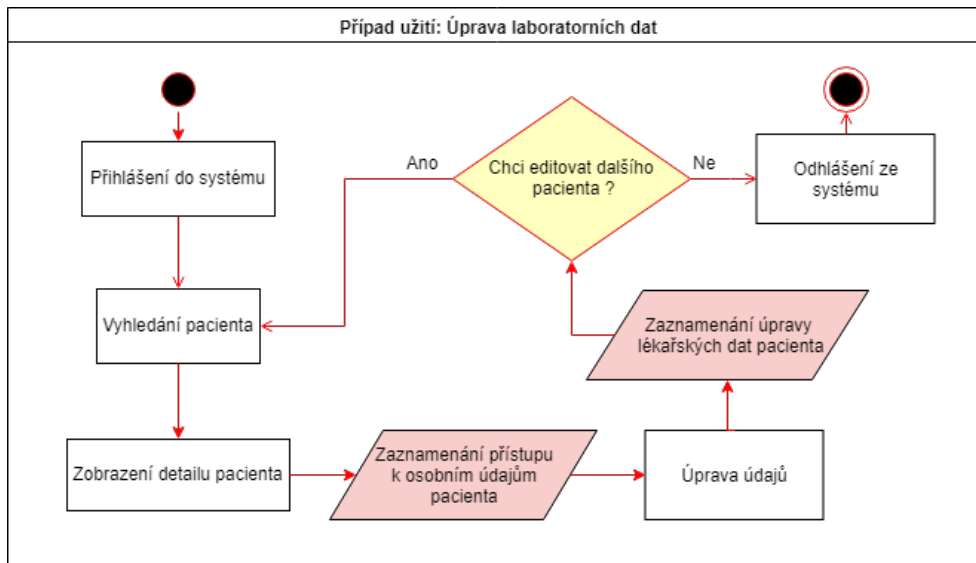
1.7.3 Výkon, škálovatelnost

- Veškeré přechody mezi stavy systému (kompletní načtení webové stránky) nepřesáhnou dobu 1,5 vteřin. Výjimkou tohoto pravidla je načtení stránky s výsledky vyhledávání a filtrování. V tomto případě bude veškerá interakce se systémem zablokována, dokud nedojde k plnému načtení stránky, nebo uživatel nepřeruší načítání.
- Systém bude programově optimalizovaný pro evidenci 1500 pacientů. Hardwarovou optimalizaci provádí IT oddělení ústavu.

1.8 Případy užití

Diagram případů užití reprezentuje uživatelské interakce se systémem formou vztahů mezi uživatelem a jednotlivými případy užití. Diagram 1.12 reprezentuje základní případy užití a jejich souvislost s různými uživateli.

V následujících příkladech blíže představím na třech ukázkách každodenní scénáře, které budou uživatelé vykonávat při práci se systémem.



Obrázek 1.13: Diagram průchodu scénářem: Úprava dat pacienta

1.8.1 Úprava dat pacienta

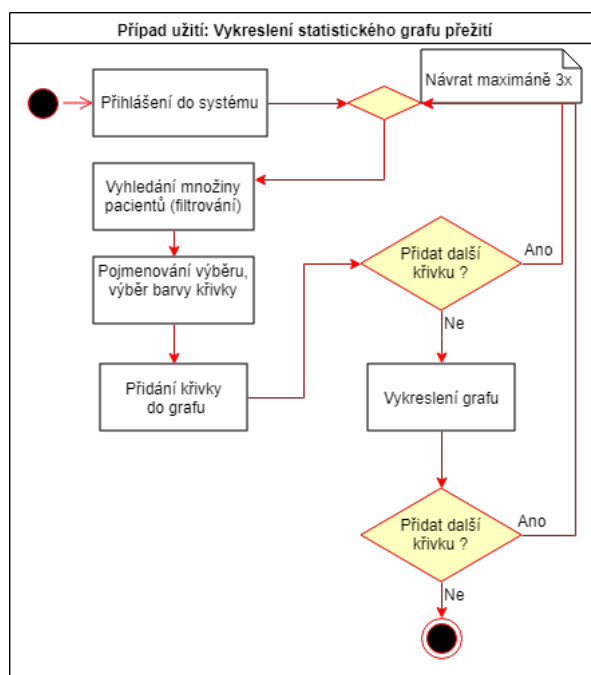
Nejčastějším úkonem uživatelů v systému bude editace dat. Některé uživatelské role budou mít omezený přístup k datům. Diagram 1.13 ilustruje základní scénář editace dat pacienta.

Uživatel začne celý scénář přihlášením dle svého uživatelského jména a hesla. Následně vyhledá konkrétního pacienta dle identifikačního čísla, případně jiných unikátních osobních údajů. V systému si uživatel následně zobrazí detailní přehled daného pacienta a vybere možnost editovat příslušné údaje. Systém zaznamená do logu události přístup k osobním údajům pacienta. Po dokončení úpravy dat se uživatel odhlásí ze systému, případně vyhledá dalšího pacienta pro editaci, pokud chce upravit záznamy více pacientům při jednom přístupu do systému. Při každém uložení změny údajů pacienta dojde rovněž k zaznamenání této události do logu.

1.8.2 Vykreslení statistického grafu přežití

Druhý scénář, ilustrovaný na obrázku 1.14, popisuje funkcionalitu vykreslování statistických grafů přežití.

Přihlášený uživatel libovolného oprávnění vyhledá (případně vyfiltruje) množinu pacientů ze systému. Tuto množinu následně pro popis osy grafu pojmenuje, například „Pacienti 50+“, a vybere barvu křivky. Následně může uživatel graf vykreslit, případně vyhledat novou skupinu pacientů a opakovat proces, dokud bude mít nedefinované všechny požadované křivky. Kdykoliv po vykreslení grafu může uživatel znovu definovat další křivku a nechat graf



Obrázek 1.14: Diagram průchodu scénářem: Vykreslení grafu přežití

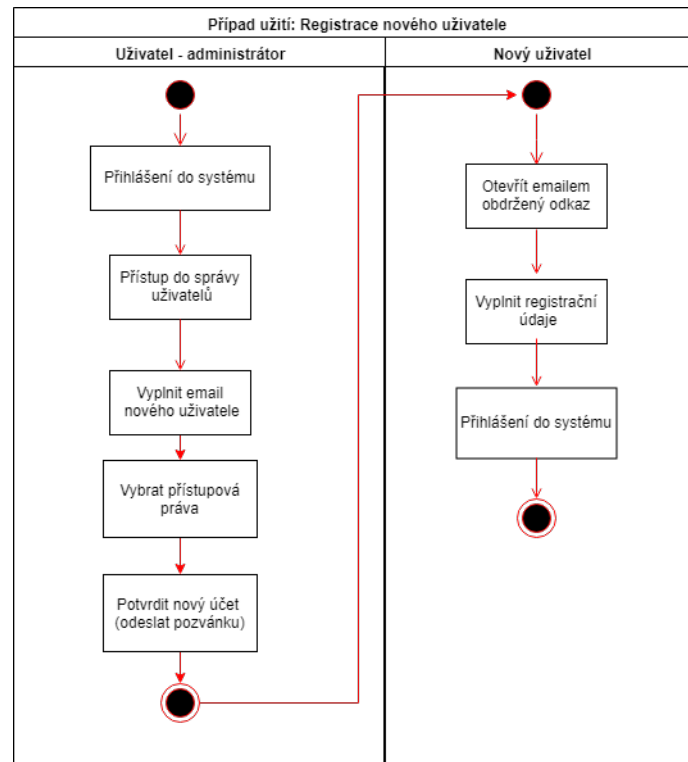
překreslit. Je na vůli každého uživatele, jestli si křivky nejdříve nadefinuje a vykreslí hromadně, nebo bude křivky do vykresleného grafu přidávat postupně. Celkově může uživatel do jednoho grafu vykreslit maximálně 4 různé křivky přežití.

1.8.3 Založení uživatelského účtu

Třetí scénář na obrázku 1.15 popisuje postup založení nového uživatelského účtu. Tuto akci může iniciovat pouze uživatel s právy *administrátor*.

Uživatel se po přihlášení naviguje do sekce správy uživatelů. Zde vyplní emailovou adresu, na kterou bude zaslána pozvánka s unikátním odkazem obsahujícím registrační token. Je na zodpovědnosti uživatele zvát do systému osoby k tomu oprávněné. Pozvánka bude platná pouze 30 dní. Po uplynutí této doby bude na zadanou emailovou adresu odesláno oznámení o propadnutí platnosti pozvánky.

Zvaný (budoucí) uživatel započne svůj scénář v emailové schránce, kde otevře zasláný odkaz do systému. Tento odkaz bude směřovat na registrační obrazovku, kde vyplní svoje registrační údaje: celé jméno (pro účely identifikace ostatními uživateli), kontaktní (pracovní) email, uživatelské jméno a heslo. Následně bude přesměrován systémem na přihlašovací obrazovku, kde se může již přihlásit jako registrovaný uživatel s (administrátorem přidělenými) příslušnými právy.



Obrázek 1.15: Diagram průchodu scénářem: Založení uživatelského účtu

1.8.4 Ostatní případy užití

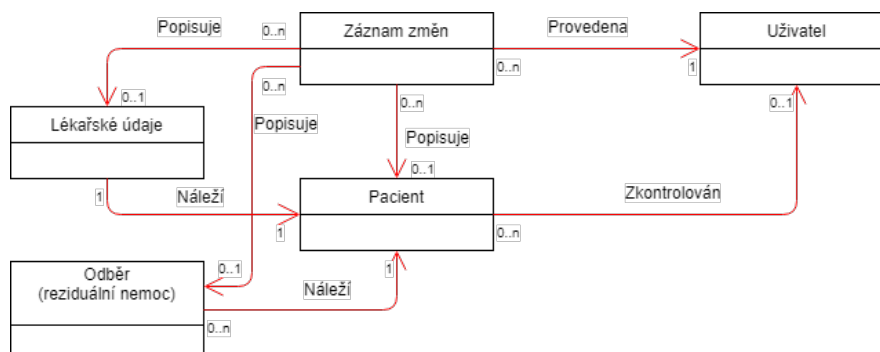
Změna přístupových údajů popisuje případ užití, kdy uživatel s rolí *Administrátor* upraví přístupová práva libovolnému uživateli s nižšími přístupovými právy, případně odebere přístup uživateli.

Potvrzení importu pacienta znamená jednoduché stisknutí tlačítka z pohledu uživatele, kterým potvrdí data pacienta za správně importovaná z excelových dokumentů. Zároveň může libovolný uživatel toto potvrzení odvolat, pokud nalezne chybu, která byla přehlédnuta.

Změna přístupových údajů značí případ užití, kdy libovolný uživatel systému vyvolá možnost změny přístupových údajů. Tím jsou myšleny osobní údaje uživatele (jméno, mail) a přístupové údaje (heslo a uživatelské jméno).

1.9 Doménový model

Cílem této kapitoly je navrhnout doménový model, který definuje a popisuje vztahy mezi základními entitami navrhovaného systému. Navržený model 1.16 zobecňují na několik základních nejdůležitější entit, jejich atributy budu blíže popisovat v následující kapitole věnující-se návrhu databáze.



Obrázek 1.16: Doménový model

1.9.1 Pacient

Ústřední entitou modelu je *Pacient* reprezentující osobní a identifikační údaje pacienta. Tato entita je oddělena od lékařských údajů, neboť v řadě častých aplikací (například výpis pacientů) nepotřebujeme znát detailní informace o léčbě pacienta. Zároveň v případě budoucího rozvoje systému a případné potřeby anonymizovat údaje, navrhuji osobní údaje oddělit od lékařských.

1.9.2 Lékařské údaje a údaje odběrů

Entita *Lékařské údaje* je zjednodušením tabulky pacientů. V následném návrhu databáze se rozpadne do více specializovaných tabulek databáze.

Tabulky měření reziduálních nemocí reprezentuje entita *Odběr*. Pacient nemusí mít asociované naměřené výsledky, proto vazba není povinná ze strany odběru.

1.9.3 Uživatel

Entita *Uživatel* popisuje uživatele přistupujícího do systému. Současně také uživatel může být pozvaný a nemít zatím práva pro přístup.

1.9.4 Záznam změn

Záznam změn reprezentuje log, do kterého se zaznamenávají přístupy k osobním údajům pacienta a změny lékařských (citlivých) dat. Každá změna musí být jednoznačně identifikována uživatelem, který ji provedl, pacientem, který je cílem změny a daty, které jsou součástí změny. Přičemž sledovaným změnám podléhají data osobní, lékařská a data odběrů (měření) reziduálních nemocí.

Návrh relační databáze

Cílem této kapitoly je popsat postup návrhu relační databáze, která bude sloužit pro uložení dat pacientů.

2.1 Požadavky na databázi a transformaci dat

Návrh databáze je klíčový úkol v rámci návrhu celého systému. Data o pacientech, jejich diagnóze a léčbě jsou nejcennějším aspektem systému. Samotná existence databáze řeší jeden ze základních problémů a to je společný přístup k aktuálním datům. V této části se věnuji dalším nárokům na databázi a transformaci dat z excelové podoby s tím spojenou.

2.1.1 Zavedení pojmů

Problematiku této kapitoly vysvětluji za pomoci několika pojmů používaných v teorii návrhu databáze. Některé z nich nepoužívám ve standardním významu, proto je před začátkem popisu návrhu definuji.

2.1.1.1 Integrita a čistota dat

Porušením *integrity dat* v čase nazývám proces, kdy uživatelé vytváří nové nadbytečné datové sloupce a nedodržují stanovenou strukturu dat v tabulce pacientů.

Pojmem *znečištění dat* popisuje existenci nechtěných hodnot v datových sloupcích. Například textové popisky v číselných sloupcích.

Tyto dva pojmy v textu často zaměňuji a považuji za synonyma.

2.1.1.2 Normalizace dat

Normalizací se myslí proces nápravy dat jejich přeorganizováním tak, abych dosáhl efektivnějšího uložení (například odstranění opakujících-se dat), vyhledávání (budováním databázových indexů) a zpracovávání.

2.1.2 Úplnost dat

Při návrhu databáze je nezbytné myslet na základní požadavek, kterým je úplnost převedených dat. Není přijatelné navrhnout databázový model, který nezvládne uložit všechna data aktuálně shromažďovaná ústavem, neboť by mohlo dojít ke ztrátě cenných dat. Cílem návrhu je připravit databázový model, který zohlední aktuální denormalizovaný stav dat a umožní jejich částečnou normalizaci, avšak nezpůsobí ztrátu informací.

Představím situaci na praktickém příkladu, kdy různí zaměstnanci vyplňují sloupec nazvaný *leukocity*. Prakticky se jedná o počet bílých krvinek a z pohledu datové analýzy bych zde čekal celočíselnou hodnotu výsledku, případně prázdné pole, pokud nedošlo k vyhodnocení. Fakticky je ale situace výrazně komplikovanější. Ve většině případů sice naleznou vyplněnou číselnou hodnotu, poté ale sloupec obsahuje řadu dalších hodnot, jejichž význam je jasný pouze autorovi. Při konzultaci se mi dostalo následujícího vysvětlení. Jednou z hodnot je „NT“, zkratka pro *not tested*, tedy případ, kdy vzorek krve byl obdržán, ale nebyl proveden požadovaný test na počet leukocitů. Další vyzorovanou hodnotou je „NA“, zkratka pro *not available*, což znamená, že test byl vyžádán, ale vzorek nebyl k dispozici. Dále mezi výčtem hodnot lze nalézt prázdné pole, které může znamenat, že se nikdo dané problematice nevěnoval, nebo pomlčku, jejíž význam neznám. Případně poznámku laboranta vysvětlující nějakou anomálii, třeba neprůkazné testování.

Těchto hodnot je celá řada a jedna z možností by bylo tyto speciální stavy sloučit, nebo vyčlenit na malé spočetné množství. Při této operaci ale hrozí riziko ztráty důležité informace omezením počtu stavů, nebo jejich sloučením, což je nepřijatelné. Obzvláště když tento proces normalizace provádím jako člověk neznalý lékařské problematiky, kterou data popisují.

Z tohoto důvodu jsem se rozhodl nezasahovat do sledovaných dat jakýmkoliv způsobem, který by mohl pozměnit jejich význam.

2.1.3 Normalizace dat

Dalším ze základních požadavků na nový systém je přinést řád do procesu evidence dat, jinými slovy data normalizovat. Jelikož jsem si v předchozím požadavku stanovil požadavek na úplnost dat, nemohu tedy měnit současná data, ale mohu připravit prostředí, které zamezí budoucímu úpadku integrity dat a stanovit nová integritní omezení.

Základním nástrojem pro normalizaci dat je vyčlenění číselníkových hodnot. Číselníkové hodnoty identifikuji jako opakující-se hodnoty v datech, které vyčlením do speciální tabulky. Tímto přeskupením (nikoliv změnou) dat dosáhnu dvou důležitých vylepšení, které jsem si předsevzal. Prvním vylepšením je normalizace dat, především však zamezení další denormalizaci (znečištění dat). Pokud uživatel uvidí omezený výběr číselníkových (normalizovaných) hodnot, bude pro něj snazší stanovenou konvenci dodržovat a nedojde tak

často k definování zbytečných nových hodnot. Zároveň také eliminuje problematiku budoucích překlepů a nejednotného dodržování velkých a malých písmen v určitých hodnotách.

Druhým vylepšením, které mi číselníkové hodnoty umožní, je snazší vyhledávání v datech. V současném stavu vyhledávání znamená sekvenčně skenovat stovky záznamů a hledat v nich shodu, většinou navíc textovou schodu s vyhledávaným řetězcem, případně podřetězcem. Číselníková hodnota ve vlastní tabulce obsahuje vlastní klíč, který bude použit jako cizí klíč v původní tabulce, což výrazným způsobem přispěje ke zrychlení vyhledávacího procesu.

2.1.4 Odstranění redundantních datových sloupců

Zdrojový soubor tabulky pacientů obsahuje řadu sloupců obsahujících vypočtené hodnoty z jiných datových sloupců. Například důležitý údaj „věk při dg.“ lze snadno dopočítat z data diagnózy a data narození, přičemž oba údaje jsou pro každého pacienta povinné a ve všech případech vyplněné. Z toho důvodu nebudu takový sloupec uvažovat pro modelování v databázi a jeho výpočet bude zajišťovat systém.

2.2 Konvence modelování

Před samotným návrhem modelu si stanovím konvence modelování, které budu následně dodržovat při celém postupu.

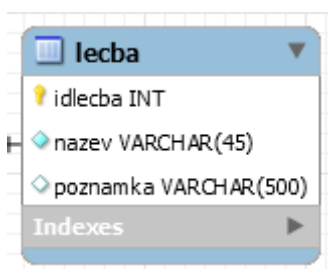
2.2.1 České názvy

Používání českého jazyka vnímám jako negativní fenomén v technické dokumentaci, či přímo v programovém kódu, neboť výrazným způsobem ztěžuje rozšířitelnost (a pochopitelnost) aplikace programátory neovládajících český jazyk.

Přesto jsem se však rozhodl modelovat databázi za užití českých názvů, používaných v excelových datech, neboť tím zamezím případům, kdy nedovedu daný pojem přeložit, nebo neznám jeho skutečný význam z důvodu nedostatečné kvalifikace v lékařském oboru. Zároveň také usnadním práci programátorům zabývajícím se transformací dat, kteří budou mít jednodušší orientaci v databázovém modelu vycházejícím z aktuálního stavu česky evidovaných dat.

2.2.2 Podoba s excelovými daty

Dalším z mých cílů při modelování databáze je usnadnit následující vývoj systému a převod dat z excelových tabulek. Programátor pracující na převodu dat by měl být schopen se rychle zorientovat v modelu databáze a osvojit si konvenci mapování dat. Z toho důvodu jsem se rozhodl názvy datových sloupců databáze co nejvíce přiblížit aktuálnímu názvosloví. Pokud je tomu



Obrázek 2.1: Ukázka číselníkové tabulky

možné, název sloupce ze souboru převedu na název sloupce databázové tabulky tak, že odstraním diakritiku, převedu veškerý text na malá písmena a mezery nahradím podtržítkem `_`. V případě, že je název sloupce příliš dlouhý nebo komplikovaný, tak jej zkrátím, případně vynechávám slova.

Dalším mechanismem, který v návrhu implementuji je zachování pořadí dat z excelových souborů. Jakmile programátor nalezne sloupec databáze korepondující se sloupcem excelové tabulky, může se spolehnout na to, že všechny následující sloupce databázové tabulky sledují pořadí sloupců tabulky ze souboru.

2.2.3 Struktura číselníkových tabulek

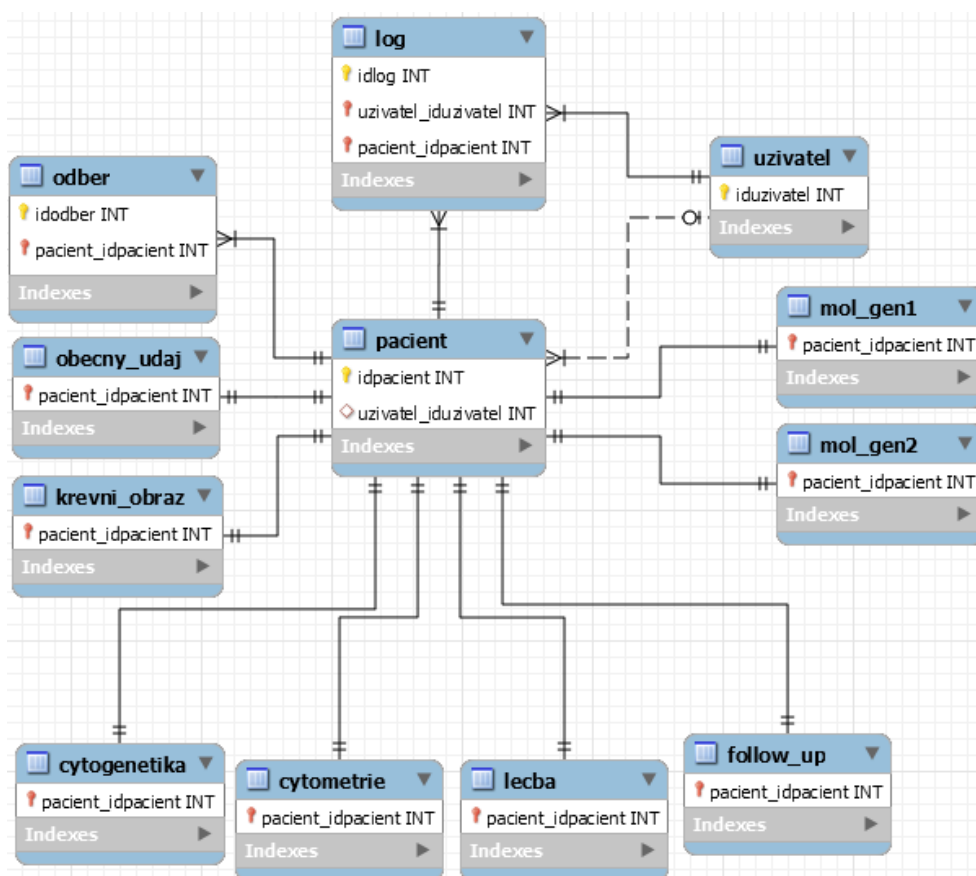
V případě nutnosti vytvořit číselníkovou tabulku budu dodržovat konvenci ilustrovanou obrázkem 2.1. Každá tabulka bude mít povinný atribut *nazev* s maximální délkou 45 znaků a volitelnou poznámku (*poznamka*) s maximální délkou 500 znaků. Tímto chci docílit snadnější manipulaci s tabulkami v programovém kódu a umožnit systematický přístup k tabulkám se stejnou strukturou.

2.3 Rozdělení dat do tabulek databáze

Databázový model na obrázku 2.2 vyobrazuje základní rozvržení tabulek s jejich identifikátory. Jedná se o jednoduchou implementaci doménového modelu do databázového prostředí.

2.3.1 Rozdělení tabulky pacientů

Největším rozdílem oproti doménovému modelu je rozdělení entity *Lékařské údaje* (reprezentující tabulku pacientů) na databázové tabulky *obecny_udaj*, *krevni_obraz*, *cytogenetika*, *cytometrie*, *mol_gen1*, *mol_gen2*, *lecba* a *follow_up*. Důvodem rozdělení je rychlejší přístup k datům jedné tabulky a přehlednější model databáze. Každá z těchto tabulek sleduje datový rozsah rozdílných laboratoří stanovených ve funkční specifikaci 1.6.



Obrázek 2.2: Zjednodušený databázový model

Výjimkou je rozdělení datového rozsahu *Klinické sledování* do tabulek *lecba* a *follow_up*. Důvodem rozdělení je rozdílná frekvence a období přístupu k datům. Zatímco data o léčbě jsou frekventovaně čtena a editována v relativně krátkém časovém úseku léčby pacienta, *follow_up* obsahuje data sledovaná po skončení léčby, které se přistupují mnohem vzácněji, avšak jsou editována roky po skončení léčby.

Všechny tyto tabulky mají identifikující relaci 1:1 vůči tabulce *pacient*, což znamená, že by se všechny zmíněné tabulky daly sloučit do jedné. Ve své implementaci jsem se ale rozhodl data od sebe oddělit, neboť nejčastější operací bude přístup k datům právě jedné z těchto tabulek a úprava jejich dat příslušnou laboratoří. Nevýhodou mého postupu je však vyšší cena dotazů nad všemi daty současně, což přináší drahou operaci spojování tabulek.

2.3.2 Tabulka log

Tabulka obsahuje složený klíč ze tří atributů: identifikátoru uživatele, pacienta a dále vlastní klíč. V doménovém modelu je vyznačena relace i k lékařským datům pacienta, kterou jsem se rozhodl neřešit na databázové úrovni. Tuto zodpovědnost přebere datová vrstva architektury systému, která bude identifikovat konkrétní datové záznamy a tabulky analýzou databázových dotazů. Domnívám se, že implementování na databázové úrovni by bylo z relačního hlediska zbytečně složité.

2.3.3 Ostatní tabulky

Zbytek tabulek sleduje doménový model. Tabulka *odber* obsahuje složený klíč, neboť je v relaci n:1 vůči tabulce *pacient*. Tedy jeden pacient bude mít vícero záznamů odběru. Relace mezi tabulkou *pacient* a *uzivatel* je neidentifikující (přerušované spojení) a nepovinná (prázdné kolečko u tabulky *uzivatel*), neboť relace značí „potvrzení karty pacientem“, což nemusí nastat. Zároveň také platí, že jeden uživatel může potvrdit několik pacientů.

2.4 Rozbor datových typů

Rozbor datových typů znamená rozhodnout, jakým způsobem budou data z excelových tabulek neprezentována v podobě datových typů a tabulek relační databáze. Mým cílem bylo vymyslet systematický postup, který mohu aplikovat na každý sledovaný údaj, tedy sloupec tabulky.

2.4.1 Identifikace dat

Prvním krokem k vybudování datové struktury je zjistit, co se skutečně nachází v předaných datech. Obecně to lze v jednodušších případech zjistit pohledem na data, ale jelikož je tabulka pacientů rozsáhlá a obsahuje řádově stovky záznamů, bylo potřeba vytvořit nástroj, který pomůže s analýzou. K tomuto účelu jsem vytvořil program pro analýzu dat (dále označován také jako program).

2.4.2 Program pro analýzu dat

Samotný program obdrží na vstupu excelový soubor a v nastavení programu určí uživatel obdélníkovou oblast dat, ve kterých program provede své vyhodnocení. Program vyhodnotí každý sloupec dat a vypíše unikátní hodnoty a počet jejich výskytů v daném sloupci, případně data agreguje podle předaného regulárního výrazu (například datum).

Obrázek 2.3) ukazuje, jak vypadá konzolový výstup běhu programu pro sloupec *Indukce*. První řádek výstupu obsahuje informaci o zkoumaném sloupci. Následně každý další řádek začíná číslem indikujícím multiplicitu výskytu

```
Column DU (Indukce):
275: L Ida+AC100 3+7
159: L DNR 90+AC100 3+7
120: NULL
59: L ICE
47: L paliace
29: L DNR+SD-AC 3+7
21: L AIDA (Pethema LPA05)
19: L HAM
18: L paliace LD-AC
16: L DNR+SD-AC 4+7
13: L AIDA
13: L mini-ICE
11: L AIDA (EORTC)
10: L Ida+AC100 2+5
9: L AIDA (Pethema LPA99)
9: L Ø
8: L MICE
7: L ATRA
6: L paliace HU
6: L PETHEMA LPA 2005
6: L mini-MICE
6: L HD AML12
6: L DCE
5: L LD-AC
5: L ano, nespecif.
4: L NT
4: L LPA 2005 (PETHEMA)
4: L Vidáza
3: L Ida (reduk. d. 25%) + ATRA
3: L Indukce 3+7
3: L 5Azac
3: L FLA-Ida
3: L DNR 90/AC 100 3+7
2: L paliace LD-AC, HU
2: L DNR 60/AC 100 3+7
2: L Pethema 2005
2: L HU
2: L paliace LD-AC od 19.10.2016
2: L DNR+ SD-AC 3+7 + 6TG
2: L Ida (reduk. d. 50%) + ATRA
2: L DNR 60+AC100 3+7
2: L ATRA (Pethema 2005)
2: L LPA 2005 /“(PETHEMA)
1: L DNR+SD-AC 2+7
1: L DNR 90/AC 100 3+5
1: L DNR+SD-AC 2+8
1: L Vidáza od 4.5.2015, LD-AC od 25.8.2015
1: L k 17.1.02 neléčena
1: L PETHEMA
1: L Ida+AC100 3+7 (red.na 2+5)
1: L AC100 5
```

Obrázek 2.3: Ukázka výstupu analytického programu

hodnoty. Další informací na řádku je značka „L“, nebo „R“ indikující, zdali se jedná o literál, či regulární výraz. Alternativou je označení „NULL“ pro nevyplněný řádek. Následuje výpis samotné hodnoty.

2.4.2.1 Algoritmus analýzy dat

Program sekvenčně projde každý záznam ve sloupci a pokusí se jej uložit do unikátní kolekce `hashmap`¹. V případě, že program najde duplicitní záznam, inkrementuje čítač výskytu daného záznamu. Pro každý sloupec dat, lze také definovat regulární výraz, dle kterého algoritmus bude vnímat data jako duplicitní. Například stanovíme regulární výraz pro celá kladná čísla a každé takové různé číslo bude program vyhodnocovat jako opakovaný výskyt v daném sloupci, což mi pomůže například odhalit textové hodnoty (poznámky lékařů) v datovém sloupci, kde bych očekával striktně číselné hodnoty.

2.5 Databázové datové typy

Pro své modelování jsem se rozhodl vybrat systém MySQL². Jedná se o jeden z nejpoužívanějších open-source databázových relačních strojů [10]. Ve své práci jsem se dále rozhodl používat jednoduché datové typy, které jsou dostupné v každém podobném databázovém stroji. Dále pro dosažení maximální přenositelnosti kódu nespecifikuji číselné datové přesnosti typů *TINYINT*, *FLOAT* a *DOUBLE*.

Pro tvorbu databázového modelu jsem se rozhodl využít nástroj MySQL Workbench³.

2.6 Modelování dat tabulky pacientů

V následujícím textu se zabývám transformací tabulky pacientů do databázové podoby. Každý sloupec tabulky pacientů jsem analyzoval za pomoci mého programu a zařadil mezi jeden z následujících datových typů. Každý z definovaných typů má svoji databázovou reprezentaci a jasně stanovená pravidla pro převod dat z tabulky pacientů. Ve své definici kladu důraz na zachování podoby dat a snažím se dosáhnout maximální normalizace.

2.6.1 Jednoduché skalární datové typy

Nejjednodušším datovým typem z hlediska převodu na databázový model jsou skalární datové typy: číslo, desetinné číslo a textový řetězec.

¹<https://docs.oracle.com/javase/8/docs/api/java/util/HashMap.html>

²<https://www.mysql.com/>

³<https://www.mysql.com/products/workbench/>

2.6.1.1 Textové hodnoty

Textové hodnoty obsažené ve sloupcích tabulky pacientů, které neobsahují žádné opakující-se hodnoty, které by podléhaly požadavku na normalizaci, převádím na databázový datový typ řetězec. Zástupcem takové skupiny hodnot je třeba sloupec *Příjmení a jméno* z tabulky pacientů.

Pro tyto hodnoty jsem zvolil databázovou reprezentaci typem *VARCHAR*. Jedná se o typ s proměnnou délkou, přičemž při jeho definování určuji maximální délku. To si mohu dovolit vzhledem k tomu, že znám veškerá převáděná data a díky jejich dlouholetému sledování dovedu určit maximální délku uloženou v datovém sloupci, která nebude s vysokou pravděpodobností překročena. Na rozdíl od datového typu *CHAR*, který jako pole znaků je také definován s maximální délkou, kterou v případě potřeby doplní do definované délky, *VARCHAR* využije pouze potřebnou délku. Nevýhodou je pomalejší přístup [11].

2.6.1.2 Číselné hodnoty

Podobně jako unikátní textové hodnoty, tabulka pacientů obsahuje řadu číselných hodnot, které nemá význam slučovat. Jedná se o hodnoty, které jsou výsledkem měření, například sloupec *Počet leukocytů*, nebo sloupce reprezentující výsledky měření.

Celočíselné hodnoty reprezentuji typem *INT*. Desetinné hodnoty číslem s plovoucí desetinou čárkou *FLOAT*. Desetinná čísla v tabulce pacientů jsou zapsána s maximální přesností na tři desetinná místa.

2.6.2 Rozhodovací typy

Dalším častým sloupcem z tabulky pacientů jsou takzvaná „rozhodovací data“. Jedná se o sloupce, jejichž výčet hodnot v ideálním případě obsahuje hodnoty 1 a 0. V takto čisté podobě je v tabulce pacientů evidován pouze sloupec *Mužské pohlaví*.

Většina rozhodovacích datových sloupců v tabulce pacientů obsahuje velké množství „znečišťujících“ hodnot (poznámky autorů) a speciálních stavů, jako jsou hodnoty *NT*, *???* (a různé variace počtu otazníků), *NA* a podobné. Tyto data nemůžu sloučit, jak jsem již popisoval v sekci 2.1.2, proto musím vymyslet normalizující způsob, kterým data uchovám v databázové podobě.

2.6.2.1 Rozhodovací typy s konstantami

Jednodušší ze dvou variant převodu jsou rozhodovací sloupce obsahující opakující-se data z podmnožiny definované tabulkou 2.1. Tato tabulka obsahuje časté nestandardní hodnoty, které převádím na číselné konstanty. Datové sloupce z tabulky pacientů tohoto typu tudíž mohu převést na datový typ *TINYINT*, který dle dokumentace MySQL v jednom bytu dat uloží číselný rozsah -128 až

Tabulka 2.1: Tabulka převodních konstant

Hodnota v tabulce pacientů	Nová databázová hodnota
1	1
0	0
[nevyplněno]	null
[hodnota v poznámce]	-1
NT	-2
NA	-3
? (více opakování)	-4
nelze hodnotit, NH	-5
neued., neued, neuedeno, ND	-6

```

Column U (Extramedulární postižení):
523: L 0
401: NULL
87: L 1
2: L nelze hodnotit
1: L NT
1: L ?

```

Obrázek 2.4: Datová analýza sloupce *Extramedulární postižení*

127 [11], což je ideální datový typ pro uložení rozhodovací hodnoty a malého množství definovaných konstant.

Veškeré konstanty uvádím v záporném rozsahu čísel, neboť v tabulce pacientů není datový sloupec, který by evidoval záporná čísla, tudíž nehrozí konflikt se skutečnými daty.

Jedním z příkladů datového sloupce tohoto typu je *Extramedulární postižení*, který obsahuje data popsaná obrázkem 2.4.

2.6.2.2 Rozhodovací typy s komentáři

O něco málo komplikovanější typ jsou sloupce obsahující také komentáře autorů. Jedním z takových datových sloupců tabulky pacientů je *Předchozí chemo- nebo radioterapie*. Obrázek 2.5 popisuje výčet hodnot, ve kterém můžeme vidět, že se někteří výzkumníci rozhodli do sloupce zapisovat i textové poznámky o typu předchozí terapie mimo (silně převažující) konvenci zápisu jedniček a nul.

Převod takového sloupce vyřeším tak, že hodnoty vyhovující tabulce konstant 2.1 převedu na číselné hodnoty podobně, jako v předchozím bodu. Pro ostatní (většinou unikátní) hodnoty vytvořím nový sloupec v databázové tabulce pojmenovaný stejně, jako číselná reprezentace s koncovkou „_pozn“ a do sloupce s konstantami zapíšu hodnotu -1(hodnota v poznámce. Databá-

```

Column R (Předchozí chemo- nebo radioterapie):
741: NULL
250: L 0
18: L 1
4: L RA
1: L CHE,RA
1: L AKTINOT.
1: L FC, FCR

```

Obrázek 2.5: Datová analýza sloupce *Předchozí chemo- nebo radioterapie*

```

| ◆ pred_chem TINYINT |
| ◆ pred_chem_pozn VARCHAR(500) |

```

Obrázek 2.6: Databázová implementace sloupce *Předchozí chemo- nebo radioterapie*

Tabulka 2.2: Tabulka převodních konstant pro typ datum

Hodnota v tabulce pacientů	Nová databázová hodnota
[validní datum]	[validní datum]
[nevyplněno]	null
[hodnota v poznámce]	1000-01-01
NT	1000-01-02
NA	1000-01-03
? (víceero opakování)	1000-01-04
nelze hodnotit, NH	1000-01-05
neued., neued, neuedeno, ND	1000-01-06

zovou implementaci datového sloupce tabulky pacientů znázorňuje obrázek 2.6.

2.6.3 Typ datum

Tabulka pacientů obsahuje několik datových sloupců obsahujících informace o datu určité události, například datum narození pacienta, nebo datum chemoterapie. K těmto datům se stavím obdobně, jako k rozhodovacím typům. Sloupce obsahující validní hodnoty data a známé konstanty převedu na datový typ DATE dle tabulky 2.2. V případě, že sloupec obsahuje další poznámky, vytvořím v databázi další sloupec s poznámkou a zapíšu příslušnou konstantu značící hodnotu v poznámce.

Dle specifikace datového typu DATE, je v MySQL garantovaný rozsah „1000-01-01“ až „9999-12-31“. Je rozumné předpokládat, že žádný pacient se nenarodil před tisíci lety, tudíž mi rozsah dává velký prostor pro vyčlenění

několika hodnot pro známé konstanty.

2.6.3.1 Užití programových konstant

Důležitým návrhovým rozhodnutím je užití programových konstant, neboť konstanty pro výše popsané rozhodovací typy neukládám do vlastní databázové tabulky, ale počítám s jejich interpretací v programovém kódu systému.

Pro vyčlenění hodnot do samostatné tabulky shledávám důvod pouze v případě, že se očekává editace daným záznamů, jejich přidávání a odebírání. V případě těchto konstant chci však zamezit tvorbě dalších hodnot a v ideálním případě omezit jejich aktuální počet. Danou konstantu můžu vymazat pouze v případě, že se již nikde v systému nebude vyskytovat, což je dlouhý proces uživatelské normalizace, který ani nemusí proběhnout.

Současně také do budoucího vývoje systému navrhuji možnost danou konstantu nahradit programově jinou (například otazníky hodnotou NT), případně některé konstanty úplně odstranit pod přísným dohledem odborného pracovníka, čímž celkově napomůžu vyčistění a normalizaci dat.

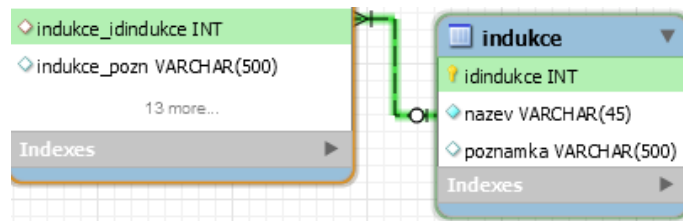
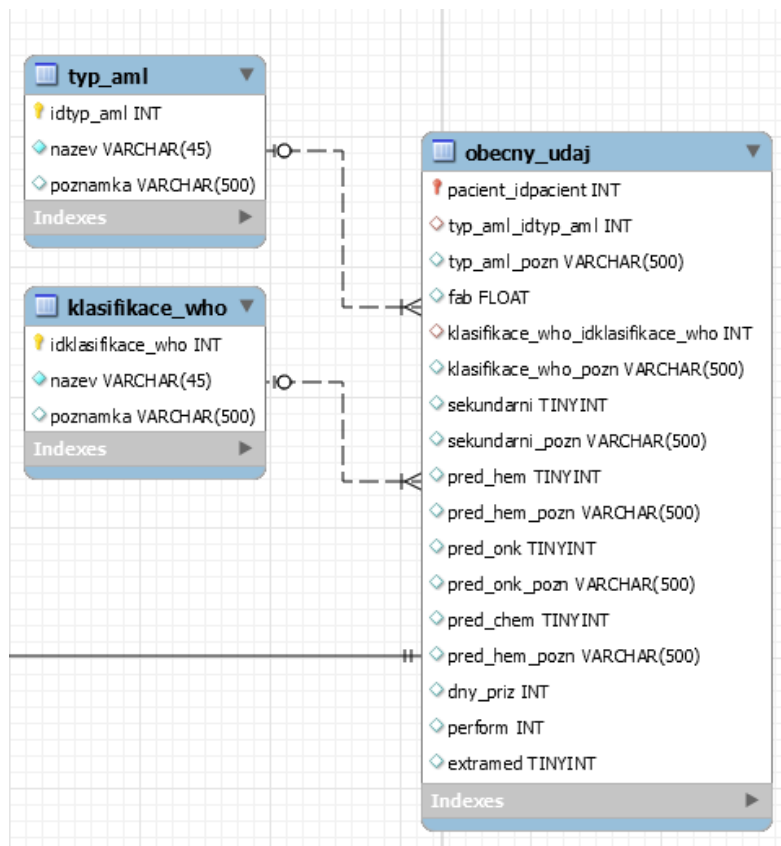
2.6.4 Hybridní číselníkové typy

Nejzajímavějším aspektem návrhu databáze jsou datové sloupce obsahující textové hodnoty, které nemohu jednoznačně vyčlenit do číselníkové tabulky, neboť obsahují velké množství unikátních záznamů. Takové sloupce, které obsahují řadu opakujících se hodnot při velkém počtu opakování a zároveň velké množství unikátních hodnot, nazývám hybridní číselníkové typy.

Takové sloupce historicky začaly jako číselníkové hodnoty, ale postupem času došlo k deformaci dat vlivem mnoha editorů, kteří postupně začali zavádět nové hodnoty a zapisovat unikátní poznámky právě do takového sloupce. Vzhledem k požadavku na úplnost dat není v mé kompetenci tato data „čistit“, proto jsem vymyslel přístup, jak dosáhnout částečné normalizace.

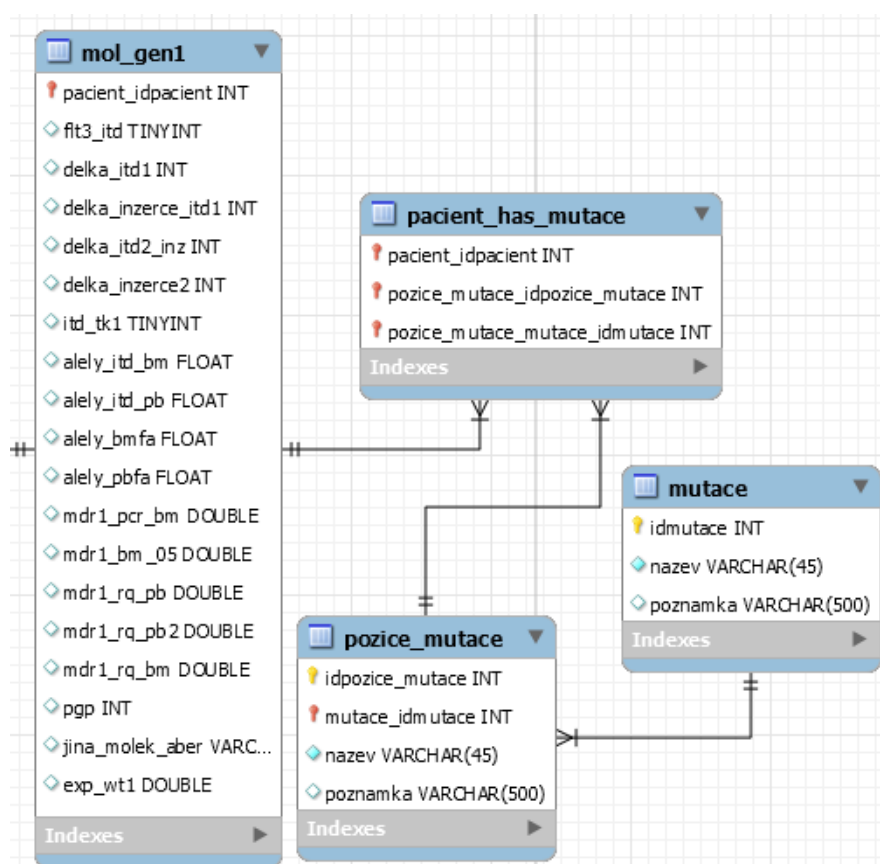
Převod dat vysvětlím na následujícím příkladu pro sloupec DU nazvaný *Indukce*. Na obrázku 2.3 lze vidět výstup z programu pro analýzu dat. Z prvních řádků výstupu lze vidět, že sloupec obsahuje řadu opakujících se hodnot, pro které by bylo vhodné vytvořit číselník v rámci normalizace. Dále ale vidíme také hodnoty, které jsou ve sloupci použity pouze jednou. Zde podotýkám, že výpis není kompletní a obsahuje další desítky hodnot, které nejsou na obrázku vidět. Pro tyto hodnoty se domnívám, že nemá smysl vytvářet číselníkovou hodnotu, neboť se s největší pravděpodobností jedná o unikátní případy, anomálie, nebo prostě poznámky zanesené lékaři.

Pro podobné datové sloupce, které obsahují takto částečně normalizovatelná data, jsem se rozhodl aplikovat následující reprezentaci demonstrovanou obrázkem 2.7. Každá hodnota, která se ve výpisu programu zobrazí alespoň dvakrát, je vyčleněna jako číselníková hodnota a při převodu dat bude zapsána do příslušné číselníkové tabulky *indukce*. Pro všechny ostatní hodnoty bude

Obrázek 2.7: Databázová implementace sloupce *Indukce*Obrázek 2.8: Databázová tabulka *obecny_udaj*

zapsána do číselníkové tabulky hodnota „jiné“ a jejich skutečná hodnota bude zapsána do pole poznámky.

Tento způsob dosáhne částečné normalizace vyčleněním nejpoužívanějších hodnot do vlastní tabulky a zároveň uloží (podobně jako v rozhodovacím případě) i nestandardní hodnoty.



Obrázek 2.9: Databázový model mutací

2.6.5 Ukázka kompletní tabulky

S využitím mnou stanovených pravidel jsem převedl tabulku pacientů do databázové podoby. Obrázek 2.8 ukazuje výslednou tabulku *obdecny_udaj*. Na tabulce můžeme vidět aplikování pravidla pro jednoduché skalární typy (*fab*), rozhodovací typy (*extramed*), rozhodovací typy s komentáři (*sekundarni*) a hybridní číselníkové typy (*typ_aml*).

2.7 Výjimky ze systematického modelování

Přestože jde mnou vymyšlený systematický způsob transformace aplikovat bez potíží na celou tabulku pacientů, rozhodl jsem se přistoupit ke dvěma výjimkám v databázovém modelu.

2.7.0.1 Mutace

První z výjimek jsou data mutací, která jsou součástí datového rozsahu *Molekulární genetiky 1*. Mezi těmito sloupci excelových dat se několikrát opakuje dvojice „mutace“ a „pozice mutace“. Mutace stanovuje název zkoumané mutace a pozice značí pozici mutace v genech pacienta, přičemž možné pozice jsou částečně unikátní (existuje průnik některých podmnožin) pro každou mutaci (vysvětleno lékařem, není výstup analýzy skutečných dat).

Obrázek 2.9 znázorňuje realizaci problému v databázovém modelu. Tabulka *mutace* obsahuje všechny sledované mutace ve formě číselníkové tabulky. Tabulka *mutace_pozice* obsahuje všechny možné pozice a zároveň u každé z nich informaci o mutaci, ke které náleží. Každý pacient má několik pozic mutací a stejně tak každá pozice náleží několika pacientům, tudíž mezi tabulkami existuje relace n:n, kterou dekomponuje tabulka *pacient_has_mutace*.

Při plnění dat do databáze se nejprve naplní tabulka *mutace* všemi sledovanými mutacemi. Následně se u každého pacienta zkontroluje, jestli je mutace přítomná na nějaké pozici a vyplní se tabulka *pozice_mutace* příslušnými daty. Současně vznikne záznam v tabulce *pacient_has_mutace*, tam se zapíše identifikátor pacienta, pozice a samotné mutace.

Tato implementace má svoji výhodu v tom, že při budoucím požadavku na sledování nově nalezených mutací lze jednoduše vytvořit záznam nové mutace a k ní náležících možných pozic, aniž bych musel měnit databázový model.

2.7.0.2 Konsolidace

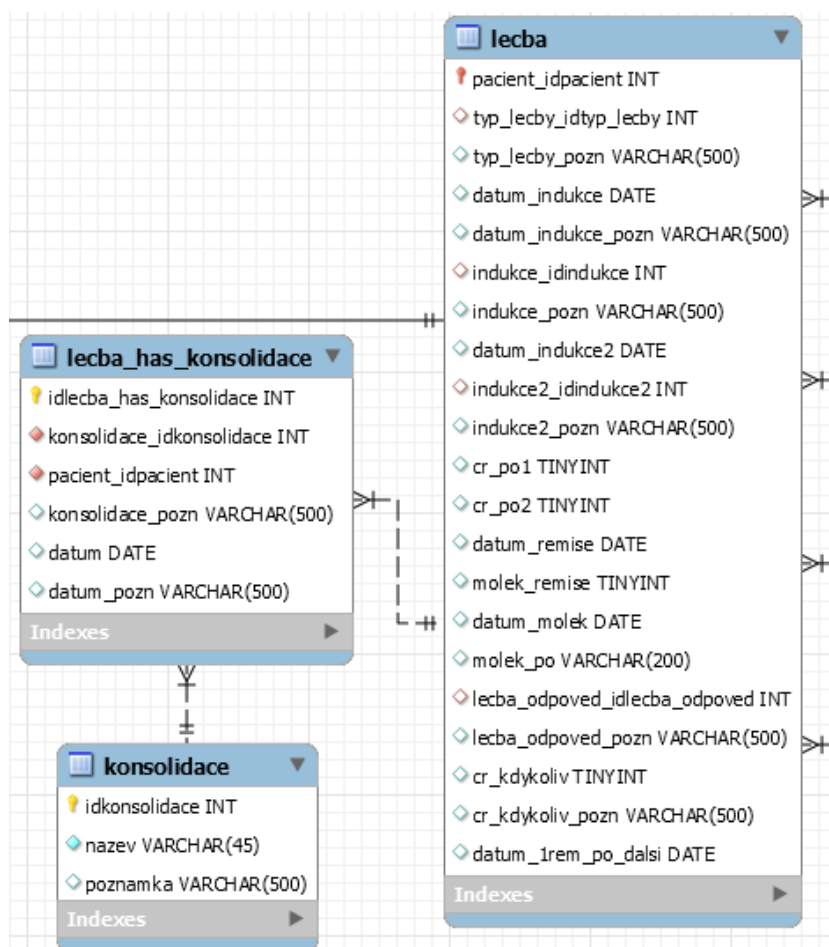
Druhou výjimkou je řešení dat konsolidace, které jsou součástí rozsahu *Klinické sledování*. Jedná se o konsolidační léčbu, například chemoterapii, přičemž každý pacient může projít maximálně šesti iteracemi léčby. Informace se skládají především z data léčby a konkrétního typu léčby.

Řešení na úrovni databáze je jednoduché dekomponování vztahu n:n mezi tabulkami *lecba* a číselníkovou tabulkou *konsolidace*. Dekompozice vztahu je provedena tabulkou *lecba_has_konsolidace* popsanou obrázkem 2.10.

Při převodu dat se nejprve naplní tabulka *konsolidace* všemi možnými hodnotami, následně se u každého pacienta zjistí, kolik měl konsolidačních léčeb a dle toho vyplní tabulka *lecba_has_konsolidace*.

2.8 Modelování další funkcionality

Ostatní tabulky databáze již vyžadují individuální přístup návrhu. Jedná se o implementaci tabulky odběrů reziduálních nemocí, pacienta, uživatele a logu.



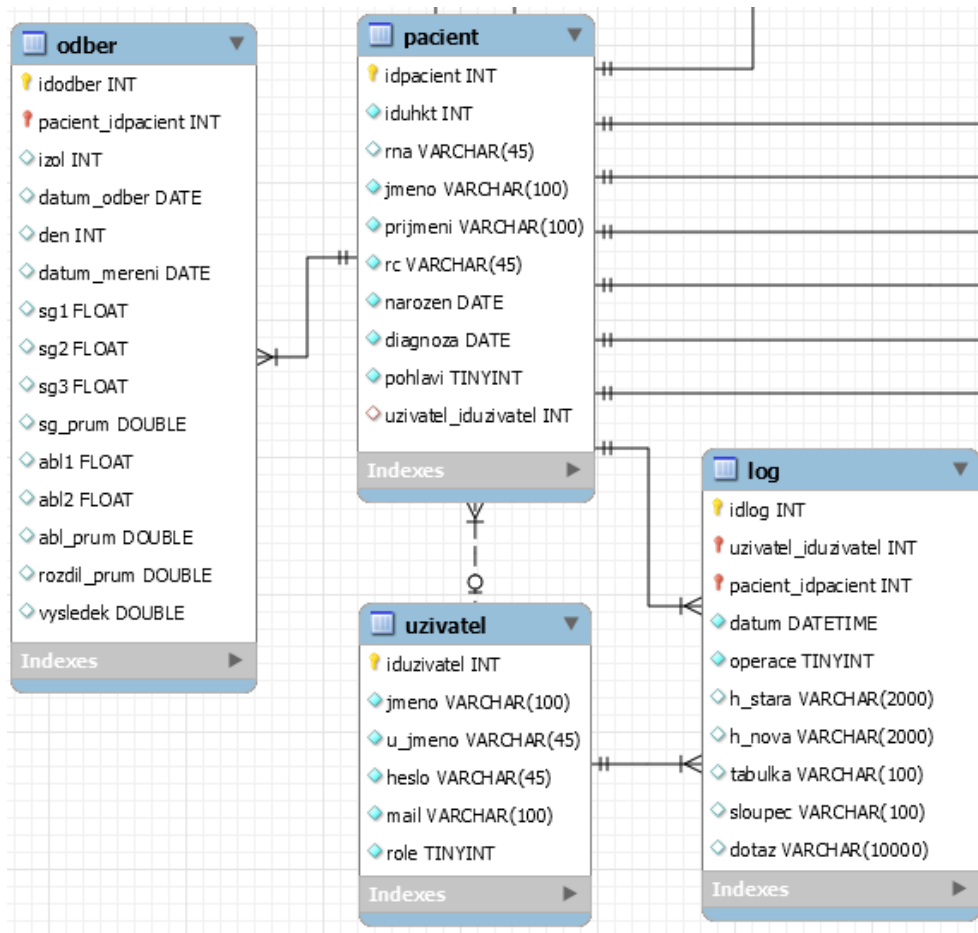
Obrázek 2.10: Databázový model léčby

2.8.1 Tabulka odběrů reziduálních nemocí

Na rozdíl od tabulky pacientů je tabulka měření reziduálních nemocí evidována s větší mírou normalizovanosti, neboť má jediného správce. Přesto však trpí nedostatkem vyplněných dat, které znemožňují omezit ukládané hodnoty na základní naměřené hodnoty potřebné pro výpočet popsany v kapitole 1.3.3. Obrázek 2.12 ukazuje tabulku, ve které chybí vyplněná data odběrů, přesto však existuje výsledná hodnota měření (například řádek 6). Z toho důvodu jsou data v tabulce nepovinná (viz obr. 2.11).

Pro sloupce zaznamenávající naměřené hodnoty genů (*sg1*, *sg2*, *sg3*, *abl1* a *abl2*) jsem zhodnotil jako dostatečný datový typ plovoucí řádové čárky *FLOAT*, který by měl garantovat přesnost minimálně na dvě desetinná místa [11]. Laboranti také nevyplňují žádná další desetinná místa.

Pro ostatní číselné sloupce používám datový typ *DOUBLE*, který posky-



Obrázek 2.11: Výřez databázového modelu

tuje lepší přesnost výpočtu, které je v aritmetických operacích potřeba využít.

2.8.2 Tabulka log

Dle legislativních požadavků popsanych v kapitole 1.6.8 je třeba implementovat mechanismus záznamu přístupu k datům pacienta.

Jednou z implementačních variant, které jsem zvažoval, je použití databázových spouštěčů (triggery). Ty nabízejí pohodlnou funkcionality provádění kódu při vyskytnutí určité události (například selekce z tabulky *pacient*). Bohužel jsem neshledal způsob, jak získat v databázovém stroji informaci o aktuálně přihlášeném uživateli, což je zásadní údaj popisu změny dat. Z toho důvodu jsem se rozhodl funkcionality přenechat datové vrstvě architektury systému a nadefinovat databázovou reprezentaci provedených změn.

Samotná tabulka obsahuje povinně klíčové atributy a údaje o datu a pro-

2. NÁVRH RELAČNÍ DATABÁZE

	A	B	C	D	E	F	G	H	I	J	K	L	M
	Č. izol.	Datum odběru	Den č.	2 (Ct.abl - Ct.SG)	Datum měření	C _t SG PB hodn 1	C _t SG PB hodn 2	C _t SG PB hodn 3	C _t SG PB prům	C _t abl PB hodn 1	C _t abl PB hodn 2	C _t abl PB prům	C _t abl - C _t SG
1													
2	U882	25.06.15	0	0,2597148	14.08.15	30,2	30,23	30,44	30,29	28,18	28,51	28,345	-1,95
3	U883	25.06.15	0	0,0886952	09.08.16	34,13	33,94	33,51	33,86	30,63	30,1	30,365	-3,49
4	V068	12.08.15	48	0,0000971	14.08.15	45,07			45,07	31,59	31,89	31,74	-13,33
5				0,0004556		42,84			42,84	31,59	31,89	31,74	-11,10
6				0,0005527/3					#####			#####	#####
7			48	0,0001840					#####			#####	#####
8	V219	16.09.15	83	#####	17.09.15	neg.			#####			#####	#####
9	V744	08.01.16	197	#####	11.01.16	neg.			#####			#####	#####
10	W193	08.04.16	288	#####	12.04.16	neg.			#####			#####	#####
11	W655	01.07.16	372	0,0002560	07.07.16	42,42	43,44	45,47	43,777	31,94	31,75	31,845	-11,93
12	W712	19.07.16	390	0,0013621/3	20.07.16	43,83			43,83	34,25	34,37	34,31	-9,52
13				0,0004540									
14				0,0001313	20.07.16	45,74			45,74	33,02	32,67	32,845	-12,90
15	W713	19.07.16	390	0,0001818		45,27			45,27	33,02	32,67	32,845	-12,43
16				0,0003131/3					#####			#####	#####
17				0,0001044					#####			#####	#####
18													

Obrázek 2.12: Nekompletní tabulka měření reziduální nemoci

Tabulka 2.3: Tabulka konstant operací logu

Operace	Databázová hodnota
Vytvoření záznamu	1
Úprava záznamu	2
Odstranění záznamu	3
Přístup k datům	4

vedené operaci. Operace je definována jako konstanta, přičemž platí konvence stanovená tabulkou 2.3. Tyto údaje postačí k bezpečnému identifikování operace přístupu k osobním datům pacienta. Stará a nová hodnota (h_stara , h_nova) je reprezentovaná v textové podobě s maximální délkou 2000 znaků, což postačí na uložení nejdelšího textového údaje uloženého v datech pacientů. Číselné hodnoty budou převedeny na text. Atribut *dotaz* slouží pro uložení SQL dotazu. Systém by měl být schopen programově vytěžit veškeré ostatní informace právě ze samotného dotazu, který je ukládán společně s ostatními daty.

2.8.3 Tabulky pacient a uživatel

Ostatní tabulky na obrázku 2.11 jsou implementovány jednoduše dle funkčních požadavků v kapitole 1.6.

Tabulka *pacient* obsahuje atributy potřebné pro identifikaci pacienta, převážně jeho osobní údaje. Většina z nich je povinná, neboť tomu tak je ve zdrojových datech a požaduje se dodržování této konvence.

Tabulka *uzivatel* obsahuje atributy uživatelů systému. Veškerá data jsou povinná pro vznik záznamu.

2.9 Postup převod dat

V posledním bodu kapitoly zabývající se návrhem databáze chci připomenout navržený postup převodu dat z excelových souborů do navržené databáze.

1. Prvním krokem převodu dat je upravení tabulky pacientů vynecháním nepotřebných sloupců. Pokud existuje sloupec v tabulce pacientů a neexistuje s ním korespondující sloupec příslušné databázové tabulky, jedná se o jeden z následujících případů:
 - Sloupec byl zadavatelem vyhodnocen jako nechtěný v novém systému
 - Sloupec obsahoval prázdná data a nebude do budoucna vyžadován
 - Sloupec, respektive data obsažená v něm, je lehce dopočitatelný z ostatních údajů převedených v databázi
2. Následně je třeba vytvořit mechanismus mapování názvů sloupců tabulky pacientů na atributy databázových tabulek. To by mělo ulehčit mnou dodržené pořadí atributů v tabulce, které sleduje pořadí dat v tabulce pacientů a moje intuitivní snaha pojmenovat datové sloupce co nejpodobněji.
3. Dle předepsaných pravidel v sekci 2.6 převést datové sloupce tabulky pacientů.
4. Vyřešit mapování souborů s odběry reziduálních nemocí na korespondující pacienty v tabulce pacientů a následně naplnit tabulku odběrů. Neexistuje žádný způsob mapování jiný, než je název souboru, který je tvořen příjmením pacienta, které obsahuje 4 duplicitní názvy. Tyto kolize je nutné vyřešit osobní konzultací.

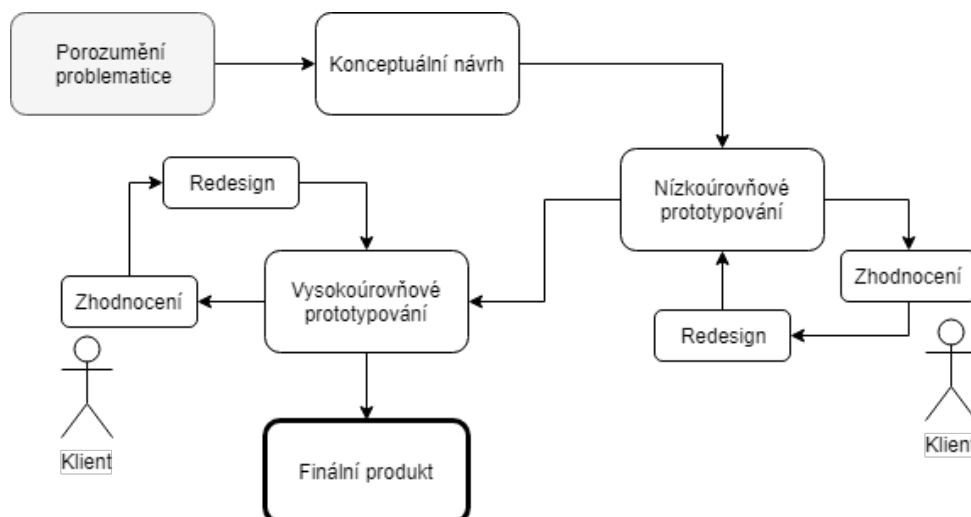
Výslednou správnost převodu dat ručně kontrolují uživatelé. Zkontrolovaný pacient bude mít ve své tabulce zapsáno id kontrolujícího uživatele.

Návrh uživatelského rozhraní

V této kapitole se věnuji návrhu uživatelského rozhraní (dále také zkratkou UI) systému. Cílem této části práce je popsat postup, kterým jsem vytvořil prototyp rozhraní.

3.1 Postup návrhu

Diagram 3.1 znázorňuje obecný proces návrhu UI. Následující text se věnuje podrobnému vysvětlení jednotlivých kroků a jejich implementaci při návrhu UI v mé práci. Postup návrhu vychází z předmětu MI-NUR [12].



Obrázek 3.1: Proces návrhu uživatelského rozhraní

3.1.1 Porozumění problematice

Prvním krokem v návrhu UI je porozumět tomu, co klient potřebuje (jaký problém řešíme). Domnívám se, že jsem tuto problematiku dostatečně popsal v první kapitole, konkrétně specifikováním funkčních požadavků na systém a analýzou případů užití 1.6. Tento krok tedy považuji za splněný a nebudu dále problematiku popisovat.

3.1.2 Konceptuální návrh

V tuto chvíli již vím, čeho chci návrhem docílit. Nyní pokračuji tím, že konceptuálně rozvrhnu rozhraní na atomické části (task list), následně jednotlivé části poskládám do obrazovek (task groups) a nakonec pomocí orientovaného grafu popíši vztah mezi jednotlivými obrazovkami (task graph).

3.1.3 Nízkoúrovňové prototypování

Následujícím krokem návrhu je vyprodukovat prototyp výsledného rozhraní. V této fázi je důležité myslet na to, že se jedná o prvotní návrh, který se bude často a výrazně měnit. Takový prototyp nazývám nízkoúrovňový, často se také užívá názvu *low-fidelity (lo-fi) prototype*, *mock-up* nebo *wireframe*. Prototyp musí být jednoduchý, rychle a levně upravitelný. Při tomto prototypování se neimplementuje žádná finální funkcionální systém, současně také prototyp vzniká na vlastní platformě (papír, pdf soubor, webové rozhraní, prezentace), kde je výrazně jednodušší návrh upravit. Nástroje k tvorbě takového prototypu by měli být snadné k naučení a měly by umožňovat jednoduše produkovat obrazovky a další elementy UI.

Takovým nástrojem může být i tužka a papír, neboť má vysokou ilustrativní hodnotu a každý jej dovede ovládat. Ovšem měnit výraznějším způsobem takový návrh znamená překreslit celé rozhraní, což není časově efektivní. Z toho důvodu jsem se pro svoji práci rozhodl využít specializovaný software Balsamiq⁴, který umožňuje vytvářet prototypy na různých platformách. Program má přednastavené UI elementy pro každou ze zvolených platforem, umožňuje mi provázat vytvořené obrazovky do interaktivních celků (zvaných wireframes) a tento prototyp následně vyexportovat do pdf formátu.

Výsledný návrh poté slouží například pro prezentaci klientovi, kde získáme zpětnou vazbu a můžeme levně a rychle změnit náš návrh. Tento proces (prototyp, zhodnocení, redesign) opakujeme, dokud nepanuje spokojenost s výsledným návrhem. Následně pokračujeme dále k vysokoúrovňovému prototypování.

⁴<https://balsamiq.com/>

Tabulka 3.1: Rozdíl úrovní prototypů UI [12]

Low fidelity	High fidelity
Hodiny/dny na vývoj	Dny/měsíce na vývoj
Mnoho alternativních návrhů	Málo (ideálně žádný) alternativních návrhů
Žádné finální interakce	Zakomponované finální interakce (funkčnost)
Neimplementování na cílové platformě	Implementováno na cílové platformě
Jednoduchý (ruční) náskres	Zakomponován grafický design
	Součást procesu vývoje
	Součást finální implementace celého systému

3.1.4 Vysokoúrovňové prototypování

Nyní již mám dobrou představu o modelovaném rozhraní a můžu přistoupit k tvorbě prototypu na vyšší úrovni (označován také jako high-fidelity prototyp). Na této úrovni již kladu velký důraz na přiblížení realitě. Model vzniká na konkrétní platformě, na které bude systém fungovat. Zároveň také tento model simuluje funkcionalitu výsledného systému. Tabulka 3.1 shrnuje rozdíly mezi popisovanými úrovněmi prototypů. Prototyp je součástí vývoje systému a každá změna je v této fázi vývoje drahá, proto se neočekávají velké změny, především díky použití low-fidelity prototypu.

Prototyp se opět za kolaborace s klientem formuje do své finální podoby. Součástí testování UI je také uživatelské testování. Dokončený vysokoúrovňový model lze vnímat jako finální návrh uživatelského rozhraní systému, čímž tedy končí celý proces popsany v této sekci.

3.1.5 Záběr této práce

Z principu absolventské práce, která má velmi striktní interval vypracování, nelze zachytit iterativní vývoj návrhu uživatelského rozhraní při komunikaci se zadavatelem. Z toho důvodu v této práci popisuji návrh uživatelského rozhraní do bodu vytvoření prvního nízkourovňového prototypu. Domnívám se, že nemá smysl pokračovat s prototypováním vyšší úrovně, dokud nedostanu zpětnou vazbu od zadavatele a nedokončím nízkourovňový prototyp do uspokojivé podoby z hlediska zadavatele. K první prezentaci návrhu rozhraní dojde až po odevzdání této práce.

3.2 Správnost návrhu

Základní otázkou návrhu UI je zjistit míru správnosti návrhu. Rozhraní se může jevit jako subjektivní záležitost každého uživatele, přesto však dovedeme při každodenní práci s různými programy rozhodnout, zdali je návrh jejich UI přívětivý. Optimální návrh je tedy takový, který vyhovuje svým uživatelům co nejvíce. Problémem je však zjistit, co uživatelům vyhovuje a to především, když to sami nevědí (nemají vlastní představu), což je případ návrhu v této práci. Problém jsem se rozhodl vyřešit aplikováním obecně známých a léty prověřených pravidel návrhu a rozbořem obdobných systémů popsanych v následující sekci 3.3. Zkombinováním těchto dvou metod bych měl docílit úspěšného návrhu, který bude pro uživatele intuitivní a snadný na pochopení.

Při svém návrhu se řídím základními pravidly desatera obecných principů interakce Jakoba Nielsena „10 Usability Heuristics for User Interface Design“ [14]. Jedná se o řadu principů často nazývaných heuristik, neboť se jedná o obecná pravidla a doporučení dobrého návrhu UI, které Jakob Nielsen publikoval v roce 1994 jako součást jeho práce „Enhancing the explanatory power of usability heuristics“ [13]. Tyto heuristiky se zpravidla používají pro zhodnocení již navrženého rozhraní a hodnotí se, do jaké míry je každý popisovaný bod splněn. Nicméně popisovaná pravidla lze velmi dobře využít pro samostatný návrh díky jejich obecnosti.

V následujícím textu postupně rozeberu každou heuristiku a pokusím se vyvodit praktické požadavky na návrh uživatelského rozhraní.

3.2.1 Viditelnost stavu systému

Systém by měl vždy informovat uživatele o tom, co se děje, a to prostřednictvím vhodné zpětné vazby v přiměřené době [14].

Z této heuristiky lze vyvodit požadavky na plynulý běh systému. Nemělo by dojít k „zamrznutí“ při časově náročných operacích. Měl by existovat progress bar při dlouhém čekání, nebo animace nahrávání, aby byl uživatel informován, že se děje nějaká událost.

Potenciálně náročnou operací v navrhovaném systému shledávám filtrování a vyhledávání pacientů, což je operace sjednocení a vyhledávání nad velkou skupinou tabulek, která může trvat i jednotky vteřin. Během této doby by měl být uživatel informován, že dochází k vyhledávací operaci. Zároveň bych doporučil možnost přerušit tuto operaci, pokud by mělo vyhledávání trvat několik vteřin a uživatel upřednostňuje jinou akci.

3.2.2 Shoda mezi systémem a reálným světem

Systém by měl mluvit jazykem uživatelů, spíše slovy, frázemi a koncepty známými uživateli než systémově orientovanými výrazy. Dodržujte konvence v reálném světě, aby se informace zobrazovaly v přirozeném a logickém pořadí [14].

Z tohoto bodu heuristiky plyne jednoduchý požadavek na převod systémových hlášek a nápisů do lékařských pojmů, na které jsou uživatelé zvyklí.

3.2.3 Minimální zodpovědnost

Uživatelé často vybírají systémové funkce omylem a budou potřebovat jasně označený „nouzový východ“, aby opustili nechtěný stav, aniž by museli projít rozšířeným dialogem. Podpora vrácení zpět a vpřed (undo and redo)[14].

Systém by měl varovat uživatele před potvrzením nevratné akce. Dále také umožnit rychlý návrat z obrazovek formulářů a mít vždy k dispozici na přehledném místě navigaci do domovské obrazovky. V optimálním případě by měl systém umožnit vrátit zpět nechtěnou operaci.

Nevratnou akcí v systému může být například smazání záznamu odběru a nebo uživatele. V takovém případě bude uživatele varovat dialog s potvrzením akce a možností zrušení. Na veškeré operace manipulující s daty pacientů a uživatelů bude v systému samostatný formulář, takže uživatel musí nejprve kliknout na tlačítko vyvolávající formulář, následně vyplnit data a potvrdit uložení dat. Systém navrhnu tak, aby neexistoval případ, kdy uživatel omylem klikne na tlačítko, které způsobí nevratnou operaci. S přihlédnutím ke všem těmto mechanismům se domnívám, že není nutné implementovat operaci vrácení zpět nechtěné změny.

3.2.4 Shoda s platformou a obecnými standardy

Uživatelé by si neměli klást otázku, zda různá slova, situace nebo akce znamenají totéž. Dodržujte konvence platformy[14].

Systém a případné hlášky budou dodržovat konvence webové platformy. Například stránkou je myšlena webová stránka.

3.2.5 Prevence chyb

Lepší, než smysluplné chybové hlášky, je navrhnout systém tak, aby předcházel vzniku chyb. A to buď eliminací podmínek, které vedou na chybový stav, nebo ověřením těchto podmínek před potvrzením akce uživatele[14].

Prakticky to znamená navrhnout systém tak, aby například formulářové prvky vyžadující číselné hodnoty neumožnily uživateli zadat jiné hodnoty, například textové. Podobně by se měly chovat prvky pro práci s daty (časovými).

Speciální pozornost si zaslouží akce, které by mohly způsobit narušení integrity databáze a vyvolat tím programovou chybu. Příkladem může být odstranění číselníkové hodnoty, která je provázána s daty pacientů. V tomto případě by mělo vymazání hodnoty být zablokováno a informovat uživatele o nutnosti nahradit výskyt těchto hodnot jinými před dalším pokusem o smazání.

3.2.6 Rozpoznávání spíše než pamatování (kouknu a vidím)

Minimalizujte zatížení paměti uživatele tím, že objekty, akce a možnosti budou viditelné. Uživatel by si neměl pamatovat informace z jedné části dialogu na druhou. Pokyny pro používání systému by měly být viditelné nebo snadno dostupné, kdykoli je to potřeba[14].

V navrhovaném systému nepočítám s dlouhými dialogy na několik stran. Nicméně při každé editaci údajů pacienta bude systém zobrazovat jeho jméno, aby bylo evidentní u koho se změny provádí. Systém dále umožní v rozhraní návrat k poslednímu navštívenému pacientovi pro případ, že uživatel potřebuje provést nějakou změnu například v nastavení.

3.2.7 Flexibilita a efektivita užívání

Akcelerátory, neviditelné uživatelem začátečníkem, mohou často urychlit interakce pro odborného uživatele tak, že systém mohou obsluhovat jak nezkušení, tak zkušení uživatelé. Umožněte uživatelům přizpůsobovat časté akce[14].

Všichni uživatelé mnou navrhovaného systému jsou odborníci, kteří budou využívat většinu funkcionality. Vyjímkou je nastavení přístupu uživatelů, které nebude uživatelům s nižšími právy zobrazeno. Uživatelé konkrétních laboratoří zase nebudou mít možnost vyvolat formulář pro editaci dat, které nepřísluší dané laboratoři, ale uvidí veškerý obsah.

3.2.8 Estetika a minimální design

Dialogy by neměly obsahovat informace, které jsou irelevantní nebo zřídka potřebné. Každá další jednotka informací v dialogu soutěží o pozornost uživatele a snižuje jejich relativní viditelnost[14].

Jelikož je systém navrhován na míru a uživatelé jsou odborníci v doméně, předpokládám, že všechny informace jsou relevantní.

3.2.9 Smysluplnost chybových hlášení

Chybová hlášení by měla být vyjádřena v prostém jazyce (žádné kódy), přesně naznačit problém a konstruktivně navrhnout řešení[14].

Prakticky tedy chybová hlášení musí být srozumitelná pro uživatele neznalého informatiky. Tedy namísto chybových kódů budou hlášky obsahovat jasně popsané problémy a jejich řešení.

3.2.10 Dokumentace a nápověda

I když je lepší, pokud lze systém používat bez dokumentace, může být nutné poskytnout pomoc a dokumentaci. Jakékoli takové informace by měly být snadno dohledatelné, zaměřené na úkol uživatele a uvádět konkrétní kroky, které je třeba provést[14].

Systém vzniká na míru pro zadavatele, který bude osobně seznámen s každou funkcionalitou systému a jeho rozhraní. Podobně budou obeznámeni i další klíčoví uživatelé. Samotná dokumentace vznikne až v době implementace, neboť na úrovni nízkoúrovňového prototypování, kam se v rámci této práce dostanu, je stále vysoce pravděpodobné, že se bude rozhraní systému výrazně měnit.

3.3 Rozbor obdobných systémů

Součástí návrhu uživatelského rozhraní je také rozbor obdobných systémů. Tímto krokem chci zjistit běžné konvence stanovené v obdobných systémech a usnadnit tím používání mnou navrženého systému.

Text této sekce čerpá ve značné míře ze semestrální práce předmětu Návrh uživatelského rozhraní (MI-NUR). Úkolem bylo navrhnout uživatelské rozhraní pro obdobný systém pacientů akutní lymfoblastové leukémie. Semestrální práce byla kolektivním dílem autorů: Adam Sekereš, Radim Janda, David Jirout a Tomáš Pastorek. Průběh práce dozoroval Ing. Jiří Hunka.

Navrhovaný systém je individuální řešení pro konkrétního zadavatele. Obdobné systémy zaměřené konkrétně na problematiku sledování pacientů s leukémií jsem nenalezl. Nicméně základní funkcionality navrženého systému jsou vyhledávání (a filtrování) pacientů, zobrazení a editace dat konkrétního pacienta a historie jeho léčby. Tyto požadavky jsou velice podobné téměř každému ambulantnímu systému užívaného v ordinacích lékařů. Z toho důvodu jsem se rozhodl průzkum obdobných systémů zaměřit právě na tyto ambulanční systémy používané v České Republice.

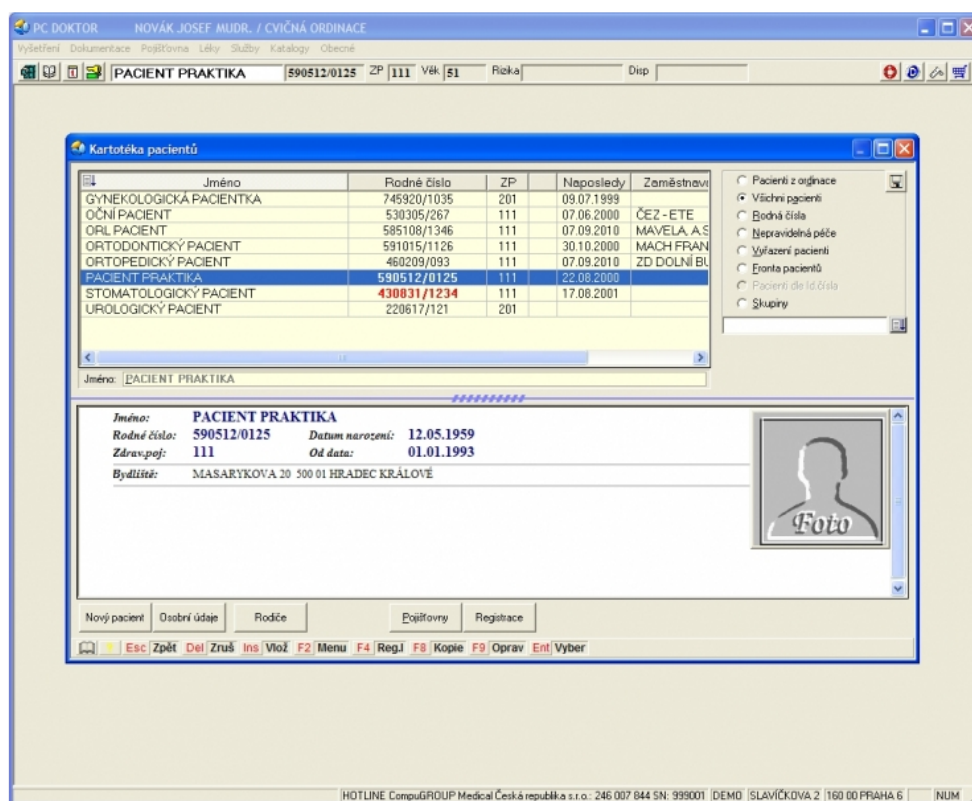
Ve své analýze těchto systémů jsem se zaměřil na sledování řešení zmíněných tří základních problémů. Ve své analýze nezkoumám funkčnost systémů, pouze jejich uživatelské rozhraní. Funkčnost je striktně definována analýzou domény a požadavky zadavatele.

3.3.1 Vybrané systémy

Pro svoji analýzu jsem zvolil programy PC DOKTOR[15], WinMed 2[16] a Medicus[17]. Jedná se o rozšířené programy s dlouholetou tradicí, které pokrývají široké spektrum potřeb uživatelů od evidence pacientů přes vypisování receptů až po evidenci tržeb. Programy se používají v ordinacích praktických lékařů, ambulancí a menších nemocničních zařízení.

Ve všech zkoumaných systémech se jedná o placené licencované produkty. Analýza rozhraní se opírá o dokumentaci a příručky nalezené na webových stránkách produktů.

3. NÁVRH UŽIVATELSKÉHO ROZHŘANÍ



Obrázek 3.2: PC Doktor - kartotéka pacientů

3.3.2 PC Doktor

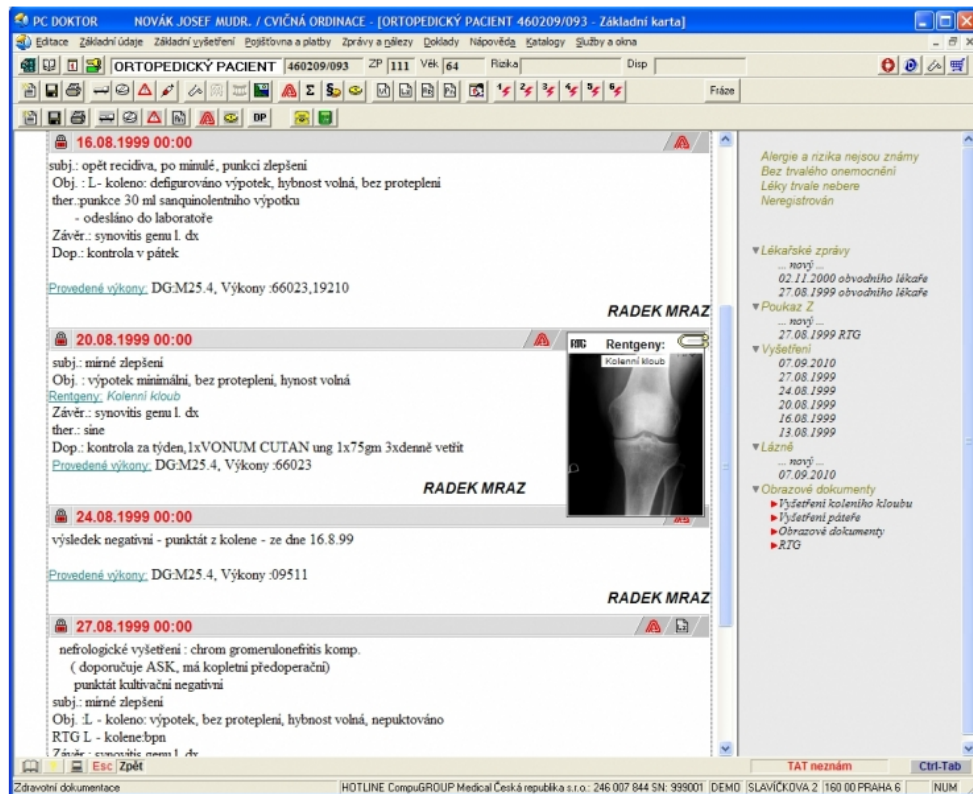
PC DOKTOR je moderní ambulantní program, určený pro všechny ambulantní praxe včetně praktických lékařů a pediatrů. Poskytuje ideální řešení jak pro malou ambulanci, tak i pro velkou polikliniku či malé lůžkové zařízení[15].

3.3.2.1 Řešení evidence pacientů

Systém zobrazuje seznam pacientů ve formě tabulky, vybráním konkrétního pacienta se zobrazí jeho osobní údaje (viz obr. 3.2). Systém dále zobrazí po výběru pacienta jeho historii léčebných zákroků (viz obr. 3.3). Obrazovky a orientace mezi nimi jsou řešeny systémem pozicovatelných „oken“ známých z operačního systému Windows.

3.3.2.2 Zhodnocení rozhraní

Výhodou rozhraní systému je malý počet obrazovek, který přispívá k rychlému zorientování. Hlavní systémové menu v horní části obrazovky je vždy přístupné.



Obrázek 3.3: PC Doktor - historie léčby

Negativně vnímám velký počet tlačítek na obrazovce nebo víceúrovňové menu při zanoření uživatele do více obrazovek. Na první pohled není zjevné, jaké tlačítko souvisí s jakou obrazovkou.

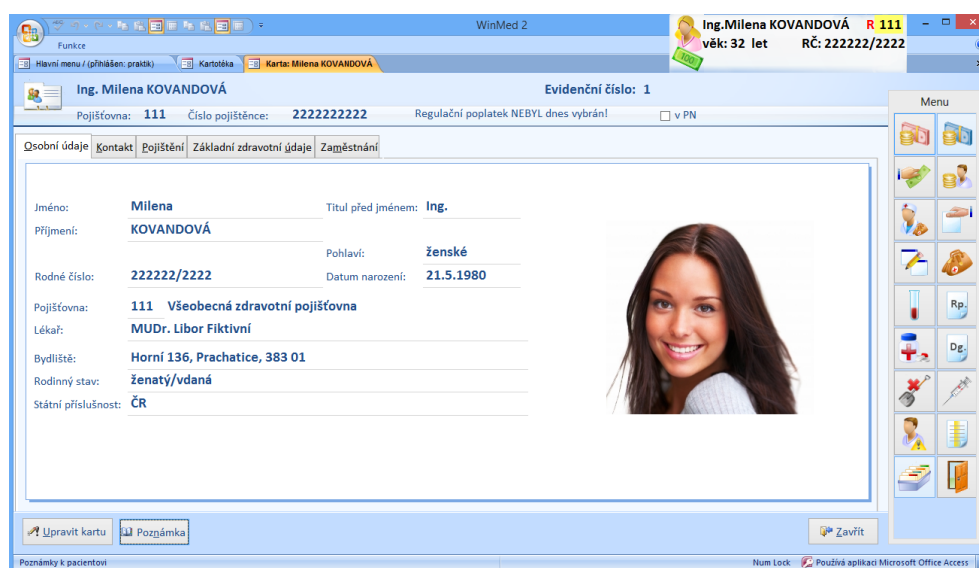
3.3.3 WinMed 2

WinMed 2 je komplexní počítačový program pro lékaře řešící veškerou potřebnou agendu soukromých ordinací a ambulantních klinik. Je založen na důsledné orientaci na konkrétní lékařskou odbornost, a nabízí tak řešení přímo pro Vaši specializaci[16].

Druhý zkoumaný program se snaží svým rozhraním přiblížit sadě kancelářských programů Microsoft Office. Navigace v systému je řešena formou záložek, mezi kterými lze libovolně přepínat. Informace o pacientovi jsou odděleny do vlastní záložky a rozčleněny dle logických skupin do několika obrazovek (viz obr. 3.4). Historie léčby je oddělena do vlastní obrazovky (viz obr. 3.5).

Formu zpracování kartotéky, neboli seznamu pacientů, se mi nepodařilo nalézt, každopádně obrázek 3.6 zobrazuje použití filtrování pacientů.

3. NÁVRH UŽIVATELSKÉHO ROZHŘANÍ



Obrázek 3.4: WinMed 2 - detail pacienta

3.3.3.1 Zhodnocení rozhraní

Rozhraní podobné sadě Microsoft Office vnímám jako výhodu, neboť velké množství uživatelů v oboru se s takovými programy již setkalo, což urychlí zorientování v systému. Zároveň také rozhraní působí modernějším dojmem, než předchozí PC Doktor.

Systém záložek dovede poměrně dobře rozdělit informace a umožní uživatelům rychlou navigaci. To ale pouze, pokud si uživatel neotevře záložek příliš mnoho a neztratí mezi nimi orientaci.

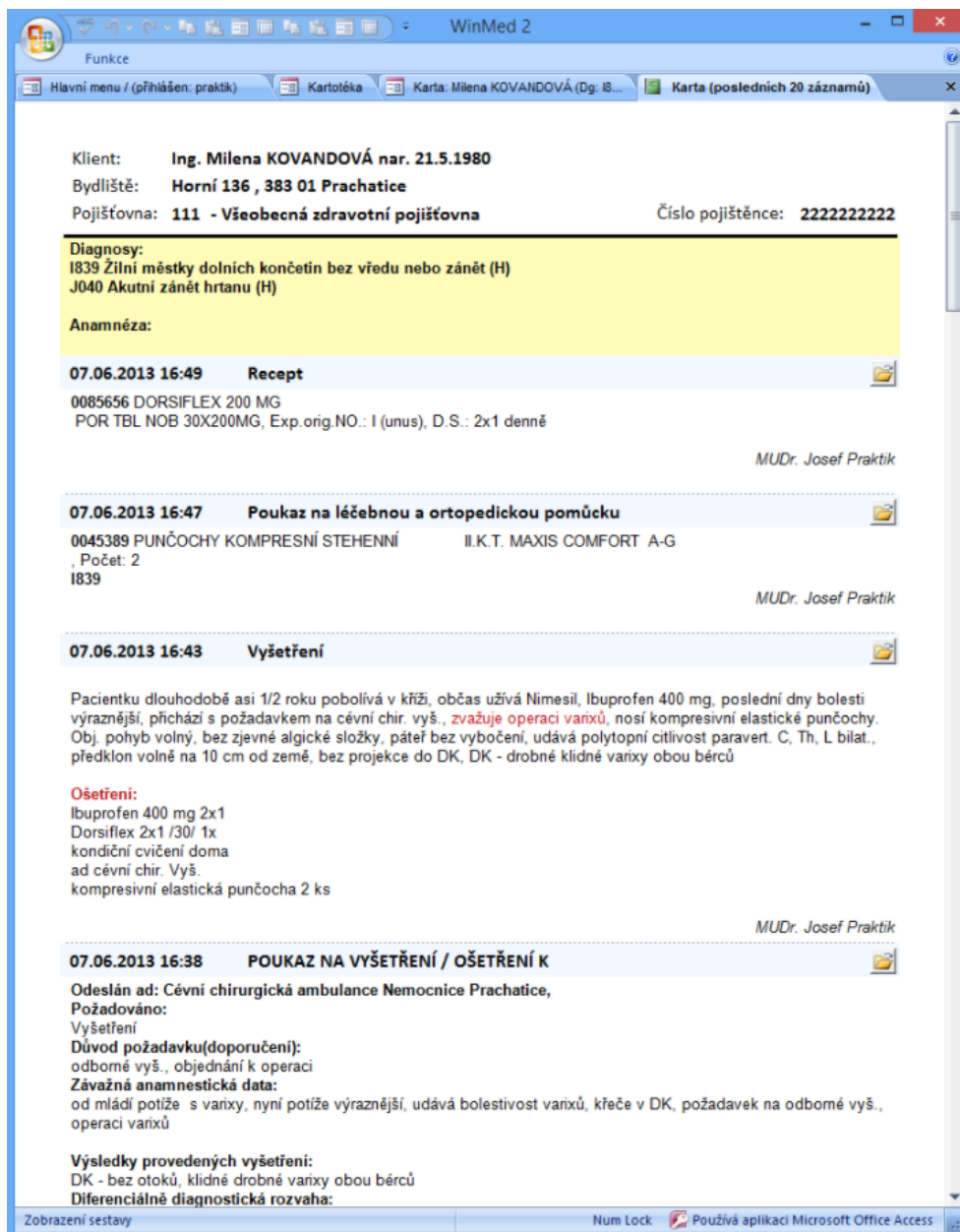
Jako nevýhodu vnímám záložku hlavního menu, která by neměla být součástí ostatních záložek, nebo by měla být zvýrazněna, aby umožnila uživateli rychlý přístup do výchozího stavu (v souladu s 3. bodem Nielsenovy heuristiky 3.2.3). Při používání systému si dovedu představit případ, kdy uživatel otevře větší množství záložek a ztratí přehled o tom, která je záložka hlavního menu.

3.3.4 Medicus

Programy řady MEDICUS patří mezi nejmodernější a nejprodávanější zdravotnický software v naší republice. Díky své rozmanitosti a dostupnosti mnoha funkcí jsou vhodné pro široké spektrum odborností. Využijí je jak samostatní lékaři, tak i velká zdravotnická zařízení[17].

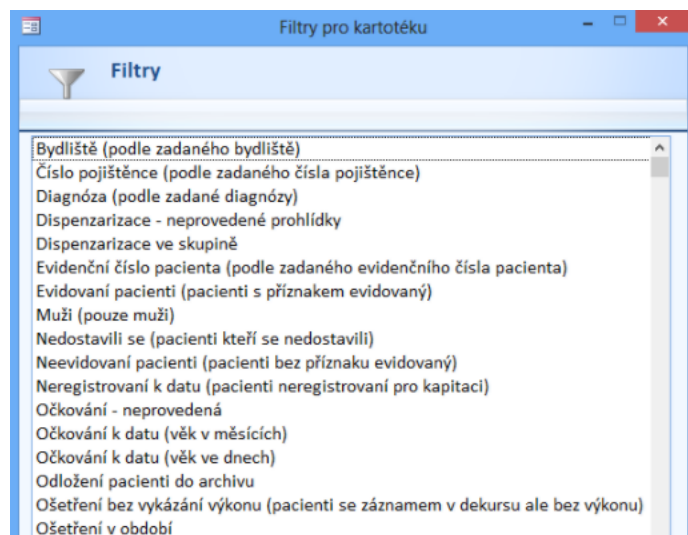
Poslední zkoumaný program sleduje konvenci operačního systému Windows a organizuje data v rámci systému do otevíratelných oken. Kartotéka pacientů je implementována formou tabulky (viz obr. 3.7), přičemž řádek tabulky je přímo propojen s kartou pacienta (viz obr. 3.8). Karta je tvořena

3.3. Rozbor obdobných systémů

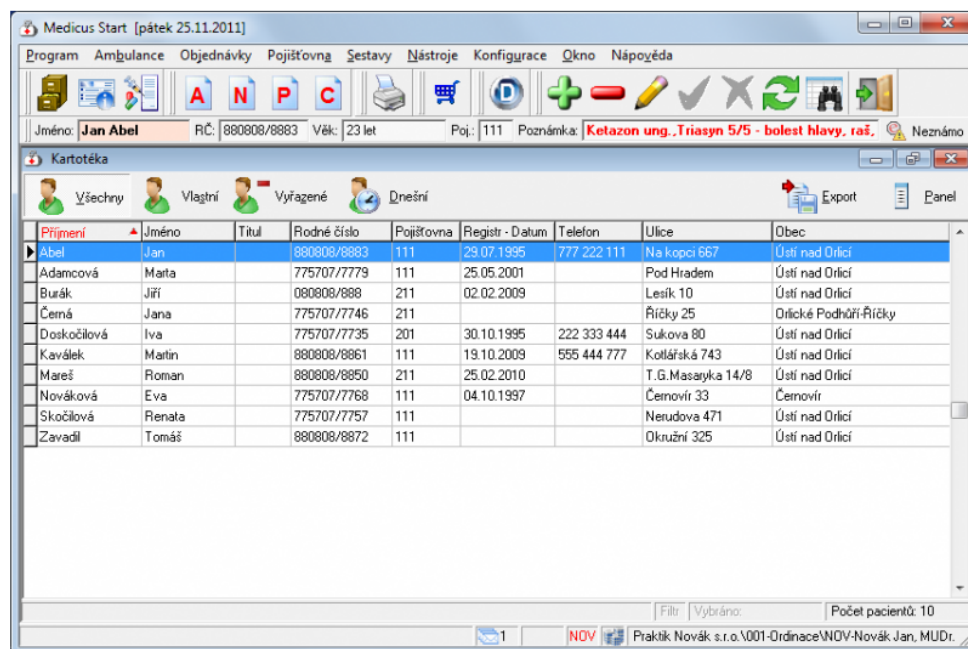


Obrázek 3.5: WinMed 2 - historie léčby

3. NÁVRH UŽIVATELSKÉHO ROZHRANÍ



Obrázek 3.6: WinMed 2 - filtrování pacientů



Obrázek 3.7: Medicus - kartotéka pacientů

3.3. Rozbor obdobných systémů

Identifikace [Jan Abel]

Identifikační údaje

Rodné číslo: 880808/8888 + Osobní číslo: + Datum narození: 08.08.1988 [15]
 Příjmení: Abel Pohlaví: Muž
 Jméno: Jan Stav: 0 - nezjištěno
 Titul: Kartička založena: [15]
 Rozená: Státní občanství: Česká republika ...
 Etnická skupina: Průkaz: Cizinec:

Pojišťovna | **Ošetřující lékaři** | **Bydliště** | **Kontakt** | **Zaměstnání** | **Registrace** | **IZIP**

Kód: 111 ... Všeobecná zdravotní pojišťovna ČR
 Ode dne: 29.07.1995 [15] Druh pojištění: 1
 Registroval: Typ péče: Cenové pásmo: Stav pojištěnce na portálu VZP

Datum	IČP	Odb.	Příznak	Zrušeno	Poj
29.07.1995	11-111-001	001	D		111

Poznámka | **Poznámka [001]**
 Studie Atractiv

Obrázek 3.8: Medicus - detail pacienta

Ambulantní karta [Jan Abel]
 Ambulantní karta [NOV út 28.6.2011 00:00]

Subj.:
 Celkové prev. vyš.,
 Subj. se cítí dobře.,

Obj.:
 Normosth. habitu, kůže a viditelné sliznice prorkvené, ve kšticí derematitis, psych.kompl., hlava V korig. skly ODS 5/7.5 šepot 5 m billat., nosní ragáda, thy. nehm., uzliny nezvět., Hrudník klenutv. pohyblivv. ookle olnv. bez díf. dvých.alveol. čistě. S prav.. ozvv

Dnešní záznam [NOV pá 25.11.2011 08:34]
 Dop. lipidy, ionty, ALT
 Hipres 5 III 1.0.0, Amprilan H 5/25 III I 1.0.0
 Atorvastatin active 20 III 1x1, Triamcinolon sol. ivax 10 III I x na noc.
 Schopen jako fidič sk. A,AI,B - schopen s brýlemi.

Předepsané léky: AMPRILAN H 5 MG/25 MG POR TBL NOB 84. 0104720 (1x), FROMILID UNO POR TBL RET 14X500MG, 0023315 (1x), TRIAMCINOLON-IVAX DRM SOL 1X10ML, 0016443 (1x)

Výkony: 01021 (I10), 09543 (I10)

MUDr. Jan Novák

Základní dg.: I10 ... Ostatní dg.: ... Barva: ...
 Odesílající lékař: ... Příští návštěva: 02.12.2011 [15]

Přehled Dekursy

Dg.	Datum
I10	29.06.2011
I10	28.06.2011
I10	30.03.2011
I10	17.12.2010
I10	24.11.2010
I10	22.09.2010
I10	25.06.2010
I10	30.04.2010
I10	23.03.2010
I10	01.12.2009
I10	23.09.2009
Z000	26.06.2009
I10	27.03.2009
I10	25.09.2008
27.6.	27.06.2008
I10	07.05.2008
I10	27.03.2008
	18.12.2007
	05.10.2007
Z000	04.10.2007
	15.08.2007

Obrázek 3.9: Medicus - historie léčby

přímou editovatelnými formuláři a rozčleněna do několika logických skupin obsahující informace o pacientovi (podobně jako v předchozím případě). Historie léčby, v tomto případě ambulantní karta, je oddělena opět do samostatné obrazovky (viz obr. 3.9).

3.3.4.1 Zhodnocení rozhraní

Rozhraní působí jako jednoduché a přehledné. Systém je ze zmíněných nejednodušší pochopitelný. Systém oken trpí podobnou nevýhodou jako záložky, přičemž větší množství otevřených oken je matoucí pro uživatele.

3.3.5 Zhodnocení

Všechny zkoumané systémy působí nepřehledným dojmem, neboť se svojí funkcionalitou snaží pokrýt široké spektrum potřeb různých odborností. Vzhledem ke specifické aplikaci mnou navrhovaného systému bude systém výrazně jednodušší.

Každý systém odděluje logické celky vyhledávání pacientů (kartotéka), karta pacienta (detail) a historie léčby do vlastních obrazovek. Ve svém systému převezmu tento koncept. Historie léčby má obdobu v mém systému v podobě měření reziduálních nemocí, tato data zobrazím společně s požadovaným grafem průběhu nemoci.

Podobně, jako je tomu o zkoumaných aplikacích, úvodní obrazovkou bude seznam pacientů ve formě tabulky, odkud bude možné zobrazit detail pacienta.

Zkoumané systémy používají pro zobrazení velkého množství informací o pacientech systém záložek, který také převezmu do svého návrhu.

3.4 Konceptuální návrh

Po předchozí analýze konkurence nyní přistoupím k samotnému návrhu. Před tím, než začnu modelovat rozhraní, začnu proces konceptuálním návrhem a rozložením obrazovek. Proces začíná definováním všech uživatelských akcí (*task list*). Následně tento seznam akcí rozdělím do logicky souvisejících skupin a přiřadím obrazovkám (*task groups*). Závěrem konceptuálního návrhu vytvořím graf, který ilustruje vztah mezi obrazovkami, akcemi a přechody mezi nimi (*task graph*).

3.4.1 Task list a task groups

Prvním krokem konceptuálního návrhu je definovat všechny atomické akce uživatelského rozhraní. Následující seznam obsahuje výpis všech těchto akcí. Seznam zároveň jednotlivé akce seskupuje do logických množin, které rozdělují do obrazovek.

- Obrazovka kartotéky
 - Vyhledání pacienta dle identifikačních údajů
 - Zobrazení seznamu pacientů
 - Filtrování pacientů
 - Resetování nastavených filtrů
 - Přejít na kartu vybraného pacienta
 - Exportování vyhledaného seznamu pacientů
 - Přidání aktuálně vyhledané (filtrované) množiny pacientů do grafu přežití
 - Vyvolání formuláře pro přidání nového pacienta
- Obrazovka kontroly přístupu
 - Zobrazení seznamu uživatelů
 - Vytvoření pozvánky pro nového uživatele
 - Úprava přístupových práv stávajícímu uživateli
 - Odstranění uživatelského účtu
 - Zneplatnění odeslané pozvánky
- Obrazovka nastavení číselníků
 - Výběr číselníku pro úpravu
 - Vytvoření/editace/smazání hodnoty
- Obrazovka grafů přežití
 - Upravit libovolnou křivku (změnit název a barvu)
 - Odstranit křivku,
 - Zobrazit graf *Overall survive* a *Progression free survive*
- Obrazovka logu systému
 - Zobrazení tabulky historie změn v celém systému
 - Možnost zobrazit detail konkrétní změny (vhodné pro případ, kdy se data nevejdou do tabulky)
- Obrazovka karty pacienta
 - Zobrazení identifikačních údajů pacienta
 - Zobrazení lékařských údajů pacienta (tabulka pacientů)
 - Vytvoření/editace/smazání údajů

3. NÁVRH UŽIVATELSKÉHO ROZHŘANÍ

- Omezení editace uživatelům dle laboratorních rozsahů
- Obrazovka reziduální nemoci
 - Zobrazení tabulky měření
 - Možnost upravit (vytvořit, editovat, smazat) záznam měření
 - Zobrazení grafu naměřených hodnot v čase
- Obrazovka logu pacienta
 - Zobrazení tabulky historie změn daného pacienta
 - Možnost zobrazit detail konkrétní změny
- Společné na všech obrazovkách
 - Odhlášení uživatele
 - Změna přístupového hesla
 - Úprava uživatelských údajů
 - Navigace na domovskou obrazovku (Kartotéka)
- Vlastní obrazovky
 - Přihlášení uživatele
 - Registrace nového uživatele
 - Obnova zapomenutého hesla

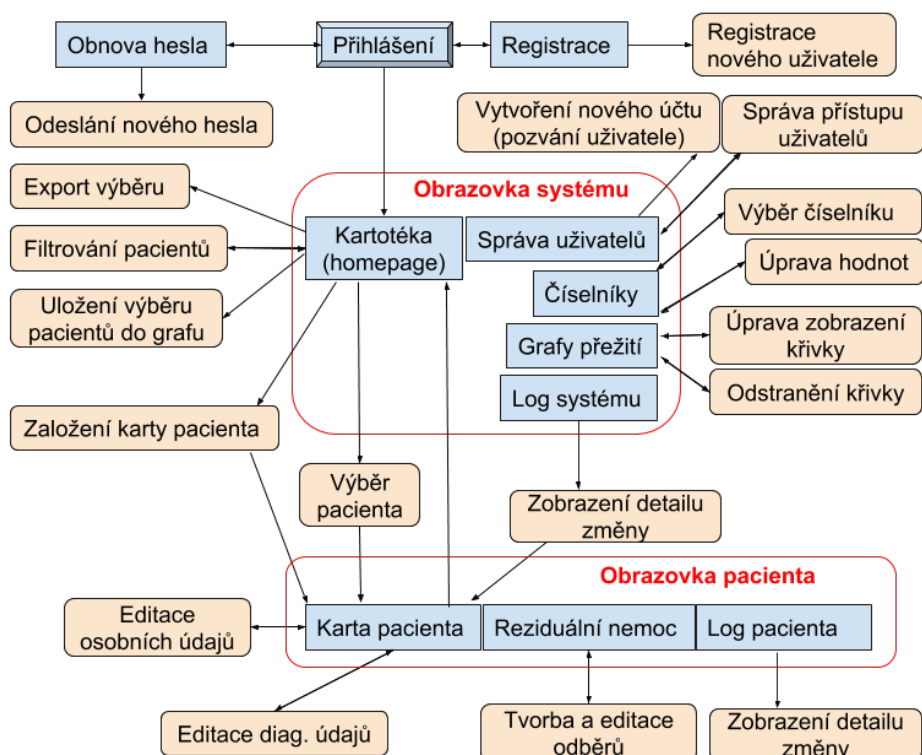
3.4.2 Task graph

Dalším krokem konceptuálního návrhu je *task graph*, který vizualizuje koncepci návrhu. Graf znázorňuje obrazovky a jednotlivé akce mezi nimi.

Ve svém návrhu rozhraní jsem se rozhodl vytvořit dvě skupiny obrazovek a to „obrazovku systému“ a „obrazovku pacienta“. Obrazovka pacienta bude zobrazovat veškeré údaje spojené s jedním konkrétním pacientem a umožní veškeré editace. Obrazovka systému seskupí veškeré akce se systémem jako takovým.

Výsledný graf popisuje obrázek 3.10. Modré obdélníky značí obrazovky. Obdélník *Přihlášení* značí vstupní bod systému. Šipky mezi obrazovkami značí možnost navigace. Obrazovky seskupené červeným obdélníkem sdílí kontextové menu, tedy jsou mezi sebou úplně propojeny. Obrazovky *Kartotéka*, *Správa uživatelů*, *Číselníky*, *Grafy přežití* a *Log systému* patří do skupiny obrazovky systému. *Karta pacienta*, *Reziduální nemoc* a *Log pacienta* reprezentují obrazovku pacienta.

Ostatní obdélníky značí akce rozhraní. Šipka směřující z obrazovky na akci znamená, že danou akci lze vyvolat na dané obrazovce. Šipka vycházející z akce směřující do obrazovky znamená, že daná akce svým výstupem překreslí rozhraní dané obrazovky, například přibude nový pacient do tabulky.



Obrázek 3.10: Task graph

3.5 Prototyp

V předchozích sekcích jsem konceptuálně navrhl rozhraní, nyní přistoupím k praktické implementaci ve formě nízkourovňového prototypu.

Ve svém prototypu zanedbávám jednoduché dialogy s formuláři, které slouží pouze k účelu zadávání dat do databáze, jako například založení nové karty pacienta. Dále zjednodušuji grafy, jejich vizuální specifikace je sepsaná v práci. V prototypu se soustředím na rozvržení důležitých obrazovek a navigaci mezi nimi. Jedná se o první iteraci návrhu, u které očekávám četné změny po konzultaci se zadavatelem.

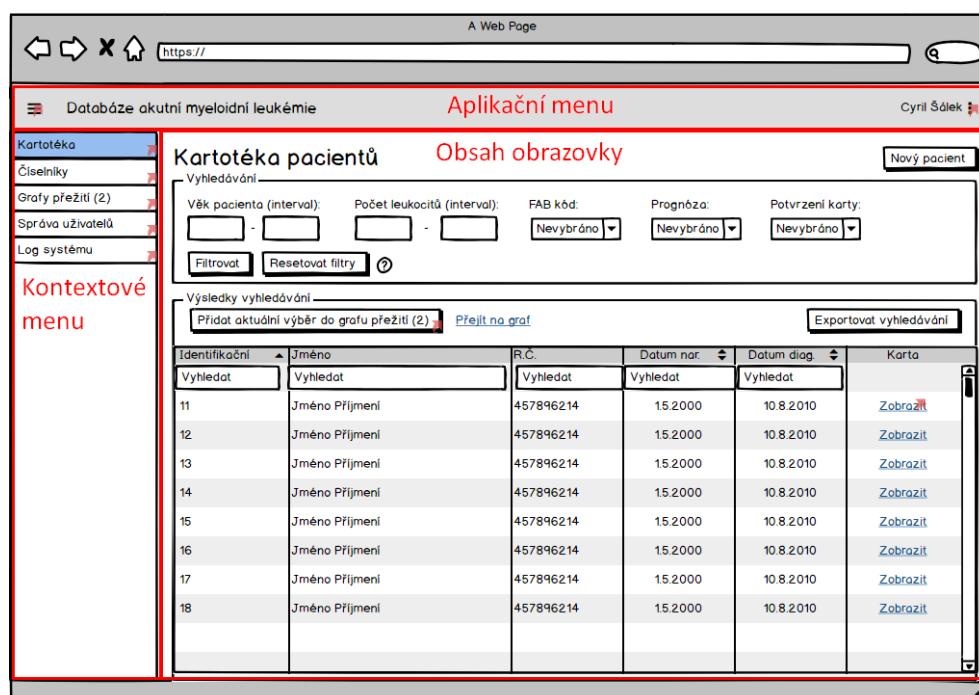
Veškeré návrhy jsou modelovány v okně prohlížeče s rozlišením odpovídající minimu dané nefunkční specifikací, tedy 1024x768px.

V textu práce dále popisují návrh několika základních obrazovek, kompletní wireframe je k dispozici ve formátu PDF na přiloženém CD.

3.5.1 Obecné rozvržení obrazovky

Obrázek 3.11 znázorňuje typickou obrazovku systému. Každá obrazovka se skládá ze tří základních částí, a to aplikačního menu, kontextového menu a samotného obsahu stránky.

3. NÁVRH UŽIVATELSKÉHO ROZHŘANÍ



Obrázek 3.11: Obecné rozvržení obrazovky

Aplikační menu je společné celé aplikaci, má vždy stejný obsah a je umístěno na vrcholu obrazovky. V levé části je tlačítko pro skrytí kontextového menu, následuje název aplikace, který slouží zároveň jako odkaz na domovskou stránku kartotéky (tato funkcionality není implementována v prototypu). V pravé části je jméno aktuálně přihlášeného uživatele společně s tlačítkem pro vyvolání uživatelského menu, umožňující odhlášení uživatele, změnu hesla a uživatelských údajů.

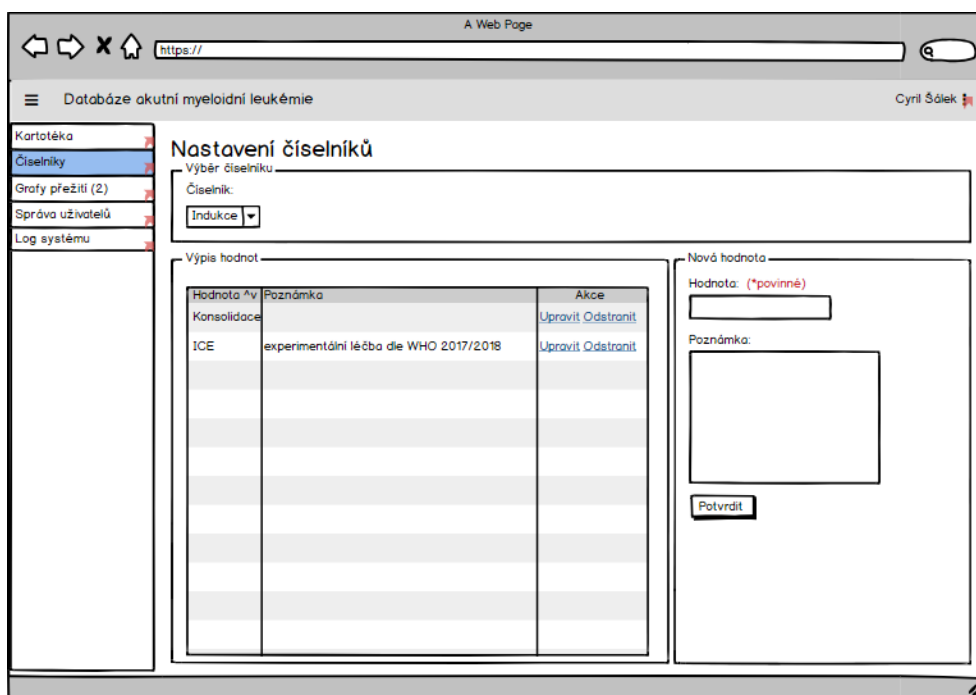
Kontextové menu je společné pro skupinu obrazovek systému a slučuje dílčí obrazovky, které jsou mezi sebou navigací propojené.

Obsah obrazovky se liší dle aktuální obrazovky.

3.5.2 Kartotéka

Asi nejdůležitější obrazovkou systému, na které budou svoji práci zahajovat všichni uživatelé je *Kartotéka* (viz obr. 3.11). Jedná se o domovskou obrazovku, na které se uživatelé ocitnou ihned po přihlášení a na kterou lze navigovat z jakékoliv obrazovky systému kliknutím na název aplikace nebo záložku kontextového menu, která je přítomná i na obrazovce karty pacienta.

Obrazovka je tvořena ve své horní části vyhledáváním, které umožní uživatelům vyhledávat pacienty dle specifikací požadovaných atributů. Ve své spodní části je tabulka pacientů. Pokud nejsou stanovena žádná vyhledávací



Obrázek 3.12: Prototyp obrazovky nastavení číselníkových hodnot

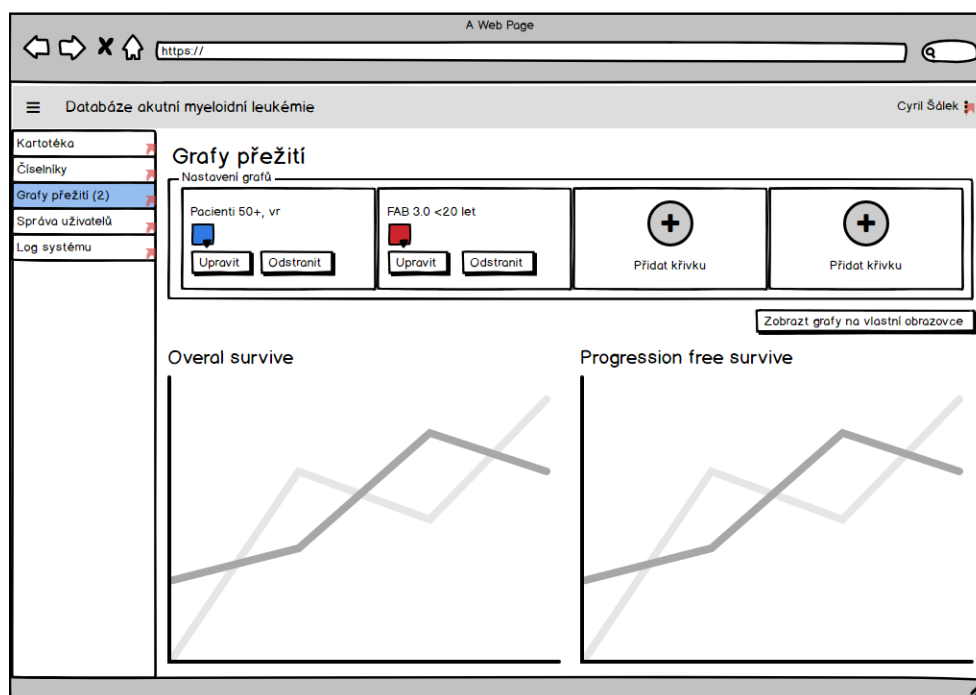
kritéria, tabulka obsahuje ve stránkované podobě všechny pacienty v databázi. U každého sloupce (krom sloupce *Karta*) je k dispozici „rychlé vyhledávání“, které umožní uživateli vyhledat pacienty dle daného sloupce. Například přistoupí-li lékař na tuto obrazovku s úmyslem co nejrychleji nalézt konkrétního pacienta, přičemž zná jeho identifikační číslo, může jej zadat do příslušného sloupce a systém okamžitě vyhledá daného pacienta. V posledním sloupci je pak možné kliknutím na odkaz přistoupit na obrazovku karty daného pacienta.

Nad tabulkou pacientů, v prostřední části obrazovky, jsou tlačítka pro ovládání grafů, kde si může uživatel pojmenovat tabulkou obsažený výběr pacientů a vytvořit pro něj křivku v grafu přežití, na kterou se může dalším tlačítkem navigovat. Poslední tlačítko na řádce umožní exportovat daný výběr do excelového formátu.

3.5.3 Nastavení číselníků

Další obrazovkou ze skupiny systémových obrazovek je *Nastavení číselníků* (viz obr. 3.12). Tato poměrně jednoduchá obrazovka má umožnit uživatelům upravovat všechny číselníkové hodnoty vytvořené při procesu normalizace databáze. Uživatel zvolí požadovaný číselník a následně se mu zobrazí výpis všech aktuálních hodnot, které může dle libosti editovat nebo mazat a současně na

3. NÁVRH UŽIVATELSKÉHO ROZHRANÍ



Obrázek 3.13: Prototyp obrazovky s grafy přežití

pravé straně formulář pro rychlé vytvoření nové hodnoty.

3.5.4 Grafy přežití

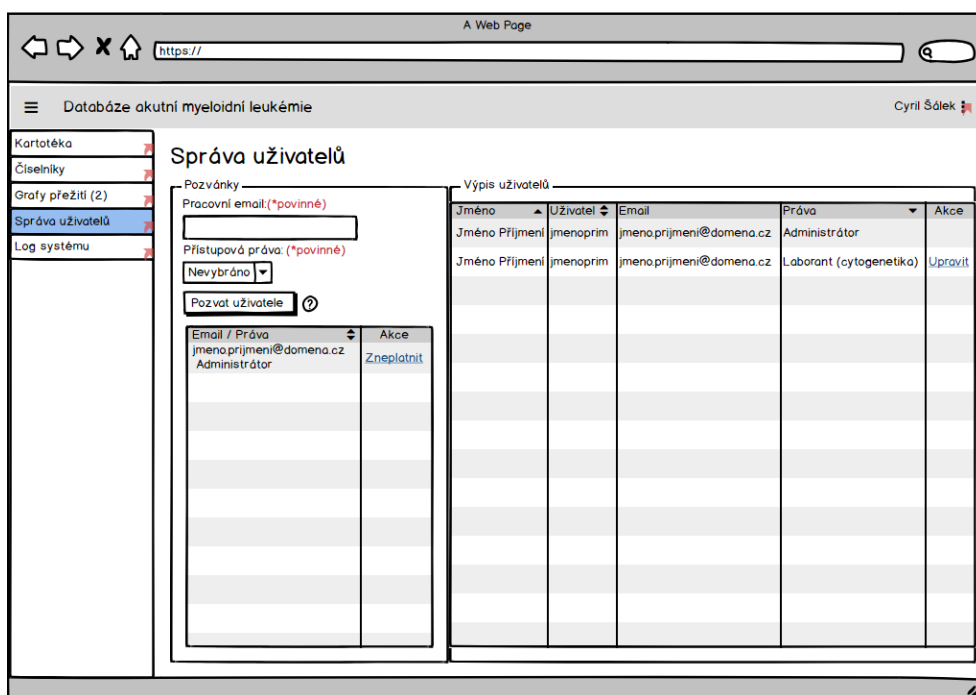
V pořadí třetí obrazovkou kontextového menu systému je *Grafy přežití* (3.13). Tato obrazovka umožní uživateli nastavit křivky grafů přežití *Overall survive* a *Progression free survive*. Oba grafy mají totožné nastavení popisu a barvy křivek. Uživatel může upravit již existující křivky a to změnit jejich barvy či popisky os. Pokud uživatel zvolí možnost upravit křivku a nebo přidat novou křivku, dojde k přesměrování na domovskou obrazovku *Kartotéka*, kde uživatel vyhledá požadovanou množinu pacientů a nahradí jednu z aktuálně vykreslených křivek novým výběrem, nebo vytvoří novou.

V případě, že uživatel má vykresleny všechny 4 křivky, systém neumožní vytvořit dalším výběrem novou, pouze nahradit jednu z existujících.

Obrazovka umožní stisknutím tlačítka zobrazit grafy v nové záložce prohlížeče pro větší plochu grafu a detailnější pohled na data.

3.5.5 Správa uživatelů

Předposlední obrazovkou ze skupiny systémových obrazovek je *Správa uživatelů* (3.14). Tato obrazovka, přístupná pouze uživatelům s právy *Administrátor*



Obrázek 3.14: Prototyp obrazovky správy uživatelů

a vyšší, umožňuje spravovat uživatelům přístup do systému.

Obrazovka ve své levé části umožní uživatelům odeslat pozvánku na email nového uživatele, sledovat všechny platné pozvánky a zneplatnit libovolnou z nich před uplynutím třicetidenní lhůty platnosti.

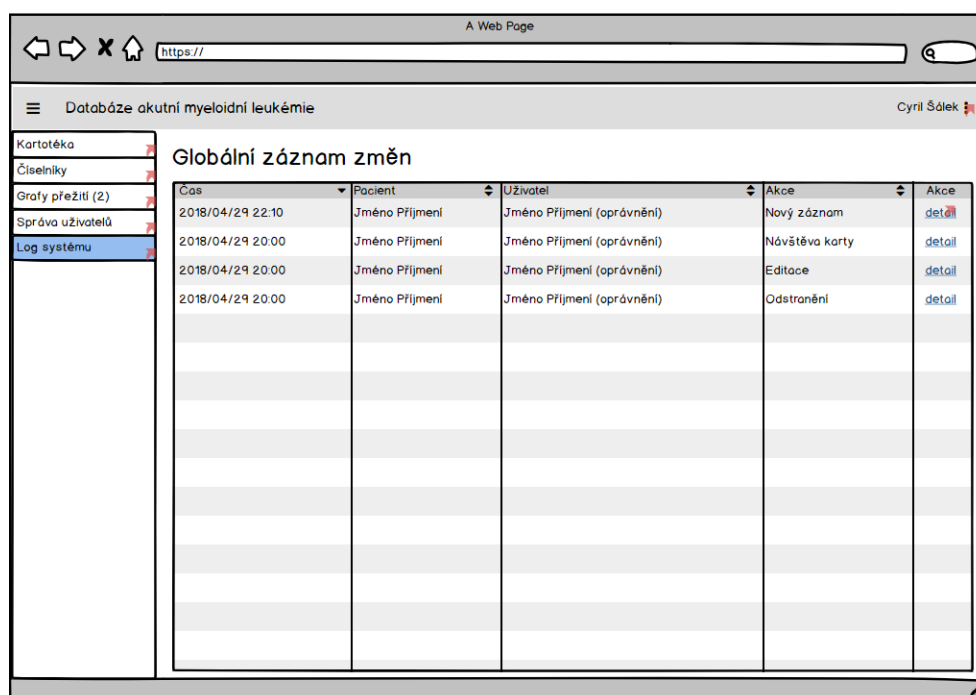
Na pravé straně obsahuje přehled všech registrovaných uživatelů (kromě právě přihlášeného) a umožní změnit přístupová nastavení uživatelů s nižším oprávněním.

3.5.6 Log systému

Poslední systémovou obrazovkou je *Log systému* (3.15). Obsahem této obrazovky je tabulka obsahující veškeré záznamy o změnách dat a přístupu k datům pacientů. V přehledu je zobrazen čas, pacient, uživatel s informací o přístupovém oprávnění a typ provedené akce.

Poslední sloupec obsahuje odkaz pro zobrazení detailu konkrétního záznamu. Na obrázku 3.16 je zobrazena ukázka dialogu s detailem změny. Zde je zobrazena také informace o konkrétní databázové tabulce a seznam změněných sloupců. Současně okno dialogu umožňuje navigovat uživatele na obrazovku karty daného pacienta.

3. NÁVRH UŽIVATELSKÉHO ROZHRANÍ



Obrázek 3.15: Prototyp obrazovky logu systému

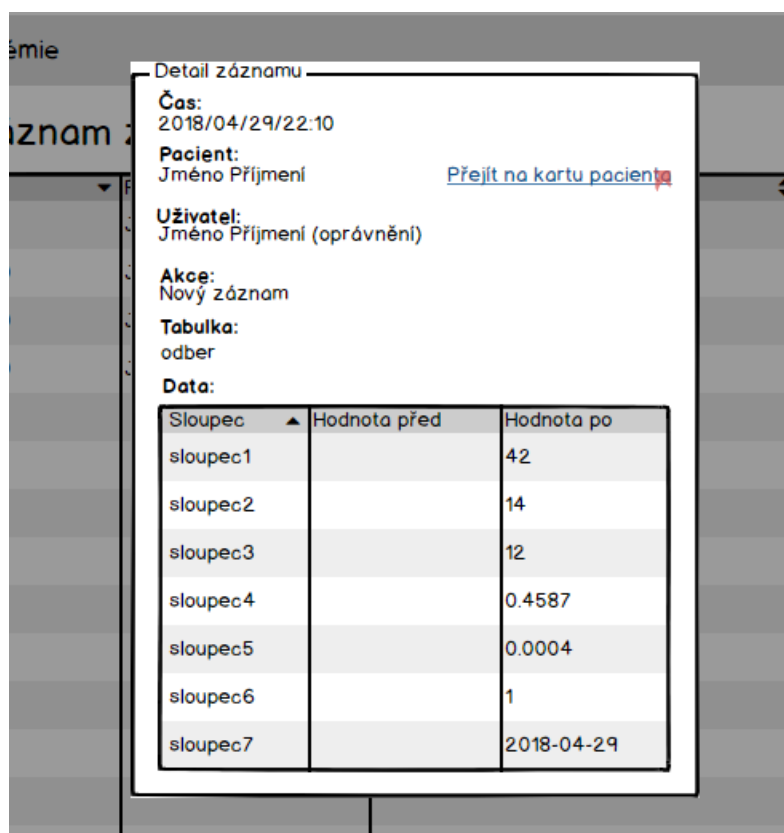
3.5.7 Karta pacienta

První obrazovkou ze skupiny kontextového menu pacienta je *Karta pacienta* (3.17). Kontextové menu obrazovky nabízí na první (nejrychleji dosažitelné) pozici návrat na domovskou obrazovku kartotéky. Následuje informace o jméně aktuálně zobrazeného pacienta. Tato informace se nachází také v nadpisu aktuální obrazovky a je zduplikována do kontextového menu pro případ, že nějaký formulář překryje nadpis stránky. Uživatel tím nikdy neztratí informaci o tom, kterému pacientovi data náleží. Následuje navigace v rámci kontextu karty pacienta.

Samotná obrazovka zobrazuje veškerá data vybraného pacienta z tabulky pacientů. Data jsou rozdělena do přepínatelných záložek zřetelně zobrazených pod nadpisem stránky. Záložky jsou seřazeny shodně s pořadím dat v tabulce pacientů, což by mělo ulehčit orientaci v systému.

Data v záložkách jsou seskupena dle specifikace rozsahů pro editování. Uživatel s omezeným oprávněním může editovat pouze svoji skupinu dat. Některé záložky obsahují více skupin, což je čistě designové rozhodnutí, v zájmu efektivního využití prostoru obrazovky.

Poslední nezmíněnou funkcionalitou obrazovky je ovládání potvrzení karty. V případě, že je karta potvrzena, je zobrazeno jméno uživatele, který kontrolu provedl a tlačítko umožňující potvrzení zrušit. V opačném případě je zobra-



Obrázek 3.16: Prototyp dialogu s detailem změny

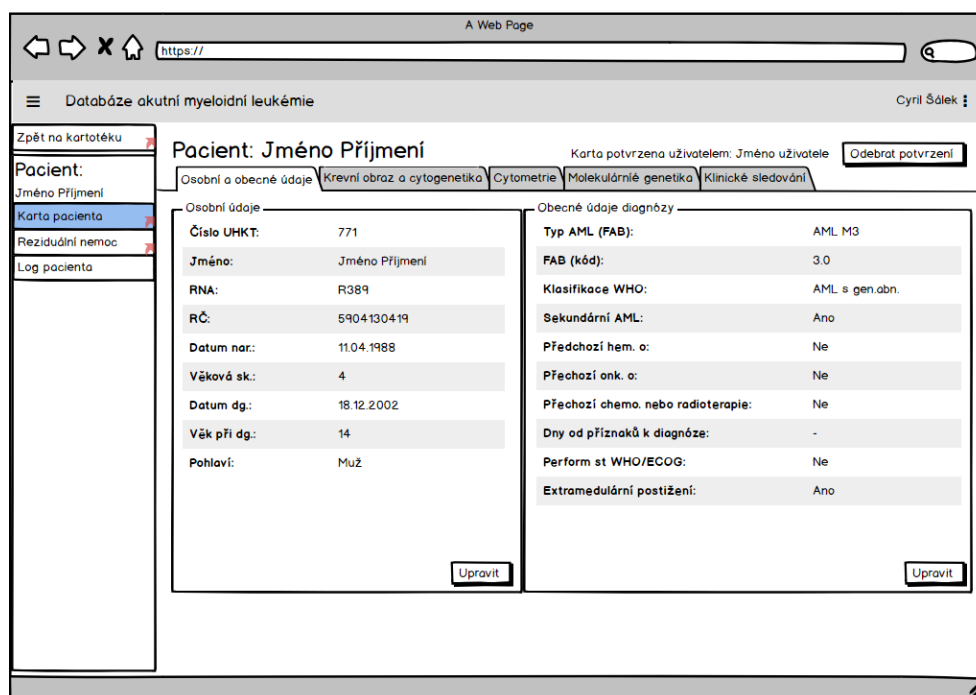
zeno pouze tlačítko pro kontrolu.

3.5.8 Reziduální nemoc

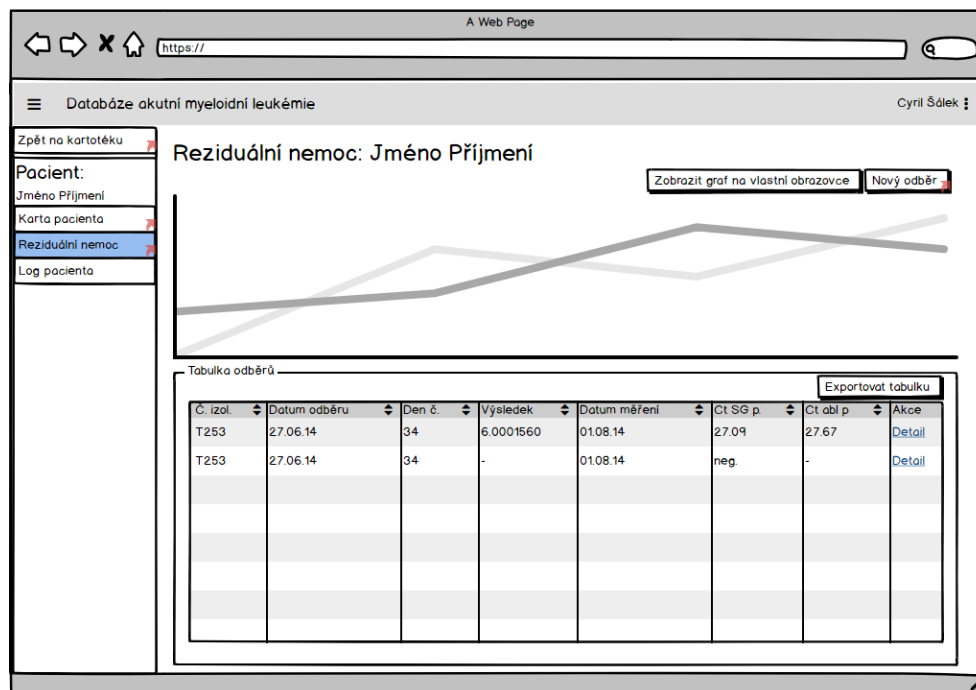
Druhou obrazovkou kontextového menu pacienta je *Reziduální nemoc* 3.18. Obrazovka je tvořena grafem, který dle specifikace vykresluje průběh nemoci. Graf lze zobrazit na vlastní obrazovce pro detailnější pohled na data.

Samotná data jsou zobrazena v tabulce pod grafem ve zjednodušené podobě zakrývající měřené hodnoty a zobrazující pouze vypočítaný průměr. Pro detailnější pohled na všechna data lze rozkliknout detail, kde je možné data upravit. Tlačítko pro nový odběr vyvolá formulář 3.19, který umožní uživateli zadat hodnoty odběru. V případě negativního odběru stačí vyplnit povinná pole formuláře a hodnota měření bude označena jako negativní. V případě naměřené nemoci stačí vyplnit pole alespoň jednoho měření zkoumaného genu a jedno měření referenčního genu abl a systém sám dopočte průměry a výsledek, což usnadní práci uživateli. Pokud by uživatel chtěl zadat jiný výsledek, než je dopočítaná hodnota, může pole sám editovat a uložit vlastní hodnotu.

3. NÁVRH UŽIVATELSKÉHO ROZHRANÍ



Obrázek 3.17: Prototyp obrazovky karty pacienta



Obrázek 3.18: Prototyp obrazovky reziduální nemoci

Nový odběr

Číslo izolace: (*povinné)

Datum odběru: (*povinné)

Den číslo: (*povinné)

Datum měření: (*povinné)

Ct SG hod 1, 2, 3:
Ponechte prázdné pro negativní odběr

Ct SG průměr:

Ct abl hod 1, 2, 3:

Ct abl průměr:

Výsledek ($2^{**}(\text{Ct abl} - \text{Ct SG})$)

Potvrdit Zrušit

V případě negativního odběru netřeba vyplňovat nepovinná pole.

Hodnoty průměrů a výsledku se dopočítávají automaticky jak uživatel zadává ostatní hodnoty. Všechny pole lze dodatečně editovat.

Obrázek 3.19: Prototyp formuláře zadání odběru

3.5.9 Log pacienta

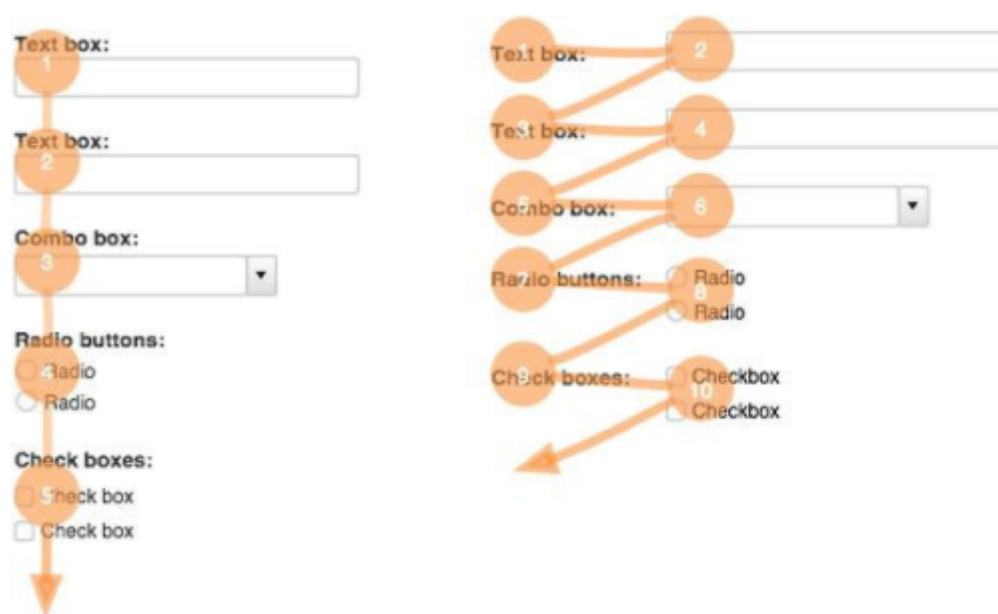
Poslední obrazovkou kontextového menu pacienta je *Log pacienta*. Tato obrazovka má totožnou funkci, jako *Log systému*. Obsahuje veškeré záznamy změn týkající se pouze aktuálního pacienta. Z důvodu podobnosti není tato obrazovka modelována.

3.5.10 Optimální návrh formulářů

Zadávání a úprava dat je nejčastějším úkonem uživatelů v navrhovaném systému. Uživatelé tedy stráví většinu svého času vyplňováním formulářů, a proto chci věnovat velkou pozornost jejich optimálnímu návrhu.

Každý formulářový prvek se obecně skládá z několika prvků. Jedná se o samotné pole, které uživatel vyplní požadovanou hodnotou. Dále popis daného pole, který informuje uživatele, jaká hodnota se zde očekává, a nakonec

3. NÁVRH UŽIVATELSKÉHO ROZHRANÍ



Obrázek 3.20: Pozicování popisků formulářového pole [18]

zástupná hodnota (placeholder), který má uživateli podat další nápovědu pro vyplnění daného pole.

3.5.10.1 Popisky formulářových polí

První zkoumanou kombinací je rozložení pole a jeho popisku. Z faktu, že lidé čtou text zleva a postupují řádky směrem dolů, přichází myslitelné dvě možnosti implementace zobrazené na obrázku 3.20. Obrázek zároveň ilustruje oranžovými pruhy cestu pozornosti uživatele při obou implementacích. Je zřejmé, že první implementace s popisky nad polem je efektivnější, neboť uživatel cestuje menší vzdálenost a vyplní formulář rychleji.

3.5.10.2 Zástupné hodnoty - placeholder

Zástupné hodnoty mají za úkol dále uživateli pomoci se správným vyplněním formulářového pole. Patří zde například ukázka formátu data, nebo doplňující informace k požadované hodnotě.

Většina implementací této mechaniky vkládá text zástupné hodnoty přímo do elementu pole. Jakmile uživatel začne pole vyplňovat, placeholder zmizí a nahradí jej hodnota zadaná uživatelem.

Dle mého názoru je tento obecně používaný standard (vyučovaný například na w3schools viz. [19]) neoptimální, neboť zatěžuje uživatelskou paměť, což je v rozporu s pravidlem 6 (viz. 3.2.6) Nielsenovi heuristiky. Optimální řešení je zobrazovat zástupnou hodnotu (nápovědu) vždy, aby uživatel nemusel

Password:
Must have at least 6 characters

Obrázek 3.21: Pozicování zástupné hodnoty formulářového pole [20]

Uživatelské jméno: (*povinné)

Obrázek 3.22: Značení povinných polí formuláře

Uživatelské jméno:

*** uživatelské jméno již existuje**

Obrázek 3.23: Označení chyby formulářového pole

vzpomínat na text, který byl zobrazen předtím, než začal dané pole vyplňovat [20]. Optimální popsané řešení zobrazuje obrázek 3.21.

Samotné použití zástupných hodnot nemá dle mého názoru smysl tam, kde je evidentní z popisku pole, co má uživatel zadat. Například pole s popiskem „Příjmení“ je dostatečně popisující, že není třeba dalšího upřesnění.

3.5.10.3 Povinná pole formuláře

Povinná pole by měla být vizuálně odlišena od těch volitelných. Optimální označení je součástí popisu pole „(*povinné)“. Pokud platí skutečnost, že všechna pole až na výjimky jsou povinná, označí se ty nepovinná [21]. Obrázek 3.22 ilustruje možnou implementaci takového pole.

3.5.10.4 Validace

Výstup validace, tedy případná chybová hlášení by měla být zobrazena přímo u vyplněných polí. Zároveň by také mělo být výrazně označeno pole obsahující chybu (viz obr. 3.23). Pro chybová hlášení bývá standardem použít červenou barvu.

Závěr

Ve své práci jsem analyzoval a pro laického čtenáře popsal problematiku diagnózy a léčby leukémie, konkrétně typu akutní myeloidní leukémie. Následně jsem analyzoval proces evidence dat spojených se sledováním pacientů Ústavu hematologie a krevní transfuze diagnostikovaných akutní myeloidní leukémií. Na základě mé analýzy a konzultací se zadavatelem z řad lékařského personálu jsem sepsal funkční a nefunkční specifikace systému, který má za cíl usnadnit práci lékařům a výzkumníkům ústavu při práci s daty svých pacientů. Svůj návrh jsem následně konceptuálně modeloval pomocí diagramu případů užití, doménového modelu a diagramů popisujících scénáře průchodu systémem.

Ve druhé kapitole práce jsem se soustředil na podrobnou analýzu dat a navrhl systematický postup pro převod dat z aktuálně používané podoby excelových tabulek do mnou navržené relační databáze. Zároveň jsem stanovil pravidla pro práci s daty zajišťující jejich normalizaci a možnost budoucího čištění. Výstupem této kapitoly je implementovaný relační model databáze MySQL.

Ve třetí kapitole práce jsem se zaměřil na návrh uživatelského rozhraní pro navrhovaný systém. Na začátku kapitoly jsem popsal postup návrhu, následně teorii správného návrhu. Před začátkem modelování jsem analyzoval rozhraní obdobných systémů, ze kterých jsem převzal několik vzorů pro dílčí části rozhraní. Modelování jsem začal konceptuálním návrhem, ze kterého jsem postupně přešel ke konkrétní implementaci nízkoúrovňového wireframe prototypu, který je výstupem této kapitoly.

Realizace a budoucí vývoj

Na základě mé práce začnou kolegové specializující se na převod dat pracovat na naplnění databáze a současně začne vývoj webové aplikace implementující mnou navržené uživatelské rozhraní. Celý tento proces odhaduji na 3 měsíce práce.

Na konci období prvotního vývoje bude existovat první prototyp systému, který bude prezentován zadavateli a proběhne diskuze nad jeho použitím. Zadavatelé vnesou další nároky na systém, které budou rozděleny dle priorit a dojde k jejich postupné implementaci. Tento cyklus iteračního vývoje bude probíhat do té doby, než bude systém ve své aktuální podobě schválen pro produkční použití. V ten moment dojde k aktualizaci veškerých dat. Následovat bude souběžně každodenní provoz systému a používání excelových souborů do doby, než bude systém ověřen jako schopný bezproblémového provozu. Následně se upustí od používání excelových tabulek jako formy záznamu dat a navržený systém kompletně nahradí starý proces.

Automatizace návrhu databáze

Při návrhu mého systematického procesu transformace dat do databázové podoby jsem si postupně uvědomil, že většinu manuální práce lze poměrně snadno automatizovat. Celý proces převodu dat tabulky pacientů je deterministický. Program pro analýzu dat by mohl být rozšířen tak, aby sám vytvářel databázovou strukturu dle zadaných parametrů. Jedním z parametrů by mohl být název tabulky a rozsah dat excelového souboru, který program automaticky přetransformuje do podoby databázové tabulky.

Pokud se ve svých úvahách přesunu ještě o krok abstrakce výše a zobecním použití mého přístupu nad obecnými strukturalizovanými excelovými daty, tak si dovedu představit nástroj pro čištění dat, který bude mít širokou praktickou aplikaci. Téměř v každém odvětví existuje obrovské množství dat, které jsou stále evidovány v (částečně) strukturované podobě excelových tabulek, podobně jako tomu je v případě popisovaném touto prací. Všechny tyto data podléhají znečištění a pozvolné denormalizaci vlivem času a množstvím editujících osob. Dovedu si představit nástroj, který by tato data zpracovával podobným způsobem, jaký navrhuji v mé práci. Takový nástroj dovede vyhodnotit výskyty různých hodnot a detekovat podezřelé hodnoty, například kalkulováním vzdáleností mezi nimi, které vyžadují korekci. Nástroj by také mohl nabídnout uživatelům tyto hodnoty nahradit hromadně za jiné. Současně by nabídl za pomoci tvorby číselníků vyhledávání a filtrování nad daty pro své uživatele.

Celá tato vize je hrubým náčrtem toho, kam by se mohl můj přístup rozšířit a vyžaduje další důkladné prozkoumání, které již není součástí této práce.

Dosažení cílů

S výsledkem své práce jsem spokojený. Povedlo se mi dosáhnout všech stanovených cílů, nicméně k dílčím cílům mám jisté poznatky, které v následujícím textu rozebírám.

Analýza a specifikace

Během analyzování procesů a sepisování specifikace jsem narazil na problém, kdy zadavatelé neznají problematiku návrhu systému nedovedou specifikovat systém v dostatečné míře, dokud nevidí první prototyp, tudíž sepsaná specifikace se bude v budoucnu dále rozšiřovat. Toto vnímám jako obecný problém, neboť nezkušení uživatelé nemají dostatečnou představivost. Z toho důvodu se vývoj odehrává v iteracích.

Další poznatek ze specifikace systému je spojen s komplexností legislativní úpravy evidence osobních a citlivých dat. Před zahájením svého výzkumu jsem naivně doufal, že mi bude stačit prozkoumat praktické důsledky GDPR, nicméně jsem zjistil, že existuje řada dalších právních norem, která zpřísňují, nebo upravují zkoumané GDPR. Samotná problematika by vystačila na samostatnou diplomovou práci. Jelikož nejsem odborníkem práva, považuji moji analýzu za nedostatečnou pro praktické použití a je třeba detailnější analýzy odborníkem.

Návrh databáze a transformace dat

S výsledkem navržené databáze a doporučení pro převod dat jsem celkem spokojený. Lituji však toho, že jsem si již v úvodu své práce neuvědomil to, že celý proces lze automatizovat dle výše zmíněného přístupu. Výstup mé práce by byl nicméně totožný.

Potenciální nedostatek vnímám v chybějícím testování mého řešení. Pokud bych měl více času, zaměřil bych se minimálně na to, abych otestoval rozdílný přístup k datům při implementaci tabulky pacientů jako jedné velké databázové tabulky. Navrhl bych alternativní model databáze a implementoval obě řešení. Následně vygeneroval testovací data v objemu odpovídajícímu počtu aktuálně sledovaných pacientů a porovnával doby provedení databázových dotazů. Ve výsledku si ale myslím, že rozdíl by byl při zmíněném objemu dat zanedbatelný.

Návrh uživatelského rozhraní

Úspěšnost návrhu je často subjektivní názor. V mém přístupu jsem se snažil odůvodnit každý krok návrhu systematickým přístupem, což mě dovedlo až k samotnému prototypu. Prototyp se v odevzdané podobě bude prezentovat zadavateli a bude se výrazným způsobem měnit. Tvorbu detailnějšího prototypu na cílové platformě považuji za zbytečnou a kontraproduktivní v rámci této diplomové práce.

Nedostatek této části vidím v průzkumu obdobných systémů, který by zasloužil větší pozornost. Volba obdobných systémů měla padnout na ty, které lze prakticky vyzkoušet a průzkum rozhraní mohl být detailnější. Přínos detailnější analýzy bych ale označil za diskutabilní, neboť mnou navrhovaný systém je a navržen na míru potřebám pracovníků ÚHKT.

Literatura

- [1] ŠTEFÁNEK, Jiří. *Leukémie* Medicína, nemoci, studium na 1. LF UK [online]. [cit. 2018-03-03]. Dostupné z: <http://www.stefajir.cz/index.php?q=leukemie>
- [2] ŠTEFÁNEK, Jiří. *Akutní myeloidní leukémie* Medicína, nemoci, studium na 1. LF UK [online]. [cit. 2018-03-03]. Dostupné z: <http://www.stefajir.cz/?q=akutni-myeloidni-leukemie>
- [3] ŠÁLEK, Cyril. *Akutní leukémie: Kurz hematologie pro sestry*. Ústav hematologie a krevní transfuze, 2015.
- [4] ŠÁLEK, Cyril. *Akutní myeloidní leukémie*. Postgraduální medicína. 2013, 2013(5), 547-558.
- [5] DOUBEK, Michael. *Leukemie*. Linkos: Pacient a rodina [online]. 30.7.2006 [cit. 2018-03-04]. Dostupné z: <https://www.linkos.cz/pacient-a-rodina/onkologicke-diagnozy/leukemie-c91-c95/leukemie-1/>
- [6] MATZNER, Jiří. *Jak to bude s ochranou osobních údajů po příchodu GDPR?*. Právní prostor [online]. 2017, 07.09.2017, , 1-1 [cit. 2018-03-02]. Dostupné z: <https://www.pravniprostor.cz/clanky/ostatni-pravo/jak-to-bude-s-ochranou-osobnich-udaju-po-prichodu-gdpr>
- [7] ŠKORNIČKOVÁ, Eva. *Co považuje GDPR za osobní údaje*. GDPR: Obecné nařízení o ochraně osobních údajů prakticky [online]. [cit. 2018-03-06]. Dostupné z: <https://www.gdpr.cz/gdpr/osobni-udaje/>
- [8] *EU obecné nařízení o ochraně osobních údajů: Článek 9 "Zpracování zvláštních kategorií osobních údajů"*. In: . ročník 2017. Dostupné také z: <http://www.privacy-regulation.eu/cs/9.htm>
- [9] BOZHANOV, Bozhidar. *GDPR: A Practical Guide For Developers*. Security Zone [online]. 25.12.2017 [cit. 2018-03-08]. Dostupné z: <https://dzone.com/articles/gdpr-a-practical-guide-for-developers-part-1>

- [10] *DB-Engines Ranking*. DB-Engines [online]. 01.04.2018 [cit. 2018-04-29]. Dostupné z: <https://db-engines.com/en/ranking>
- [11] *Chapter 11 Data Types*. MySQL: MySQL Documentation [online]. 2018 [cit. 2018-04-01]. Dostupné z: <https://dev.mysql.com/doc/refman/5.7/en/data-types.html>
- [12] ŽIKOVSKÝ, Pavel. *Návrh uživatelských rozhraní: 2. přednáška – Návrh UI, prototypy* [online]. In: . 2016 [cit. 2018-03-25].
- [13] NIELSEN, Jakob. *Enhancing the explanatory power of usability heuristics*. Proceedings of the SIGCHI conference on Human factors in computing systems celebrating interdependence - CHI '94. New York, New York, USA: ACM Press, 1994, 1994, , 152-158. DOI: 10.1145/191666.191729. ISBN 0897916506. Dostupné také z: <http://portal.acm.org/citation.cfm?doid=191666.191729>
- [14] NIELSEN, Jakob. *10 Usability Heuristics for User Interface Design*. Nielsen Norman Group [online]. 1.1.1995 [cit. 2018-04-10]. ISSN 1548-5552. Dostupné z: <https://www.nngroup.com/articles/ten-usability-heuristics/>
- [15] *PC Doktor: Nejoblíbenější ambulantní program* [online]. [cit. 2018-04-10]. Dostupné z: <http://www.pcdoktor.cz/>
- [16] *WinMed2: O programu* [online]. 2018 [cit. 2018-04-12]. Dostupné z: http://winmed.cz/o_programu/
- [17] *Medicus: Nejrozšířenější lékařský program* [online]. 2018 [cit. 2018-04-12]. Dostupné z: <http://www.medicus.cz/ambulance/programy/>
- [18] TAYLOR, Marcus. *58 Form Design Best Practices & Form UX Examples*. Venture Harbour [online]. 21.12.2017 [cit. 2018-04-16]. Dostupné z: <https://www.ventureharbour.com/form-design-best-practices/>
- [19] *HTML <input> placeholder Attribute*. W3schools [online]. [cit. 2018-04-16]. Dostupné z: https://www.w3schools.com/tags/att_input_placeholder.asp
- [20] SHERWIN, Katie. *Placeholders in Form Fields Are Harmful*. Nielsen Norman Group: Evidence-Based User Experience Research, Training, and Consulting [online]. 11.5.2014 [cit. 2018-04-16]. Dostupné z: <https://www.nngroup.com/articles/form-design-placeholders/>
- [21] ŽIKOVSKÝ, Pavel. *Návrh uživatelských rozhraní: 7. přednáška – Form usability* [online]. In: . 2016 [cit. 2018-04-16]

Seznam použitých zkratk

AML Akutní myeloidní leukémie

ÚHKT Ústav hematologie a krevní transfuze

GDPR General Data Protection Regulation

UI User Interface (Uživatelské rozhraní)

Obsah přiloženého CD

readme.txt.....	stručný popis obsahu CD
db	
├─ create.sql.....	SQL skript pro vytvoření navržené databáze
├─ model.mwb.....	model databáze ve formátu pro MySQL Workbench
├─ model.png.....	model databáze ve formátu PNG
excel.....	adresář obsahující vzorové excelové soubory
├─ pacienti.xlsx.....	struktura dat tabulky pacientů
├─ rezidualka.xlsx....	struktura dat tabulky měření reziduální nemoci
prototyp.....	adresář obsahující soubory prototypu
├─ prototyp.pdf.....	interaktivní prototyp ve formátu PDF
├─ prototyp.bmpr.....	zdrojový soubor projektu pro Balsamiq
thesis	
├─ src.....	zdrojová forma práce ve formátu \LaTeX
├─ thesis.pdf.....	text práce ve formátu PDF