

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

AMBULANTNÍ SYSTÉM ZALOŽENÝ NA IZIP A JEHO ZABEZPEČENÍ

DIPLOMOVÁ PRÁCE

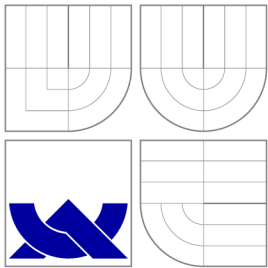
MASTER'S THESIS

AUTOR PRÁCE

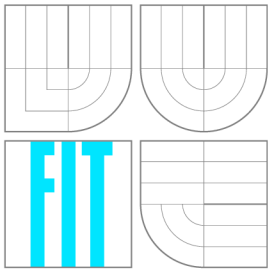
AUTHOR

Bc. LUKÁŠ JANEČKA

BRNO 2008



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

AMBULANTNÍ SYSTÉM ZALOŽENÝ NA IZIP A JEHO ZABEZPEČENÍ

MEDICAL SYSTEM BASED ON IZIP AND ITS SECURITY

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. LUKÁŠ JANEČKA

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. PAVEL JURKA

BRNO 2008

Abstrakt

Tato práce pojednává o analýze, návrhu a implementaci zjednodušeného ambulantního informačního systému praktického lékaře komunikujícího se systémem IZIP. Při návrhu systému je užita architektura Model–View–Controller, návrhové vzory a technika objektového modelování pomocí UML. V práci je také popsán způsob komunikace se systémem IZIP, struktura odesílaných dat a zabezpečení této komunikace. Pro implementaci navrženého systému byl použit programovací jazyk C++, pro zpracování XML dokumentů parser Xerces-C++, pro grafické uživatelské rozhraní toolkit Qt a data jsou ukládána do relační databáze Firebird.

Klíčová slova

IZIP, ambulantní informační systém, databáze, bezpečnost, Model–View–Controller, UML, návrhový vzor

Abstract

The work discusses the analysis, proposal and implementation of simplified outpatient information system of general practitioner (GP), which communicates with IZIP information system. The proposal of system uses the Model–View–Controller architecture; the design patterns and subject simulated technique use UML. The way of communication with IZIP system, the structure of sent data and the communication security is also described in the work. C++ programming language is used for the implementation of the proposed system, parser Xerces-C++ for XML documents processing, toolkit Qt for graphical user's interface; and the data are stored into the Firebird relational database.

Keywords

IZIP, outpatient information system, database, security, Model–View–Controller, UML, design pattern

Citace

Lukáš Janečka: Ambulantní systém založený na IZIP a jeho zabezpečení, diplomová práce, Brno, FIT VUT v Brně, 2008

Ambulantní systém založený na IZIP a jeho zabezpečení

Prohlášení

Prohlašuji, že jsem tento diplomový projekt vypracoval samostatně pod vedením pana Ing. Pavla Jurky. Další informace mi poskytli MUDr. Pavel Janečka a MUDr. Olga Janečková. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

Lukáš Janečka

19. 5. 2008

Poděkování

Děkuji vedoucímu diplomové práce za vstřícnost a mnoho dobrých rad a námětů, a Lucii Čepelákové za pomoc při testování implementovaného systému.

© Lukáš Janečka, 2008.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

| | | |
|----------|--|-----------|
| 1 | Úvod | 3 |
| 2 | Programy a Internet | 4 |
| 3 | Zdravotnický software | 5 |
| 3.1 | Ambulantní informační systém | 5 |
| 4 | Datový standard Ministerstva zdravotnictví ČR | 7 |
| 5 | Zdravotnické portály | 8 |
| 5.1 | IZIP | 8 |
| 5.2 | Komunikace s IZIP | 9 |
| 5.2.1 | Komunikace s IZICHECK | 10 |
| 5.2.2 | Komunikace s IZIVIEW | 10 |
| 5.2.3 | Komunikace s IZIGATE | 10 |
| 5.2.4 | Bezpečnost informací v IZIP | 11 |
| 5.3 | Rizika a slabá místa projektů založených na IZIP | 12 |
| 5.4 | Komunikace s pojišťovnou | 13 |
| 5.4.1 | Písemná komunikace | 13 |
| 5.4.2 | Elektronická komunikace | 14 |
| 6 | Ambulantní informační systém komunikující s IZIP | 16 |
| 6.1 | Analýza požadavků | 16 |
| 6.1.1 | Požadavky obecně | 17 |
| 7 | Návrh ambulantního informačního systému komunikujícího s IZIP | 18 |
| 7.1 | Případy užití | 19 |
| 7.1.1 | Příklady případů užití první iterace | 19 |
| 7.2 | Databáze | 20 |
| 7.2.1 | Konceptuální schéma databáze | 20 |
| 7.2.2 | Schéma databáze | 21 |
| 7.3 | Architektura systému | 22 |
| 7.4 | Bezpečnost | 23 |
| 8 | Implementace | 25 |
| 8.1 | Databáze | 25 |
| 8.1.1 | Datové pohledy | 26 |
| 8.1.2 | Databázové procedury | 26 |
| 8.2 | Datová vrstva | 27 |

| | | |
|----------|--|-----------|
| 8.3 | Vrstva Model | 28 |
| 8.3.1 | Třída <i>Spravce</i> | 28 |
| 8.3.2 | Třídy <i>SpravceUzivatele</i> , <i>SpravcePacientu</i> a <i>SpravceZaznamu</i> | 28 |
| 8.3.3 | Třída <i>SpravceCiselniku</i> | 29 |
| 8.3.4 | Třída <i>System</i> | 30 |
| 8.4 | Vrstva Controller | 30 |
| 8.4.1 | Komunikace se systémem IZIP | 30 |
| 8.4.2 | Odpověď systému IZIP | 32 |
| 8.5 | Vrstva View | 32 |
| 8.6 | Třída <i>SpravceChyb</i> | 34 |
| 8.7 | Architektura systému z pohledu komponent | 34 |
| 8.7.1 | Firebird | 35 |
| 8.7.2 | IBPP | 37 |
| 8.7.3 | OpenSSL | 38 |
| 8.7.4 | Libiconv | 38 |
| 8.7.5 | Xerces-C++ | 38 |
| 8.7.6 | Knihovna IZIP | 38 |
| 8.7.7 | Qt | 39 |
| 8.8 | Demonstrace projektu | 39 |
| 8.9 | Stav implementace projektu | 40 |
| 9 | Závěr | 41 |
| A | Prohlášení o cílech | 45 |
| B | Požadavky na systém | 46 |
| B.1 | Nefunkční požadavky | 46 |
| B.2 | Funkční požadavky | 46 |
| C | Analýza požadavků na systém | 48 |
| C.1 | Diagram případů užití | 48 |
| C.2 | Popis diagramu případů užití první iterace | 52 |
| D | Návrh systému | 68 |
| D.1 | Návrh obrazovek systému | 68 |
| D.2 | Návrh schématu databáze | 70 |
| D.2.1 | Popis schématu databáze | 70 |
| D.2.2 | Diagram schématu databáze | 74 |
| D.3 | Návrh architektury systému | 76 |
| D.3.1 | Návrh jednotlivých vrstev architektury | 77 |
| E | Obsah příloženého disku | 81 |

Kapitola 1

Úvod

Tato práce pojednává o vývoji ambulantního systému, který je schopen komunikovat se systémem IZIP. V úvodu práce je vysvětleno, jaké typy softwaru a za jakými účely jsou užívány ve zdravotnictví, a popsáno zařazení ambulantního systému do uvedeného rozdělení.

Protože navrhovaný systém má komunikovat s okolními subjekty pomocí sítě internet, jsou v práci probrány principy a způsoby komunikace programů v této celosvětové síti.

Jelikož v České republice je mnoho softwarových firem, které vyvíjejí produkty pro zdravotnictví, bylo potřeba stanovit standard, kterým se bude řídit výměna informací mezi těmito systémy. K tomuto účelu byl sestaven Datový standard Ministerstva zdravotnictví České republiky, který je v práci také zmíněn.

Další kapitola práce pojednává o subjektech, s nimiž ambulantní systém může komunikovat. Část této kapitoly je vyhrazena pro popis systému IZIP a bezpečnost systémů založených na IZIP.

Za touto kapitolou následují kapitoly s návrhem amulantního systému a jeho implementací.

V rámci semestrálního projektu byla provedena analýza a návrh tohoto systému. V této diplomové práci jsou ze semestrálního projektu využity kapitoly zabývající se touto problematikou a také k těmto částem náležející přílohy.

Pro projekt ambulantního informačního systému jsem se rozhodl z důvodu velkého zájmu o velmi levný, jednoduchý program do ordinace praktického lékaře, který projevili oslovení praktičtí lékaři. Poptávaný program měl umět také komunikovat se systémem IZIP, jehož užívání bylo v té době zvýhodněno různými bonusy.

Kapitola 2

Programy a Internet

Podle [1] je Internet celosvětová počítačová síť, která spojuje jednotlivé menší sítě pomocí sady protokolů IP. Komunikace v této síti je řízena protokoly, které jsou podle vykonávané funkčnosti děleny do čtyř vrstev. Protokolem se podle [2] rozumí soubor syntaktických a sémantických pravidel určujících výměnu informace mezi nejméně dvěma entitami. Toto dělení vychází z referenčního modelu ISO/OSI. Mezi tyto čtyři vrstvy patří:

vrstva síťového rozhraní – definuje standardy pro fyzická média, elektrické signály a rutiny pro přístup k fyzickému médiu,

síťová vrstva – vytváří datagramy, adresuje je a směřuje na místo určení,

transportní vrstva – předává data mezi koncovými uzly,

aplikační vrstva – je tvořena procesy a aplikacemi, které komunikují po síti.

Z hlediska programátora jsou důležité poslední dvě jmenované vrstvy, na jejichž rozhraní se nachází tzv. sockety (schránky), což je aplikační programové rozhraní (API) pro komunikující procesy po síti. Toto rozhraní je implementováno v knihovnách pro různé programovací jazyky i pro různé operační systémy (např. BSD socket API pro unixové operační systémy a Windows Sockets API pro operační systémy MS Windows).

Pro komunikaci mezi dvěma uzly v síti jsou k dispozici dvě základní schémata:

- klient–server,
- klient–klient.

Schéma klient–server rozlišuje účastníky komunikace na klienta, který posílá požadavky na zpracování a (obvykle) iniciuje komunikaci, a server, který čeká na požadavky, zpracovává je a zasílá odpověď. Při užití tohoto schématu se v návrhu rozlišuje návrh klientské části aplikace od serverové.

Ve schématu klient–klient nelze účastníky rozlišit, protože mohou simultánně vystupovat v obou rolích.

Uzly (aplikace) mezi sebou komunikují pomocí aplikačního protokolu. Tento mohou mít svůj vlastní nebo využívají některého ze standardizovaných (např. HTTP, IMAP, SMTP).

Kapitola 3

Zdravotnický software

Zdravotnictví je segment, v němž trvale vzniká velké množství informací. Tyto informace vznikají na různých místech a při použití konvenčních metod zápisu (chorobopisy, lékařské zprávy, ...) dochází k jejich fragmentaci. Velký problém představuje také doba jejich přesunu (ať už mezi různými specializovanými pracovišti, nebo například při přestěhování pacienta).

Řešením tohoto problému je digitalizace těchto informací a využití elektronické komunikace pro jejich přesun. Zdigitalizovaná informace nic „neváží“, objemově zabírá mnohem menší prostor než informace „papírová“, dá se mnohem lépe zabezpečit (a to jak proti neoprávněnému čtení, tak proti zničení) a při užití elektronické komunikace může být kdykoliv k dispozici. Pro digitalizaci a správu informací ve zdravotnictví slouží specializovaný software – zdravotnický software.

Zdravotnický software můžeme rozdělit do skupin podle oblastí jeho užití, každá oblast má na tento software jiné specifické nároky. Tyto skupiny mohou být následující (převzato z [22]):

- Ambulantní informační systémy – slouží pro samostatné a poliklinické ambulance.
- Nemocniční klinické informační systémy – hlavními úkoly jsou administrativa, ambulance, lůžka, výkaznictví a jiné.
- Laboratorní informační systémy
- Lékárenské informační systémy
- Manažerské a ekonomické informační systémy – zaměřeny na provoz zdravotnického zařízení.
- Jiné speciální produkty.

3.1 Ambulantní informační systém

Tato práce se zabývá informačním systémem ambulantním, jehož hlavními úkoly je správa zdravotních záznamů, výkaznictví a administrativa. Tyto systémy se dále rozlišují podle lékařské specializace (gynekologické, stomatologické, oftalmologické – každá specializace má jiné nároky na systém), některé z nich jsou navrženy tak, že je lze provozovat pro více odborností. V práci se budu zabývat ambulantním informačním systémem se zaměřením pouze pro praktické lékaře, protože rozšíření pro více odborností systém značně komplikuje.

Před zahájením práce na projektu jsem se seznámil s nabídkou dostupných produktů s podobným zaměřením. Zjistil jsem, že v nabídce jsou pouze produkty komerční, žádný produkt vyvíjený komunitou nebo produkt s otevřeným zdrojovým kódem, či „volnou“ licencí se na území České republiky nevyskytuje. (Otázku použitelnosti takového systému zde diskutovat nebudu.) Mezi ambulantní informační systémy patří například 3L, PC DOKTOR, AMBULANCE, MEDICUS, AMICUS.

Všechny mnou zkoumané produkty měly stejný model licencování. Uživatel nejprve zaplatí jednorázovou částku, za kterou obdrží systém a s ním i podporu, které je omezena na určitou dobu. V rámci této podpory dostává uživatel aktualizace programu zdarma. Avšak po uplynutí této doby je uživatel již nucen další aktualizace programu zaplatit, přičemž většina aktualizací programu se skládá z aktualizací číselníků, které se vesměs dají z internetu stáhnout zdarma (například číselníky léčiv, které spravuje Všeobecná zdravotní pojišťovna).

Další negativní vlastností těchto systémů je špatná podpora přechodu mezi jednotlivými produkty. Jakmile uživatel začne používat jeden produkt a má v jeho interních strukturách všechna data, je přechod na jiný produkt velmi komplikovaný, někdy až nemožný. Některé z firem nabízejí převedení dat z jiného systému do svého, cena tohoto převodu však mnohdy překročí cenu vlastního software.

Aby tento text nevyzněl jako kritika ambulantních systémů na trhu, musím ho uzavřít konstatováním, že na trhu je také mnoho kvalitních systémů, které mají mimo těchto několika negativ mnoho kladných vlastností.

Kapitola 4

Datový standard Ministerstva zdravotnictví ČR

Kvůli značné rozmanitosti informačních systémů užívaných ve zdravotnictví a potřebě sjednocení pravidel pro komunikaci mezi nimi, byl sestaven Datový standard Ministerstva zdravotnictví České republiky (dále jen DASTA).

Práce na tomto standardu započaly v roce 1997 a podíleli se na něm hlavní dodavatelé zdravotnických informačních systémů v ČR. Při návrhu standardu byla snaha zohlednit jak existující mezinárodní standardy (např. Health Level 7), tak potřeby poskytovatelů zdravotní péče. Práce na datovém standardu byly podle [19] dovršeny v roce 2002 vydáním verze 02.01.01.

Mezi hlavní vlastnosti DASTA patří:

- přenositelnost dat (zajištěna pomocí textového formátu XML),
- univerzálnost (lze použít ve všech zdravotnických systémech),
- garance dat (u všech dat lze zjistit jejich původ),
- plnohodnotnost (lze zapsat všechny typy zdravotních záznamů).

Datový standard prochází neustálým vývojem, v prosinci 2006 byla vydána verze 4.01.01 a následující verze se snaží o harmonizaci s mezinárodním standardem Health Level 7.

Kapitola 5

Zdravotnické portály

Mezi zdravotnické portály (portály, se kterými komunikuje zdravotnický software či pracovník) patří portály pojišťoven, portál Ústavu zdravotnických informací a statistiky České republiky, portály laboratoří, IZIP a jiné. Ne všechna komunikace podléhá Datovému standardu MZ ČR, příkladem mohou být laboratoře, které využívají pro komunikaci interních datových standardů svých informačních systémů (např. MEDEA II, DUMP_2 a jiné).

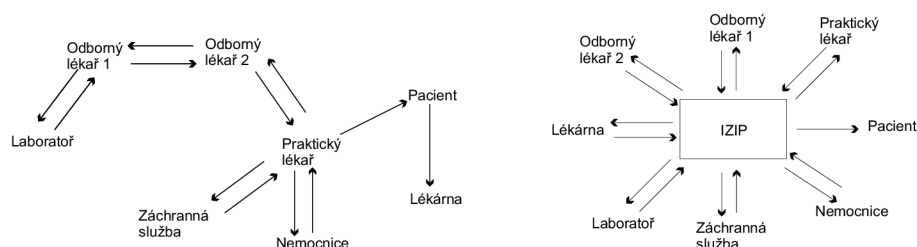
Kromě komunikace se zdravotnickými portály se uplatňuje také komunikace pomocí e-mailu (hojně využívaná zvláště laboratořemi), kdy jsou data zasílána zašifrovaná a podepsaná elektronickým podpisem.

5.1 IZIP

Systém IZIP, neboli Internetový přístup ke Zdravotnickým Informacím Pacienta, si podle [21] klade za hlavní cíle:

- zlepšení kvality poskytované péče,
- zefektivnění výměny zdravotních informací,
- zrychlení a zkvalitnění rozhodování zdravotnických pracovníků,
- uspořené finančních prostředků.

Těchto cílů chce dosáhnout díky poskytnutí zdravotních informací o pacientovi kdykoli je to potřeba a tam, kde je to potřeba. K tomu využívá celosvětové sítě Internet. Transport informací užitím systému IZIP a dosavadního řešení znázorňuje obrázek 5.1, přejatý z [20].



Obrázek 5.1: Předávání informací bez IZIP a s IZIP

Zdravotní informace jsou shromažďovány do tzv. Zdravotních knížek pacientů. Zdravotní knížka vzniká na základě žádosti pacienta a je jím vlastněna. Pacient tak má plný přístup ke svým záznamům, může je komentovat a podle [17] dokonce i požádat o jejich výmaz. (V takovém případě zůstane ve Zdravotní knížce informace o vymazání záznamu.)

Ke zdravotním záznamům pacienta mají přístup pouze pacientem vybraní lékaři a zdravotnická pracoviště. Výjimku tvoří tzv. *emergentní přístup*, kdy zdravotní pracovník může získat ze zdravotní knížky urgentní informace (krevní skupina, alergie a podobně).

Emergentní přístup je přístup do systému, který je umožněn pouze pracovníkům záchranné služby. Při tomto přístupu není vyžadováno heslo pacienta, ale je nutné, aby pracovník záchranné služby, stejně jako pracoviště záchranné služby, byl v systému IZIP řádně registrován.

5.2 Komunikace s IZIP

IZIP – internetový přístup ke zdravotním informacím pacienta je tvořen portálem, ke kterému přistupují jak lékaři, tak pacienti (klienti). Lékař smí jednorázově nahlédnout do záznamů pacienta, který mu k tomuto účelu sdělí své přístupové heslo. Pro trvalé nahlížení do záznamů musí klient v IZIP označit tohoto lékaře jako důvěrného, toto označení může později odebrat.

Pro lepší spolupráci mezi lékařským softwarem a IZIP byly zpřístupněny tři brány umožňující vzájemnou komunikaci:

IZIGATE – pro příjem informací do Zdravotních knížek,

IZICHECK – pro ověření existence klienta v databázi IZIP,

IZIVIEW – pro snadné nahlížení do konkrétní Zdravotní knížky klienta.

Komunikace je řízena zabezpečeným protokolem HTTPS a sestává ze čtyř fází:

- ustavení spojení,
- odeslání HTTP požadavku,
- obdržení odezvy,
- ukončení spojení.

Obsahem HTTP požadavku jsou přístupové údaje k IZIP (identifikace uživatele, přístupové heslo) a data ve formátu XML, který je podle [16] kompatibilní s Datovým standardem Ministerstva zdravotnictví ČR verze 03.01.01 (viz [18]).

Obsahem obdržené odezvy jsou rovněž data ve formátu XML, výjimkou je zadání špatného uživatelského jména či hesla resp. nesplnění vstupních podmínek (např. limit velikosti). V takovém případě vrací server HTTP odpověď s hlavičkou 400 resp. 403 a tělem obsahujícím text 40001 resp. 40301.

5.2.1 Komunikace s IZICHECK

Brána IZICHECK slouží pro ověření existence klienta v databázi IZIP. Toto ověření by mělo předcházet vlastním odeslání dat na bránu IZIGATE, čímž se zamezí odeslání informací o klientech, kteří nemají svou Zdravotní knížku.

Komunikace probíhá posláním dotazu s identifikačními čísly klientů, kteří se mají prověřit, ve formátu XML, na který brána odpoví identifikacemi klientů v systému obsaženými. Také je možné získat informace o počtu vyšetření klienta provedených odesílajícím zdravotnickým pracovníkem, jinými nebo všemi zdravotnickými pracovníky. Do odpovědi lze zahrnout informaci o tom, zda si daný klient svou Zdravotní knížku aktivoval.

5.2.2 Komunikace s IZIVIEW

Brána IZIVIEW byla vytvořena pro rychlé a snadné nahlížení do Zdravotní knížky klientů IZIP z jiného zdravotnického software. Od tohoto software se předpokládá implementace IZICHECK v plném rozsahu a ve spojení s IZIVIEW umožní efektivní práci s IZIP. Zdravotní pracovník získá přehled o nových záznamech ve Zdravotní knížce klienta a její otevření bude realizováno jediným stisknutím tlačítka.

Na bránu IZIVIEW se tedy odesílají identifikační údaje zdravotnického pracovníka, informace o jeho zdravotním software a identifikační údaje pacienta, do jehož Zdravotní knížky hodlá pracovník nahlédnout. IZIVIEW žádost zpracuje a v požadavku vrací HTML kód WWW stránky se Zdravotní knížkou pacienta.

5.2.3 Komunikace s IZIGATE

Brána IZIGATE slouží pro zasílání informací do Zdravotních knížek pacientů. Mezi tyto informace patří anamnéza, očkování, předepsané či vydané léky, zprávy z ambulantního vyšetření či hospitalizace, urgentní informace a výsledky laboratorních vyšetření. Následuje stručný popis informací, které umí systém IZIP zobrazit:

- Záznam Anamnéza je tvořen pouze textem, který je možné strukturovat dle vlastního uvážení.
- Záznam Očkování je tvořen určením data a času aplikace a třemi textovými popisy. První popisuje typ očkování, druhý očkovací látku (název, číslo šarže, podané množství a místo aplikace) a třetí, volitelný, je vyhrazen pro určení diagnózy.
- Záznam Ambulantní vyšetření se skládá z části Závěr, zahrnující podstatné informace o poskytnuté péči a jejích výsledcích, volitelného pole Diagnóza, udávajícího kód diagnózy, a dále z pole Indikovaná vyšetření, pro specifikaci vyšetření, o jejichž provedení lékař rozhodl, ale sám je neprovádí, Terapie (mimo léky), pro popis údajů o léčbě, kterou lékař provedl nebo doporučil (např. ošetření rány, provedená nebo doporučená punkce, doporučená dieta apod.) a Léky, pro zaznamenání předepsaných léčiv (zahrnuje název léku/zdrav. prostředku, dávkování, počet balení/kusů a kód léku/zdrav. prostředku). Poslední pole je vyhrazeno pro libovolný dokument, který lze připojit jako přílohu.
- K Urgentním informacím mají přístup mimo klientem vybrané lékaře také lékaři rychlé záchranné služby. Urgentní informace sestávají z polí Alergie, Krevní skupina, Rizikové faktory, Trvalá medikace a Datum posledního očkování tetanu.

- Záznam Hospitalizace je velmi podobný záznamu Ambulantní vyšetření. Obsahuje navíc pouze pole pro zadání kódů vedlejších diagnóz.

Systém IZIP umožňuje příjem dat zasílaných v XML souborech podléhajících standardu MZ ČR 03.01.01 buďto ve formátu „vše v jednom“, kdy jsou všechny záznamy vloženy do jednoho souboru, nebo formátu „každý zvlášť“, kdy je každý záznam v samostatném souboru.

Při zjištění chyby v souboru s více záznamy je zpracování tohoto souboru přerušeno a všechny doposud načtené záznamy jsou odmítnuty.

Zasílaná data musí respektovat limity na velikost. Tyto limity jsou:

- velikost každého jednotlivého souboru nesmí přesáhnout 2MB,
- celková velikost kolekce POST nesmí přesáhnout 8MB.

5.2.4 Bezpečnost informací v IZIP

Následující text vychází z [15]. Bezpečnost systému IZIP lze rozdělit do tří úrovní:

Bezpečnost na úrovni pacienta a lékaře

- Zdravotní knížka patří pacientovi a pouze on rozhoduje o tom, kterým lékařům ji zpřístupní. Bez souhlasu pacienta nemůže záznamy v knížce nikdo prohlížet (výjimkou je tzv. emergentní přístup, který umožňuje číst urgentní informace z pacientovy knížky).
- Pokud chce pacient v knížce číst, musí zadat své přístupové údaje, tj. identifikační číslo (s jeho svolením je to rodné číslo) a heslo, které je mu zasláno v doporučeném dopise do vlastních rukou. Může zadat i osobní heslo, které si může navíc sám vytvořit.
- Přístupová hesla generuje program zcela náhodně, žádný z pracovníků společnosti IZIP nemá možnost je zjistit. Heslo se vytiskne a zabalí do zvláštní obálky, podobně jako PIN u bankovních karet. Tato obálka je zaslána klientovi doporučeně do vlastních rukou. Klient má vždy možnost požádat o změnu hesla.
- Pokud se do knížky svého pacienta chce podívat lékař, musí zadat své přístupové údaje.
- Nahlížet do záznamů pacienta mohou jen ti lékaři, kterým to pacient umožní, tj. kterým udělí oprávnění k přístupu. Pokud pacient neudělí lékaři oprávnění, nemůže lékař v pacientově knížce číst. Své závěry z vyšetření může přesto do IZIP zapsat.

Bezpečnost na úrovni softwaru, počítačového vybavení a zařízení

- Systém IZIP využívá nejmodernější technologické prostředky proti zneužití dat.
- Zdravotnická data a jejich přenos jsou zabezpečeny na stejných principech, jako jsou zajišťovány internetové bankovní operace.
- Po zadání přihlašovacích údajů probíhá autentizace a autorizace (zjištění a ověření) požadavku na přístup k datům.

- Server zkontroluje, zda uživatel, který se přihlašuje, splňuje všechny podmínky pro to, aby mohl v knížce číst nebo do ní zapisovat.
- Pokud tomu tak je, server zpřístupní uživateli (lékaři nebo pacientovi) ty záznamy, ke kterým má dle svého oprávnění přístup.
- Pokud uživatel provede více chybných přihlášení, je jeho přístup zablokován a je nutné kontaktovat administrátory společnosti IZIP.
- Firewally jsou jedním z mnoha použitých bezpečnostních zařízení. Zabraňují neoprávněným přístupům a hlídají pokusy o průnik do systému.
- Administrátoři společnosti IZIP kontrolují, zda jsou servery v pořádku, vyhodnocují přístupy a v případě potřeby podnikají kroky k zabránění neoprávněným přístupům.
- S vlastní databází pracují pouze její administrátoři, kteří pracují v odděleném patře s elektronicky zajištěným vstupem. Jedná se o oprávněné a speciálně vyškolené osoby.
- Data na vlastním serveru jsou zašifrována.

Bezpečnost na úrovni fyzického uložení serverů s daty

- Servery jsou uloženy na bezpečném místě v podzemí a jsou fyzicky hlídány 24 hodin denně.
- Přístup k těmto serverům mají pouze oprávněné osoby, které musí vždy doprovázet pověřený pracovník společnosti IZIP.
- Každý z nich má jeden bezpečnostní klíč k opancérovaným protipožárním dveřím. K jejich otevření je potřeba použít oba klíče zároveň.

5.3 Rizika a slabá místa projektů založených na IZIP

Pro nastudování problematiky bezpečnosti informačních systémů jsem použil [14], ze které zde odcituji definice základních bezpečnostních pojmů:

Zranitelné (slabé) místo je slabina v informačním systému, která je využitelná ke způsobení škod. Existence těchto míst je důsledkem chyb v analýze, návrhu a/nebo v implementaci, složitosti softwaru, existence skrytých kanálů pro přenos informace jinou než zamýšlenou cestou a podobně.

Riziko je tvořeno kombinací slabého místa a bezpečnostní hrozby.

Hrozbou označujeme možnost využít zranitelné místo IS k útoku na něj – ke způsobení škody na aktivech. Hrozby mohou být:

1. neúmyslné – živelné události, poruchy zařízení, chyby v software, selhání osob,
2. úmyslné – krádež HW, úmyslné poškození zařízení, krádež dat, krádež SW, neoprávněná manipulace s daty.

Bezpečností rozumíme ochranu informačních systémů a informací, které jsou v nich uchovávány, zpracovávány a přenášeny. V soudobém chápání bezpečnosti IT je bezpečnost dána zajištěním:

- důvěrnosti** – ochrana proti neoprávněnému prozrazení informace,
- integrity a authenticity** – ochrana proti neoprávněné modifikaci informace,
- dostupnosti** – ochrana proti neoprávněnému odepření přístupu k datům nebo službám.

V systémech založených na IZIP jsou uložena důvěrná data (data pacientů). Musíme tedy zajistit jejich ochranu proti neautorizovanému čtení (zabezpečit přístup k datům pouze autorizovanými subjekty, zajistit komunikaci se systémem IZIP proti odposlechu), proti neautorizované modifikaci (modifikovat data mohou pouze autorizované subjekty) a v poslední řadě musíme zajistit, aby k datům měli přístup všechny autorizované subjekty.

5.4 Komunikace s pojišťovnou

Podle [4] jsou osoby, které mají trvalý pobyt na území České republiky, a osoby, které na území České republiky trvalý pobyt nemají, ale jsou zaměstnanci zaměstnavatele, který má sídlo na území České republiky, zdravotně pojištěny u některé zdravotní pojišťovny.

Podle číselníku VZP verze 640 (platnost od 1. 10. 2007–31. 12. 2007) (viz [27]) je v České republice jedenáct zdravotních pojišťoven. Jejich seznam je v tabulce 5.1.

| Kód | Název |
|-----|--|
| 111 | Všeobecná zdravotní pojišťovna ČR |
| 201 | Vojenská zdravotní pojišťovna ČR |
| 205 | Hutnická zaměstnanecká pojišťovna |
| 207 | Oborová zdravotní pojišťovna zaměstnanců bank a pojišťoven |
| 209 | Zaměstnanecká pojišťovna ŠKODA |
| 211 | Zdravotní pojišťovna Ministerstva vnitra ČR |
| 212 | Stavební zdravotní pojišťovna |
| 213 | Revírní bratrská pokladna |
| 217 | Zdravotní pojišťovna METAL-ALIANCE |
| 222 | Česká národní zdravotní pojišťovna |
| 333 | Pojišťovna VZP, a.s. |

Tabulka 5.1: Seznam zdravotních pojišťoven

Pojišťovna 333 – Pojišťovna VZP, a.s. je dceřinou společností Všeobecné zdravotní pojišťovny ČR a poskytuje zdravotní pojištění pouze pro cizince.

Pojišťovna 212 – Stavební zdravotní pojišťovna byla sloučena s pojišťovnou 207, která se nyní jmenuje Oborová zdravotní pojišťovna zaměstnanců bank, pojišťoven a stavebnictví.

Náplň komunikace mezi lékařem a pojišťovnou tvoří např. vykazování poskytnuté zdravotní péče, zasílání faktur za poskytnutou péči, ověřování zdravotní pojišťovny pacienta, ověřování registrace pacienta k ošetřujícímu lékaři.

Komunikace může probíhat těmito způsoby: písemně nebo elektronicky, a to elektronickou poštou, portálem nebo pomocí datového nosiče.

5.4.1 Písemná komunikace

Dnešním trendem je omezování písemné komunikace. Některé služby poskytuje pojišťovna dokonce pouze elektronicky (např. podá-li si praktický lékař s kapitační platbou písemnou

žádost na seznam registrovaných pacientů z důvodu prověření registrace pojištěnce u tohoto lékaře, je mu odpověď doručena pouze na datovém nosiči).

Z [25] plyne, že pro písemnou komunikaci s pojišťovnou se užívají předepsané tiskopisy nebo počítačem tištěné výstupy, jejichž datový obsah a formální členění odpovídá příslušnému předepsanému tiskopisu. Některé tiskopisy lze nalézt na stránkách pojišťovny (viz [26]), jiné pocházejí z nakladatelství SEVT.

5.4.2 Elektronická komunikace

Elektronická komunikace podléhá datovému rozhraní, toto rozhraní je pro všechny pojišťovny ČR stejné. Jeho popis lze nalézt v [24]. Data předávaná přes toto rozhraní jsou tvořena ASCII soubory, které jsou složeny z vět různých typů a pro jejichž skladbu jsou definována pravidla. Jednotlivé věty jsou odděleny znakem pro nový řádek a jsou tvořeny hodnotami atributů rozlišenými oddělovači.

Je-li elektronická komunikace uskutečněna pomocí datového nosiče, je k tomuto nosiči připojen formulář Průvodní list datového nosiče, na kterém je razítko a podpis odesílatele.

Při elektronické komunikaci pomocí elektronické pošty je nutné opatřit zprávu zaručeným elektronickým podpisem (podpisem založeným na kvalifikovaném certifikátu).

Kvalifikovaný certifikát – certifikát, který má náležitosti stanovené zákonem o elektronickém podpisu a byl vydán poskytovatelem certifikačních služeb splňujícím podmínky stanovené tímto zákonem pro poskytovatele certifikačních služeb vydávající kvalifikované certifikáty. (převzato z [3])

Pro komunikaci prostřednictvím poslední možnosti elektronické komunikace – portálovým řešením – je potřeba vlastnit dva certifikáty:

1. komerční – pro ověření totožnosti uživatele při přihlašování k Portálu a pro šifrování jeho spojení s Portálem.
2. kvalifikovaný – pro elektronické podepisování dat zasílaných přes portál.

Momentálně se toleruje vlastnictví pouze jednoho z těchto certifikátů. Portál neumožňuje komunikaci se všemi pojišťovnami v ČR. Některé pojišťovny portál nemají vůbec, jiné mají svůj vlastní. Současnou situaci vystihuje tabulka 5.2.

| Kód | Portál |
|-----|---|
| 111 | http://www.vzp.cz/cms/internet/cz/Vseobecne/Portal/ |
| 201 | http://www.portalzp.cz/index.html |
| 205 | http://www.hzp.cz/prepazka/ |
| 207 | http://www.portalzp.cz/index.html |
| 209 | http://www.portalzp.cz/index.html |
| 211 | http://eforms.zpmvcr.cz/ |
| 213 | http://www.portalzp.cz/index.html |
| 217 | http://www.portalzp.cz/index.html |
| 222 | http://www.portalzp.cz/index.html |

Tabulka 5.2: Seznam portálů zdravotních pojišťoven

Jak je patrné z tabulky 5.2, šest pojišťoven (201, 207, 209, 213, 217, 222) sdílí jediný portál. K tomuto portálu je také zdarma poskytována knihovna, která umožňuje komunikaci mezi lékařským softwarem a tímto portálem, demonstrační aplikace a dokumentace (viz [7]).

Některé ze služeb nabízených poskytovatelům péče portály pojišťoven:

- ověření aktuální registrace pojištěnce u zdravotní pojišťovny
- vyhledání zdravotnického zařízení ve smluvním vztahu k pojišťovně
- vyhledání informace o registraci pojištěnce u jeho ošetřujícího lékaře
- zasílání faktur za poskytnutou zdravotní péči
- předávání souborů vyúčtování zdravotní péče poskytnuté pojištěncům
- elektronická podatelna

Kapitola 6

Ambulantní informační systém komunikující s IZIP

6.1 Analýza požadavků

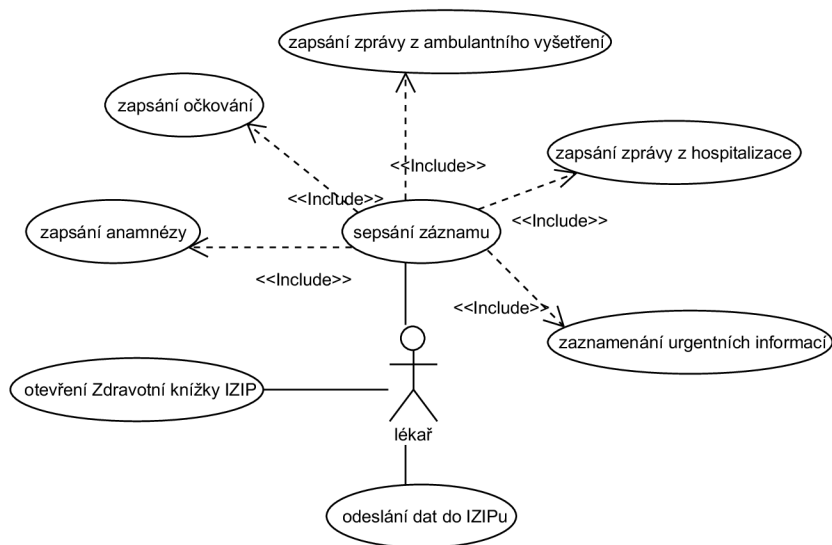
Ambulantní informační systém komunikující se systémem IZIP by měl obsahovat minimálně takové funkce, aby byl schopen do systému IZIP odeslat všechny typy dat, které je tento systém schopen zpracovat, a zároveň by tento úkon měl uživateli co nejvíce usnadnit.

Odesílání záznamů do IZIP je tedy hlavní funkcí tohoto systému. Další funkce se budou od této hlavní funkce odvíjet:

- Odesíláme-li data pacientů, musíme v systému mít funkce pro správu pacientů (funkce kartotéky).
- Vlastnímu odeslání dat do systému IZIP by měla předcházet kontrola aktivity zdravotní knížky pacienta (funkce IZICHECK).
- Odesíláme-li data do IZIP, chceme si také zobrazit data od jiných lékařů (funkce IZIVIEW).
- Abychom mohli záznamy odeslat, musíme nějaké mít:
 - zápis anamnézy pacienta,
 - zapsání urgentních informací o pacientovi,
 - zapsání očkování pacienta,
 - zapsání zprávy z ambulantního vyšetření,
 - zapsání zprávy z hospitalizace (pokud pracoviště tuto službu provádí),
 - zapsání přechodné/trvalé nově indikované diagnózy.

Z těchto funkčních požadavků lze sestavit diagram případů užití (viz obrázek 6.1).

Některé z výše jmenovaných funkcí v sobě zahrnují další funkce, jako například Zapsání zprávy z ambulantního vyšetření, jehož součástí může být stanovení nové diagnózy, předepsání léčiv, indikace vyšetření či terapie.



Obrázek 6.1: Diagram případu užití systému komunikujícího s IZIP

6.1.1 Požadavky obecně

Z pohledu lékaře je ale hlavním požadavkem na ambulantní informační systém výkaznictví a následné vyúčtování péče. Pro značnou časovou náročnost práce na výše vyjmenovaných případech užití ambulantního informačního systému komunikujícího s IZIP a také rozsáhlost a komplikovanost výkaznictví a vyúčtování lékařské péče, byla tato problematika zařazena až do dalších iterací práce na projektu.

Mezi další požadavky patří podpora pro vyplňování tiskopisů (lázeňská péče, výpisy ze zdravotní dokumentace, hlášení úrazů, tiskopisy České správy sociálního zabezpečení, žádanky a další), vedení dispenzární péče pacientů a další.

V poslední době je velký důraz kladen také na import a export dat. Mnohdy je lékař se svým dosavadním systémem nespokojen a chce přejít na jiný.

Další hodnotné požadavky na ambulantní informační systém lze získat z publikace [23].

Při analýze ambulantního informačního systému jsem spolupracoval s několika praktickými lékaři pro dospělé, které jsem situoval do role zadavatelů fiktivního projektu. Pokusil jsem se tedy analyzovat a navrhnout informační systém pro potřeby praktických lékařů. V úvodu práce na projektu jsem prověřoval možnost vyvíjet systém pro více odborností, avšak návrh takového systému se velmi zkomplikoval, neboť potřeby jednotlivých odborností jsou velmi rozmanité a mohou být i navzájem disjunktní.

Kapitola 7

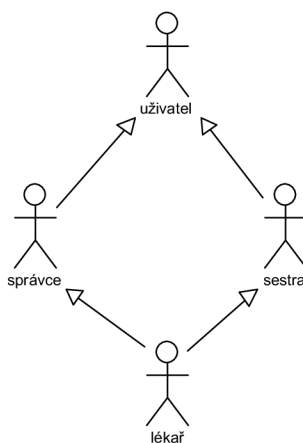
Návrh ambulantního informačního systému komunikujícího s IZIP

Na základě výsledků analýzy jsem navrhl ambulantní systém. Systém jsem koncipoval jako víceuživatelský s přístupem tří různých typů aktérů, kteří se liší v užití systému. Při modelování případů užití jsem zavedl ještě čtvrtého aktéra (Uživatel), který je pouze abstraktní a ostatní aktéři dědí jeho vlastnosti.

Data v databázi jsou z principu přístupná všem – což usnadní práci lékařům v případě, že pacient přechází mezi jednotlivými lékaři systému nebo se jedná o zástup lékaře. Pokud lékaři nebudou chtít sdílet kartotéku pacientů, lze tento případ řešit vlastním databázovým souborem pro každého lékaře, nebo specifikováním přístupových práv (lékař by směl nahlížet do záznamů pouze svých registrovaných pacientů).

Aby se zabránilo neoprávněným přístupům k datům, budou v systému specifikována oprávnění jednotlivých uživatelských skupin (později může být rozšířeno i na jednotlivé uživatele) pro spouštění jednotlivých případů užití.

Diagram s aktéry systému je znázorněn na obrázku 7.1.



Obrázek 7.1: Aktéři systému

Při návrhu systému bylo použito modelovacích technik UML. Návrh probíhal v těchto krocích:

- sestavení prohlášení o cílech,

- sestavení požadavků na systém,
- sestavení případů užití,
- popis případu užití,
- sestavení schématu databáze,
- popis schématu databáze,
- návrh obrazovek systému,
- návrh architektury systému,
- návrh tříd jednotlivých vrstev architektury.

Výsledné dokumenty všech fází lze nalézt v příloze.

7.1 Případy užití

Při sestavování případů užití byly případy užití rozděleny do čtyř iterací. V první iteraci jsou ty případy užití, které umožní zápis těch dat do systému, která následně mohou být odeslána do systému IZIP. Tyto případy užití budou implementovány v rámci této diplomové práce.

Ve druhé iteraci jsou případy užití pro správu uživatelů, správu pacientů a různé administrativní úkoly lékaře. Třetí iterace zahrnuje vyúčtování, poslední iterace zahrnuje import a export dat, komunikaci s pojišťovnou a laboratořemi.

7.1.1 Příklady případů užití první iterace

| | |
|------------------------------|--|
| Případ užití: | Vypsání zdravotního záznamu (dekurzu) |
| ID: | UC1I03 |
| Stručný popis: | Zapsání záznamu z návštěvy pacienta v ordinaci, či lékaře u pacienta. |
| Hlavní aktéři: | lékař |
| Vedlejší aktéři: | žádní |
| Vstupní podmínky: | otevřená karta pacienta, práva pro zápis záznamu |
| Hlavní scénář: | Případ užití je spuštěn otevřením karty pacienta. Je tedy vždy vytvořen nový záznam. Tento záznam je strukturovaný a skládá se z dalších připojených dokumentů. Má-li pacient aktivovanou knížku v IZIP, označí systém po jeho dokončení tento záznam jako nový záznam pro odeslání. |
| Výstupní podmínky: | nový záznam v kartě pacienta |
| Alternativní scénáře: | žádné |

| | |
|------------------------------|---|
| Případ užití: | Úprava identifikačních údajů pacienta |
| ID: | UC1I05 |
| Stručný popis: | Oprava některého z identifikačních údajů pacienta z důvodu zjištění chyby nebo změny |
| Hlavní aktéři: | lékař |
| Vedlejší aktéři: | žádní |
| Vstupní podmínky: | otevřená karta pacienta, práva pro změnu údajů |
| Hlavní scénář: | Případ užití je spuštěn změnou některého z identifikačních údajů v kartě pacienta. Dokud je měněný údaj neplatný, systém žádá uživatele, aby zadal platný údaj. Systém porovná novou hodnotu s původní, liší-li se tyto hodnoty, přidá původní hodnotu do historie. |
| Výstupní podmínky: | Změněný údaj u pacienta, nová hodnota v historii měněné. |
| Alternativní scénáře: | Neplatná hodnota měněného identifikačního údaje. Storno. |

7.2 Databáze

Pro uložení persistentních dat bude systému sloužit databáze. S přihlédnutím ke svým dosavadním zkušenostem a nabídce databázových systémů jsem zvolil databázi relační.

7.2.1 Konceptuální schéma databáze

Na obrázku 7.2 je zobrazen ER diagram navrhovaného systému, jehož popis následuje.

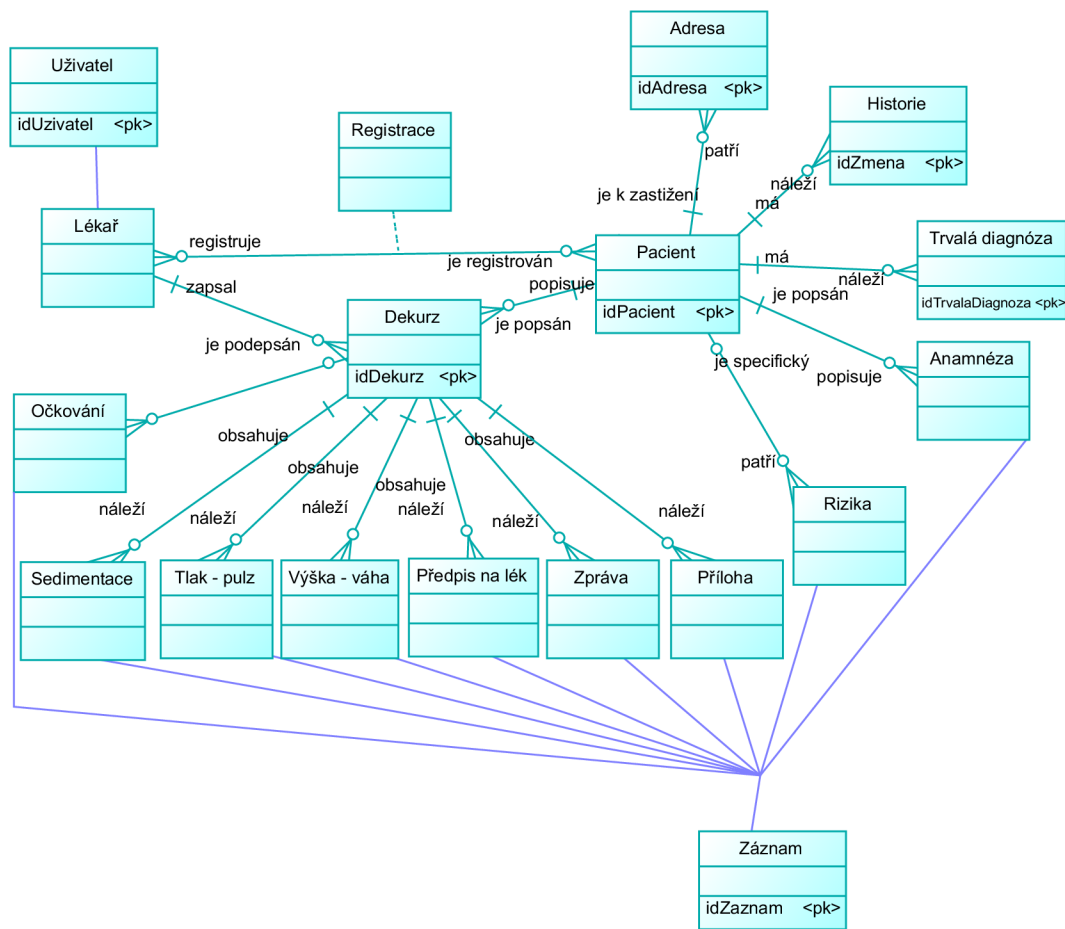
Se systémem budou pracovat tři skupiny uživatelů:

- skupina *správce*, jejímž hlavním úkolem bude nastavování vlastností systému a servisní zásahy,
- skupina *sestra*, která bude mít omezený přístup k vykonávání některých akcí a hlavním úkolem této skupiny bude nahlížení do zdravotních záznamů pacientů,
- skupina *lékař*, která je hlavní uživatelskou skupinou.

Uživatelské skupiny *správce* a *sestra* budou modelovány entitou *Uživatel* a budou rozlišeny pomocí atributu *urole*. Uživatelská skupina *lékař* bude modelována pomocí entity *Lékař* a to z toho důvodu, že k ní potřebujeme namodelovat mnohem více atributů, než k ostatním skupinám. Mezi entitami *Lékař* a *Uživatel* je vztah specializace (atributy, které má entita *Uživatel*, má také entita *Lékař*).

Systém bude obsahovat zdravotní záznamy pacientů – entita *Pacient*. Každý pacient má identifikační údaje, jejichž změny chceme v systému také zachytit, entita *Historie*, a také může pobývat na více adresách (např. nemocná babička se dočasně přestěhuje ke svým dětem), což namodelujeme entitou *Adresa*.

Mezi entitami *Lékař* a *Pacient* je vztah typu M:N s atributem *Registrace*. Každý z lékařů bude registrovat pacienty. Pacient smí být registrován maximálně u jednoho lékaře. Tohoto lékaře může časem změnit (atribut *Registrace* bude obsahovat časový záznam). Proto vztah M:N.



Obrázek 7.2: ER diagram systému

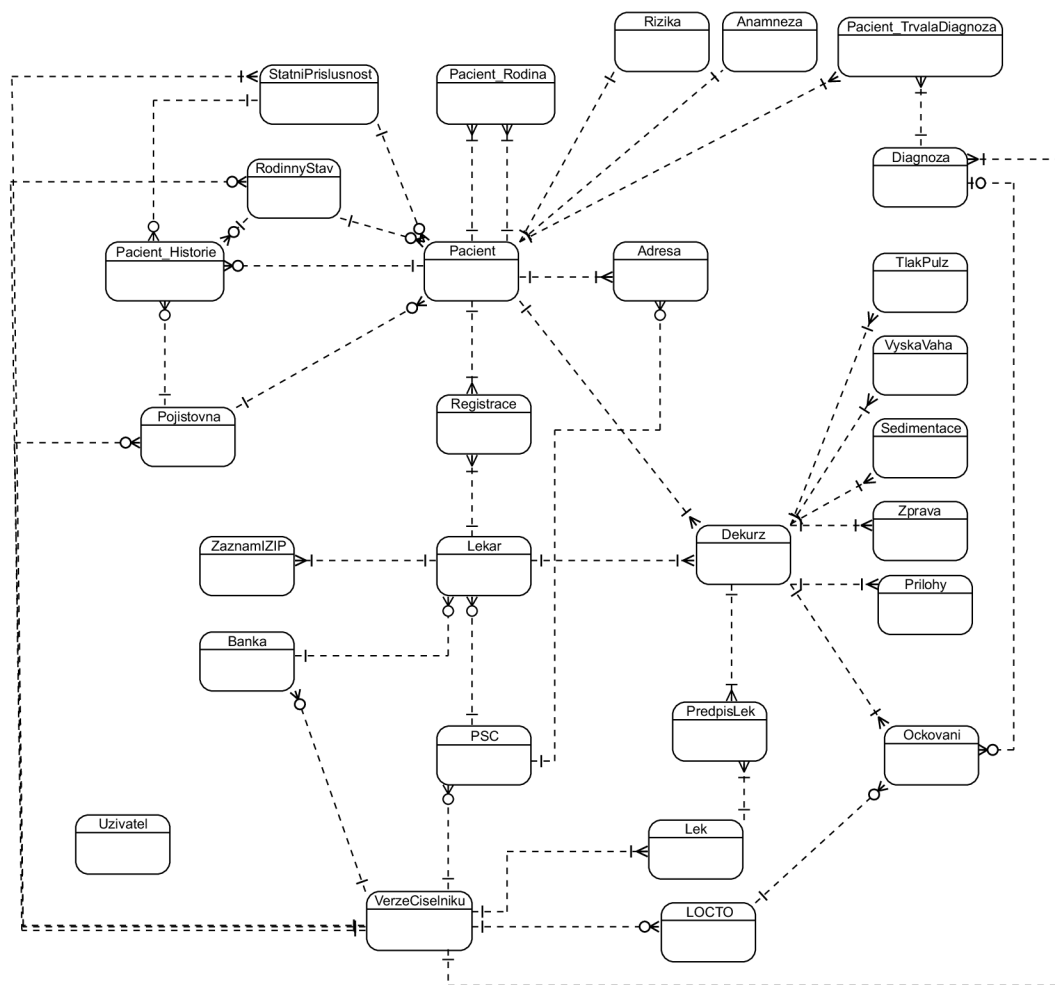
Hlavním zdravotním záznamem je dekurz (entita *Dekurz*). Dekurz se skládá z jednotlivých zápisů z návštěv pacienta u lékaře. Dekurz je víceméně strukturovaný text, skládá se z data provedení záznamu a zápisů jednotlivých provedených výkonů. Může obsahovat také závěr a doporučení pro další postup. Entita *Záznam* představuje provedený výkon. Specializuje se na entity *Očkování*, *Předpis na lék* a další (avšak pouze ty, které jsou zahrnuty do první iterace).

Entity *Anamnéza* a *Rizika* se přímo nevztahují k dekurzu (jsou v kartě zapsány někde na dobře viditelném místě a vyskytují se pouze v jednom exempláři). Ale protože jsou to také formy zdravotního zápisu, jsou modelovány jako specializace *Záznamu*. Tyto entity nejsou modelovány jako atributy pacienta proto, aby bylo možné zobrazit jejich jednotlivé verze, které byly odeslány do systému IZIP.

Každý záznam v dekurzu musí být podepsán lékařem, který jej provedl, proto je mezi entitami *Dekurz* a *Lékař* vztah *zapsal*.

7.2.2 Schéma databáze

Na obrázku 7.3 je zjednodušené schéma relační databáze. Plné schéma i s popisem je možné nalézt v příloze.



Obrázek 7.3: Zjednodušené schéma databáze

Tabulka *Data pro IZIP* slouží pro ukládání identifikátorů záznamů, které byly/ budou odeslány do IZIP. Ke každému záznamu je uchovávan také jeho typ (pro rychlé vyhledání), stav odeslání (byl odeslán/ bude odeslán/ typ nalezené chyby) a datum odeslání. Tabulka *Data pro IZIP* má vztah s tabulkou *Lékař*, díky kterému je určeno, který lékař data do IZIP odesílá (jedná se vlastně o duplicitní informaci, která byla zařazena pro rychlejší výběr záznamů z fronty jejich zapisovatelem).

7.3 Architektura systému

Jako architekturu systému jsem zvolil Model–View–Controller, která rozděljuje datový model aplikace, uživatelské rozhraní a řídicí logiku do tří nezávislých komponent tak, že modifikace některé z nich má minimální vliv na ostatní (převzato z [5]).

Tato architektura se skládá z:

Model – doménově specifická reprezentace informací, s nimiž aplikace pracuje,

Controller – reaguje na události (typicky pocházející od uživatele) a zajišťuje změny

v modelu nebo v pohledu,

View – převádí data reprezentovaná modelem do podoby vhodné k interaktivní prezentaci uživateli.

Vrstva View je tvořena třídami reprezentujícími dialogy zobrazované uživateli, vrstva Controller je tvořena jediným objektem (návrhový vzor Singleton), který přijímá uživatelské události a zabezpečuje jejich vyřízení Modelem, Model je tvořen pěti balíčky (Pacienti, Uživatelé, Číselníky, Záznamy, Databáze). Úkoly jednotlivých balíčků vyplývají z jejich názvu.

Balíček Databáze je opět tvořen jediným objektem (návrhový vzor Singleton), který se stará o komunikaci s databází.

V návrhu tříd vrstvy Model jsem použil návrhových vzorů. Návrhové vzory poskytují řešení pro často opakované problémy, např. problém procházení sekvenčně uspořádaných prvků – návrhový vzor Iterator.

Mezi často užívané vzory patří:

Proxy pattern – využívá se při potřebě zajištění kontroly nad přístupem k jinému objektu. V návrhu je konkrétně použit pro vytvoření objektu až v situaci, kdy je to opravdu potřeba: např. objekt třídy Pacient je složitý, protože obsahuje mnoho referencí na další objekty. Tyto objekty nejsou při vytváření objektu třídy Pacient vytvořeny, ale vytvoří se až v situaci, kdy jsou potřeba.

Singleton pattern – zajišťuje existenci pouze jediného objektu třídy v systému.

Iterator pattern – řeší problém procházení sekvenčně uspořádaných prvků.

Facade pattern – zjednodušuje vstupní bod do systému (balíčku, komponenty). Snižuje závislost tříd mezi balíčky, komponentami.

Factory pattern – rozhoduje za běhu programu o vytvoření instance konkrétní třídy.

Při studiu návrhových vzorů jsem vycházel z [10].

7.4 Bezpečnost

Navržený informační systém nedovoluje neoprávněný přístup k informacím. Uživatelé systému se musí autentizovat, neautentizovaní uživatelé nemají k datům v systému žádný přístup. Ihned po úspěšné autentizaci jsou uživatelům přidělena práva k vykonávání jednotlivých případů užití = autorizace. Tato práva lze modifikovat. Tím, že jsou uživatelům přidělena práva pro spouštění jednotlivých případů užití, nemůže docházet k neautorizované modifikaci informací a zároveň se k datům dostanou všichni ti, kteří k tomu mají práva.

První verze systému bude komunikovat pouze se systémem IZIP a to zabezpečeným protokolem HTTPS. Pro komunikaci bude užit komunikační modul IZIP, který je dodáván pouze v binární podobě. Proto nelze vyloučit přítomnost skrytých kanálů a ze stejného důvodu nelze tuto komunikaci označit za bezpečnou. (Probíhá sice zabezpečeným protokolem, ale nelze potvrdit užití uživatelova bezpečnostního certifikátu a vyloučit užití podvrženého útočnickova.)

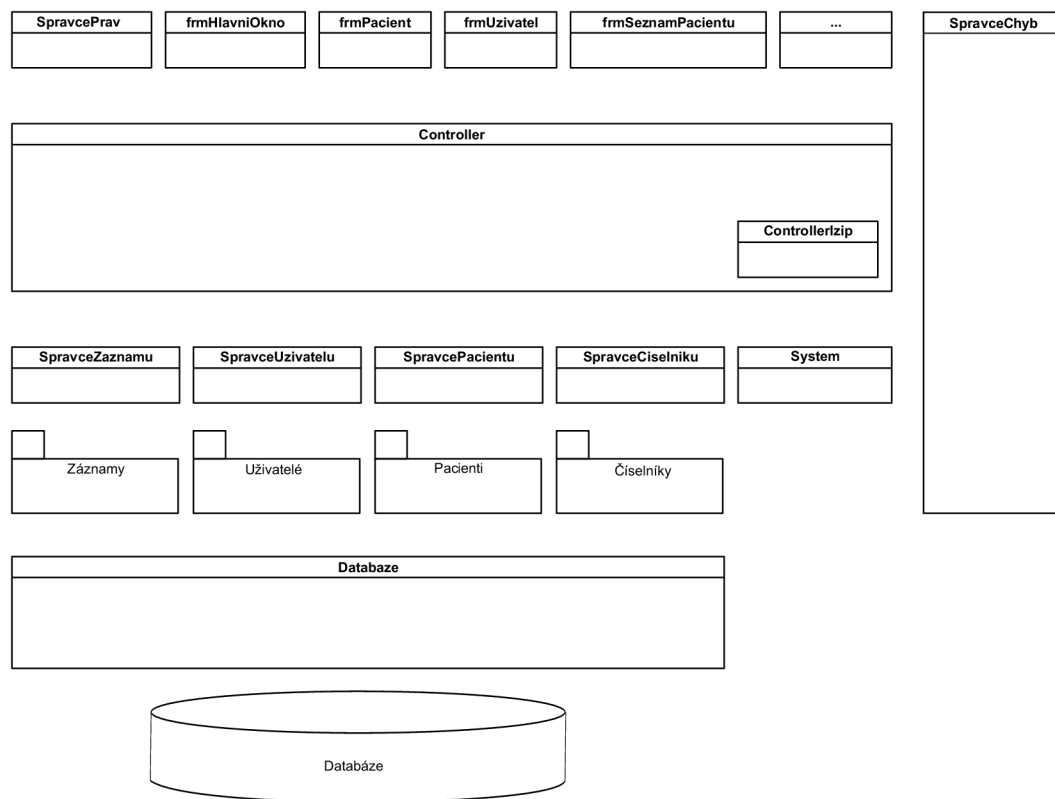
Data v databázi implementovaného systému nebudou šifrována (kromě uživatelských hesel, která budou zahashována funkcí SHA) a to z důvodu snížení propustnosti systému a problému s bezpečným úložištěm pro šifrovací klíč. K databázovému serveru budou smět

přístupovat pouze klientské programy, které budou znát přístupové údaje (autentizace v datové vrstvě). Komunikace s databází bude probíhat pomocí sdílené paměti (při užití embedded verze databáze) nebo pomocí síťového protokolu databázového serveru. Tento protokol nejspíš nebude šifrovaný a proto by mělo být zajištěno šifrování této komunikace pomocí jiného protokolu (IPSec, SSL/TLS a podobně). Zabezpečení fyzického přístupu k databázi pak bude záviset na zabezpečení hostitelského stroje a jeho správci.

Kapitola 8

Implementace

Architektura systému je tedy, jak již bylo zmíněno v kapitole 7.3, typu Model–View–Controller a obrázek 8.1 ji zobrazuje pohledem aplikačních tříd. V následujících kapitolách je pak probrána postupně vrstvu po vrstvě.



Obrázek 8.1: Architektura systému z pohledu aplikačních tříd

8.1 Databáze

Jak již bylo zmíněno v kapitole 7.2, využívá systém pro ukládání persistentních dat relační databázi. V této kapitole bude stručně popsána její struktura.

Základní tabulky jsou:

číselníkové tabulky – tabulky pro uložení číselníků. V každé z těchto tabulek může být uloženo více verzí stejného číselníku, a proto obsahují cizí klíč do tabulky *VerzeCiselniku*, který jednoznačně identifikuje verzi číselníku.

Pacient – do této tabulky jsou ukládány základní identifikační údaje pacienta. Není zde adresa, protože se v návrhu počítá s možností uložení více adres jednoho pacienta (tabulka *Pacient_Adresy*). Do databáze se také ukládají změny v těchto identifikačních údajích (tabulka *Pacient_Historie*), což je zajištěno databázovým triggerem, který je spouštěn při každé operaci UPDATE nad touto tabulkou.

Rizika – tato tabulka slouží k ukládání urgentních informací o pacientovi. Každý pacient může mít maximálně jeden platný řádek v této tabulce. Protože tato tabulka obsahuje záznamy, které jsou odesílány do systému IZIP, je pro tyto záznamy vyčleněna zvláštní tabulka (z důvodu zpětného zobrazení již odeslaných záznamů). Obdobně funguje i tabulka *Anamneza*.

Lekar – tato tabulka slouží k ukládání informací o lékaři. Obsahuje také přístupové údaje k systému IZIP.

Uzivatel – do této tabulky jsou ukládány všechny ostatní uživatelské účty. K rozlišení typu uživatelského účtu je vyhrazen jeden sloupec.

Dekurz – tabulka pro ukládání denních záznamů o návštěvě pacienta. Na dekurz mohou být navázány další záznamy jako očkování, zpráva, předpis na lék a další.

tabulky pro připojené záznamy – primární klíče těchto tabulek jsou tvořeny hodnotami, které jsou v rámci všech těchto tabulek unikátní a obsahují cizí klíč do tabulky *Dekurz*, ke kterému se vážou.

8.1.1 Datové pohledy

Databáze obsahuje dva datové pohledy. První nad tabulkou *Pacient*, který slouží pro vyhledávání pacientů podle jména a příjmení s ignorováním velikostí písmen a interpunkčních znamének, a druhý nad tabulkami *Lekar* a *Uzivatel* pro vyhledávání uživatele podle uživatelského jména a pro kontrolu unikátnosti uživatelských jmen.

8.1.2 Databázové procedury

V databázi jsou přítomny tyto triggerly:

- triggerly zajišťující přidělování jedinečných hodnot primárním klíčům,
- trigger zajišťující naplňování tabulky *Pacient_Historie* (je spouštěn po operaci UPDATE nad tabulkou *Pacient*),
- triggerly zajišťující naplňování tabulky *ZaznamIZIP*. Tyto triggerly jsou spouštěny po operacích INSERT nad tabulkami se záznamy a jejich úkolem je zkontrolovat, zda pacient, jehož se záznamy týkají, má aktivní Zdravotní knížku IZIP a zda lékař, který zápis záznamu provedl, je registrován v systému IZIP. Pokud jsou obě tyto podmínky splněny, pak je záznam přidán do fronty záznamů pro odeslání do systému IZIP.

V databázi je také procedura, která zajišťuje symetričnost vztahů v tabulce *Pribuzny*. Je-li pacient uveden jako příbuzný jiného pacienta, pak se tento příbuzenský vztah projeví i u tohoto příbuzného.

8.2 Datová vrstva

Datová vrstva ambulantního systému je tvořena relační databází Firebird (více informací v kapitole 8.7.1). Při implementaci byla využita *embedded* verze, což znamená, že systém komunikuje s databází přes sdílenou paměť (pomocí volání funkcí dynamicky linkované knihovny).

Přechod na jinou verzi databáze Firebird není nijak složitý, stačí zaměnit *embedded* knihovnu za síťovou a systém bude komunikovat s databází přes síťové rozhraní. Databáze tak může být sdílena více klienty.

Pro eliminaci přímé vazby systému na konkrétní databázi byla do systému přidána abstraktní databázová vrstva, která odstiňuje databázové operace závislé na užití databázi od vyšších vrstev. Systém tak může být přenesen na jinou databázi, přičemž stačí přepsat pouze tuto abstraktní databázovou vrstvu.

Abstraktní databázová vrstva je implementována ve třídě *Databaze*, jejímiž hlavními funkcemi jsou:

- nastavení parametrů spojení,
- připojení k databázi,
- odpojení od databáze,
- spuštění databázové transakce,
- COMMIT databázové transakce,
- ROLLBACK databázové transakce,
- provedení databázového příkazu s prázdnou množinou výsledku,
- provedení databázového příkazu s neprázdnou množinou výsledku,
- převzetí výsledku databázové operace.

Pro potřeby práce s výsledky databázových dotazů byly vytvořeny třídy pro zapouzdření základních databázových typů:

- Smallint, Integer, Bigint – pro zapouzdření celočíselných databázových typů,
- Char, Varchar – pro zapouzdření znakových databázových typů,
- Date, Time, Timestamp – pro zapouzdření časových databázových typů,
- Decimal, Numeric – pro zapouzdření číselných typů s pevnou desetinnou čárkou,
- Float, Double – pro zapouzdření číselných typů s pohyblivou desetinnou čárkou,
- Blob – pro rozsáhlé textové nebo binární datové typy.

8.3 Vrstva Model

Vrstva Model je tvořena objekty, které jsou ukládány nebo načítány z databáze, a třídami, které tyto objekty spravují – starají se o jejich načítání, ukládání a úklid z paměti. Tyto třídy jsou odvozeny od třídy *Spravce*.

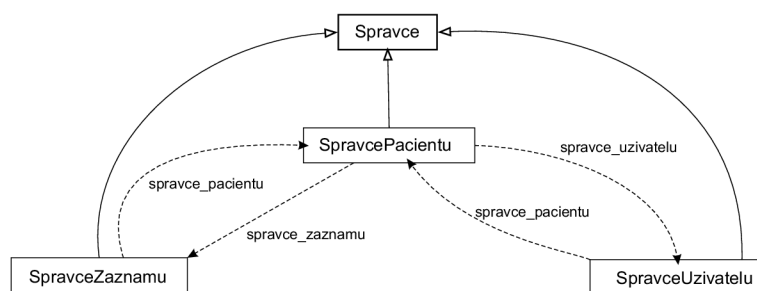
8.3.1 Třída *Spravce*

Tato třída zapouzdřuje operace nad databázovou transakcí, v rámci které databázové operace odvozených tříd probíhají. Obsahuje tedy jedinečný identifikátor databázové transakce a metody pro start transakce, COMMIT a ROLLBACK a je bázovou třídou pro třídy *SpravceUzivatelu*, *SpravcePacientu* a *SpravceZaznamu*.

8.3.2 Třídy *SpravceUzivatelu*, *SpravcePacientu* a *SpravceZaznamu*

Objekty těchto tříd mají na starosti správu objektů, které mohou být během své životnosti načítány, měněny a zpětně ukládány do databáze. K tomu potřebují pracovat v rámci databázové transakce s přístupem Read-Write, což jim zajistí bázová třída *Spravce*.

Protože jsou těmito třídami spravované objekty provázané (např. *Pacient* (spravuje *SpravcePacientu*) – *Dekurz* (spravuje *SpravceZaznamu*) – *Lékař* (spravuje *SpravceUzivatelu*)), jsou také tyto třídy provázané, což ukazuje obrázek 8.2. Z každého objektu těchto tříd lze tedy získat ukazatel na objekt zbývajících tříd z této trojice a zároveň tyto tři objekty sdílejí jedinou databázovou transakci, což je z důvodu vlastnosti transakčního zpracování *izolovanost*.



Obrázek 8.2: Asociační a dědičné vztahy mezi správci objektů

Správa objektů

Stejně jako je tomu u třídy *SpravceCiselniku* má i zde každý spravovaný objekt jedinečný identifikátor v rámci své třídy, pomocí kterého může být poptáván. Výjimkou jsou objekty tříd odvozených od bázové třídy *Zaznam*, které mají identifikátor unikátní v rámci této bázové třídy.

Tito správci objektů mají stejně jako *SpravceCiselniku* asociativní pole pro každou třídu, kam jsou tyto objekty ukládány. Na rozdíl od *SpravceCiselniku* je za běhu programu vytvořeno více instancí těchto tříd – objekty vznikají se zahájením transakce a s jejím ukončením (COMMIT, ROLLBACK) zanikají a s nimi i všechny jimi spravované objekty (i nezměněné), což se děje z toho důvodu, že tyto objekty přestávají být platné (jiná transakce je mohla změnit).

Vytváření, načítání a ukládání objektů

Vytváření objektů spravovaných těmito třídami se děje pomocí návrhového vzoru Factory Method – správce objektů je požádán o vytvoření konkrétního objektu třídy, jejíž objekty spravuje, a to zavoláním příslušné metody s minimem požadovaných parametrů. Správce vytvoří konkrétní objekt, přidělí mu jedinečný identifikátor, který je pro odlišení od objektů uložených v databázi záporný, nastaví mu požadované parametry, uloží jej to asociativního pole a nastaví případné vazby, má-li objekt nějaké. Např. vytvoření nového objektu *Dekurz* znamená zavolat metodu *vytvorDekurz* třídy *SpravceZaznamu* s parametry identifikátor pacienta a lékaře. Tato třída vytvoří nový objekt požadované třídy a také vytvoří objekty tříd *ProxyPacient* a *ProxyLekar*, kterým nastaví předané identifikátory.

Načítání objektů se na rozdíl od třídy *SpravceCiselniku* neděje po skupinách, ale po jednom objektu. Objekty závislé na načítaném se načítají až když je to potřeba, při načítání jsou nahrazeny proxy objektem.

Ukládání objektů do databáze se děje hromadně. Každý spravovaný objekt si hlídá, zda se některý jeho atribut změnil, a pokud k tomu nedošlo, do databáze se neodesílá.

8.3.3 Třída *SpravceCiselniku*

Tato třída má na starosti načítání objektů z číselníků. Tyto objekty nemohou být modifikovány, a proto využívají transakci s přístupem pouze pro čtení. Ze stejného důvodu není potřeba mít v systému více instancí této třídy, a proto byl na tuto třídu aplikován návrhový vzor Singleton. Objekt této třídy je vytvořen při spuštění programu a je uvolněn až při jeho ukončení. Protože objekt této třídy využívá během své životnosti jedinou databázovou transakci, není tato třída odvozena od třídy *Spravce*.

Správa objektů

Každý objekt číselníku je rozlišitelný pomocí identifikátoru, který je unikátní v rámci třídy objektů.

Protože se objekty číselníku během chodu programu nemění, jsou po načtení z databáze uchovávány v paměti pro případné pozdější znovupoužití. Správce tedy obsahuje pro každou třídu objektů asociativní pole, kam jsou načítané objekty ukládány. Klíčem do tohoto pole je identifikátor objektu a hodnotu tvoří ukazatel na objekt v paměti.

Načítání objektů

Je-li potřeba získat objekt, který ještě není v paměti, je nutné jej načíst z databáze. Je poměrně dost pravděpodobné, že v budoucnu bude potřeba z databáze načíst také objekty, které s daným objektem sousedí (podle identifikátoru), proto se z databáze číselníků nenačítají objekty po jednom, ale vždy po skupinách. Velikosti skupin byly experimentálně určeny tak, aby načítání netrvalo příliš dlouho a uživatel nemusel dlouho čekat.

Kromě tohoto způsobu načítání obsahuje tato třída také metody pro vrácení větších částí aktuálních číselníků. V tomto případě jsou z databáze nejprve načteny identifikátory všech objektů tvořících daný číselník a posléze jsou načítány skupiny těchto objektů (bez ohledu na jejich souslednost v databázi).

8.3.4 Třída *System*

Třída *System* zapouzdřuje systémové a systémově závislé operace jako je například detekce systémového kódování znaků, desetinného oddělovače, znaku národní měny, aktuálního data a času a otevírání dokumentů v asociovaných prohlížečích (tato metoda je zatím dostupná pouze pro operační systémy Microsoft Windows).

Poslední metodou nabízenou touto třídou je metoda pro otevření zdravotní knížky IZIP. Vývojáři systému IZIP navrhují pro tento účel použití komponenty programovacího jazyka Delfi. Tento způsob není však přenositelný a proto jsem se rozhodl pro způsob jiný, a to použití metody třídy *System* pro otevření dokumentu. Tímto dokumentem je HTML stránka s předvyplněným formulářem, který se po načtení v internetovém prohlížeči pomocí Javascriptu odešle na bránu IZIVIEW, a poté je z disku smazán (název tohoto souboru je náhodně generován). Tento způsob je systémově přenositelný, ale jeho nevýhodou je nebezpečí, že dokument se nepodaří z disku smazat a údaje v něm mohou být prozrazeny a také že v systému není pro otevírání HTML dokumentů zaregistrován internetový prohlížeč, ale některý editor. V případě prozrazení údajů v dokumentu získá útočník přístup pro odesílání dat do systému IZIP. Pro nahlížení do Zdravotních knížek pacientů potřebuje získat také osobní heslo zdravotnického pracovníka.

8.4 Vrstva Controller

Hlavním úkolem vrstvy Controller je předávat uživatelem generované události nižším vrstvám a nastalé změny v modelu předat vyšším vrstvám, jinými slovy odlišuje implementaci nižší vrstvy Model od vrstvy View.

Pro zajištění nezávislosti na počtu a typu parametrů volaných metod jsem zvolil komunikaci mezi vrstvou View a Controller ve formě XML. Pro tento účel jsem použil parsovací XML knihovnu Xerces-C++.

Vrstva Controller je tvořena jediným objektem stejnojmenné třídy. Tento objekt zabezpečuje veškerou komunikaci objektů vrstvy View s objekty vrstvy Model a také s komunikační knihovnou IZIP.

8.4.1 Komunikace se systémem IZIP

Komunikace se systémem IZIP probíhá pomocí zabezpečeného protokolu HTTPS. V hlavičce protokolu se zasílají přístupové údaje (identifikační číslo uživatele, MD5-hash přístupového hesla a informace o klientském software). Obsah těla protokolu je tvořen daty ve formátu XML (záznamy, doztazy na existenci pacientů v systému) a případnými binárními daty (přílohy).

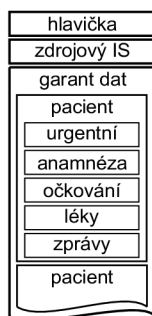
Odesílání záznamů

Pro odesílání záznamů do systému IZIP je vývojáři tohoto systému stanoven doporučený postup, který se skládá z následujících kroků:

1. Zkontrolovat na bráně IZICHECK, zda všichni klienti, jejichž záznamy se budou odesílat, mají v systému IZIP aktivovanou Zdravotní knížku. Pokud ne, pak je z odesílání vyřadit a otestovat je později (zaregistrování pacienta trvá přibližně dva až tři týdny). Pokud však ani po čtyřech týdnech není pacient aktivní, pak tyto záznamy z fronty vyřadit úplně.

2. Odeslat data.
3. Zpracovat odpověď a podle navrácené chyby data zařadit znovu do fronty na pozdější odeslání, označit za již odeslané nebo označit za chybné a do napravení chyby neodesílat.

Struktura zdravotních záznamů ve formátu XML podléhá Datovému standardu Ministerstva zdravotnictví ČR verze 03.01.01 (viz [18]) a znázorňuje ji obrázek 8.3 (obrázek zohledňuje data zasílané implementovaným systémem).



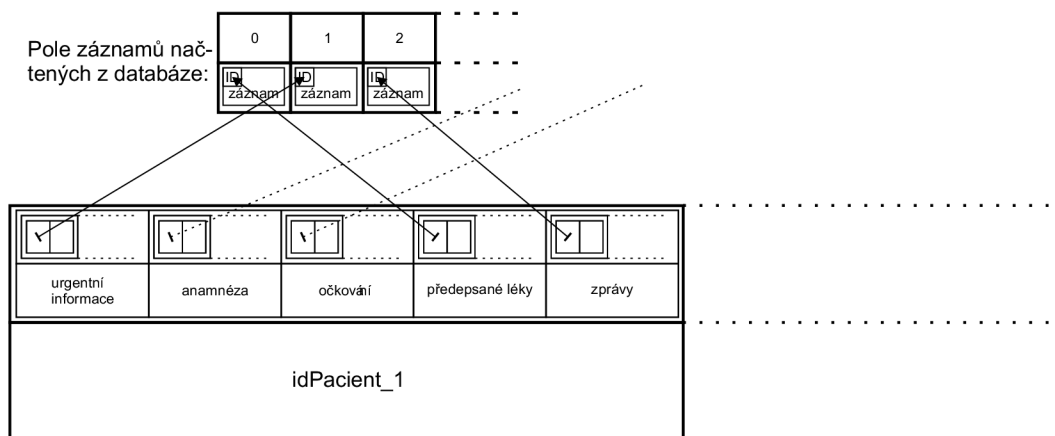
Obrázek 8.3: Struktura XML zasílaného do IZIP

Takovýto XML dokument se skládá z:

- hlavičky obsahující jednoznačnou identifikaci souboru, verzi použitých číselníků, formu odpovědi přijímacího systému,
- informace o klientském systému,
- informace o odesílateli (IČO, odbornost, ...),
- informace o pacientovi, jehož záznamy následují
- a záznamů.

Protože jsou záznamy určené pro odeslání do systému IZIP v databázi seřazeny podle času svého vzniku, musí být před vlastním sestavením XML dokumentu přetransformovány. Podstatu transformace zachycuje obrázek 8.4.

Záznamy jsou postupně brány z fronty. Nejprve se u každého záznamu určí, kterému pacientu záznam patří, poté o jaký typ záznamu se jedná a následně je přidán do seznamu záznamů tohoto typu. Jedná-li se o anamnézu nebo urgentní informace, zkontroluje se, zda již není jiný záznam téhož typu připraven k odeslání (tyto záznamy mohou být odeslány pouze v jednom exempláři). Pokud tento případ nastal, připraví se k odeslání pouze ten nejnovější (ostatní se označí jako odeslané). Po zpracování celé fronty záznamů se sestaví seznam pacientů, jejichž záznamy se budou odesílat a tento seznam je odeslán na bránu IZICHECK pro kontrolu aktivity Zdravotních knížek. Je-li u některého z pacientů zjištěna neaktivovaná Zdravotní knížka, jsou záznamy tohoto pacienta z odesílání vyřazeny. Nyní může být XML dokument sestaven a odeslán.



Obrázek 8.4: Transformace záznamů IZIP

8.4.2 Odpověď systému IZIP

Po úspěšném odeslání záznamů vrací systém IZIP XML dokument s výsledkem přijetí zasílaných dat (také podléhá Datovému standardu MZ ČR). Pokud byla při odesílání záznamů požadována podrobná odpověď, vrací IZIP seznam s návratovými kódy jednotlivých záznamů a popisy nalezených chyb (kódy rozpoznávaných chyb lze nalézt v tabulce 8.1).

Ke každé nalezené chybě je uveden nejen její kód, ale i její lokalizace a textový popis. Způsob lokalizace chyby už ale datový standard nedefinuje. Systém IZIP používá pro lokalizaci rodné číslo pacienta, v jehož záznamu byla chyba nalezena, následované názvem XML elementu, v němž se chyba nachází. Pokud je u pacienta těchto elementů více, pak je (pokud je to možné) připojen další textový diskriminant.

Textové popisy chyb jsou pro programové zpracování velmi obtížné (k jednomu typu chyby existuje více popisů) a mají nejspíš sloužit pro analýzu chybových stavů programátory.

Výše popsáný způsob lokalizace chyby není nikde zdokumentován, tento popis vznikl pouze na základě svých vlastních pozorování. Zpracování chyb detekovaných systémem IZIP a nahlášených pomocí Datového standardu MZ ČR je tedy velmi obtížné.

Po zpracování chyby jsou všechny záznamy, které její lokalizaci vyhoví, označeny za neodeslané a jejich odeslání se bude opakovat (ať už je hlášena chyba jakákoliv). Sofistikovanější zpracování chyb, stejně jako druhý způsob odesílání záznamů (jeden záznam v jednom XML dokumentu) budou implementovány v některé z nadcházejících verzí.

8.5 Vrstva View

Vrstvu View tvoří třídy grafického uživatelského rozhraní (jedna třída odpovídá jednomu dialogovému oknu), které, jak už bylo zmíněno výše, komunikují s vrstvou Controller pomocí XML. Pro grafické uživatelské rozhraní byla v projektu použita knihovna Qt, ale díky nezávislosti jednotlivých vrstev, může být kdykoliv vyměněna za jinou.

Hlavní dialogové okno aplikace tvoří třída *frmHlavniOkno*. Toto okno využívá koncepcí MDI (Multiple document interface), kdy jsou dceřiná okna na toto okno vázána a přes toto hlavní okno ovládána (sdílejí Menu, Toolbar, ...). Dceřinými okny mohou být okna reprezentující zdravotní kartu pacienta nebo editor pro zápis lékařské zprávy. Ostatní okna

| Kód | Popis |
|---|--|
| chyby nerozlišené | |
| 000 | chyba nerozlišená, popis v textu |
| chyby struktury a obsahu XML dokumentu | |
| X01 | dokument není "well formed XML" – nepárové značky, nepovolené znaky, aj. |
| X02 | chybí povinná položka |
| X03 | neznámý atribut |
| X04 | neznámý element |
| X05 | nedovolené opakované použití |
| X99 | jiná/nerozpoznaná |
| chyby obsahu a vazeb údajů DS | |
| D01 | nevhodná délka (příliš dlouhé) |
| D02 | neodpovídá povolenému výčtu hodnot |
| D03 | hodnota mimo přípustný rozsah |
| D04 | neodpovídá číselníku internímu nebo externímu = chybný kód z číselníku |
| D99 | jiná/nerozpoznaná |
| chyby obsahu (a vazeb) na úrovni aplikace | |
| A01 | chybný/neznámý identifikátor |
| A99 | ostatní |

Tabulka 8.1: Seznam chyb dle Datového standardu MZ ČR

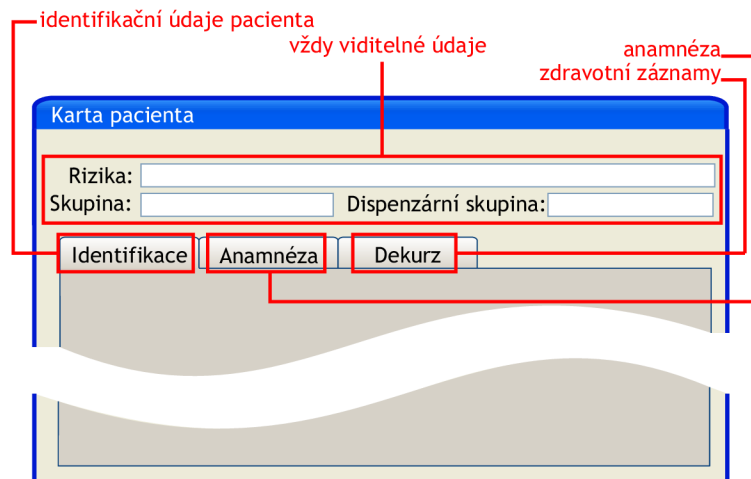
mají v porovnání s těmito kratší životnost (např. končí ihned po vybrání položky) a proto jsou zobrazovány jako dialogy s aplikační modalitou (zobrazí se nad všemi okny aplikace a s aplikací nelze pracovat, dokud nejsou zavřena).

Okno reprezentující zdravotní kartu pacienta je rozděleno do čtyř částí:

- První tvoří vždy viditelné údaje, což jsou rizikové faktory pacienta, dispenzární skupina a skupina pacientů, do níž je zařazen,
- Druhou část tvoří identifikační údaje pacienta (jméno, adresa, pojišťovna, zaměstnání, ...),
- Třetí část tvoří anamnéza pacienta. Ta se skládá jak z polí pro její zápis, tak z tabulky trvalých diagnóz pacienta a seznamem příbuzných pacienta, kteří jsou rovněž v kartotéce.
- Poslední část tvoří zdravotní záznamy pacienta – dekurzy. Ty jsou v přehledu seřazeny podle data zápisu a jsou zobrazeny i se součástmi, ze kterých se skládají (měření tlaku, očkování, ...).

Pro lepší představivost je zařazen obrázek 8.5, který popisovaný dialog znázorňuje.

Kromě tříd pro grafické rozhraní je zde také třída *SpravcePrav*, která má za úkol spravovat práva pro spouštění jednotlivých uživatelských akcí. Práva pro akce budou v některé z příštích verzí načítána z databáze, nyní jsou explicitně definována přímo ve třídě.



Obrázek 8.5: Dialog se zdravotní kartou pacienta

8.6 Třída *SpravceChyb*

Objekt třídy *SpravceChyb* slouží pro zachytávání chyb nastalých při běhu programu. Chyby nijak nezpracovává, pouze zaznamená čas jejich vzniku, typ a případně popis. Je přístupný všem objektům ze všech vrstev architektury. Za běhu programu existuje pouze jediná instance této třídy (návrhový vzor Singleton), která před svým ukončením vypisuje všechny sesbírané chyby.

Stručný popis funkčnosti této třídy:

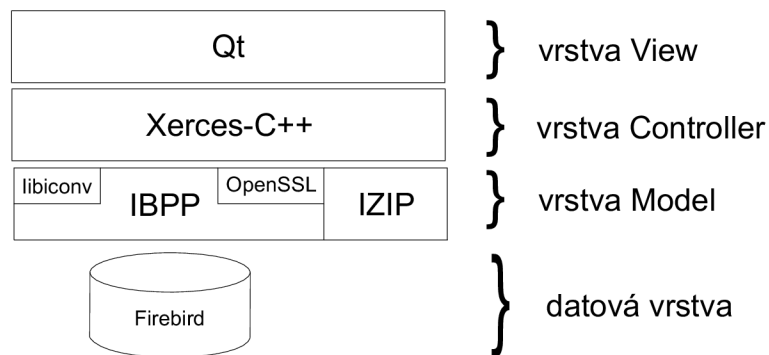
- Při načítání adresy pacienta (v objektu třídy *SpravcePacientu*) dojde k chybě.
- *SpravcePacientu* tuto skutečnost oznámí správci chyb (nastaví její typ a popis), přičemž ten mu vrátí identifikátor, pod kterým je tato chyba evidována.
- *SpravcePacientu* tento identifikátor předá vyšší vrstvě a ta může informovat uživatele a zobrazit mu i podrobný popis chyby. Výhodou tohoto postupu je předávání pouze identifikátoru chyby a možnost přístupu k jejímu popisu.

8.7 Architektura systému z pohledu komponent

Tato kapitola má za úkol představit jednotlivé komponenty, které byly v systému použity, a návaznost těchto komponent na jednotlivé vrstvy architektury. Celkový pohled na architekturu z pohledu těchto užitých komponent dává obrázek 8.6.

Při implementaci informačního systému byly užity tyto komponenty:

- Firebird – relační databáze – datová vrstva
- IBPP – rozhraní k relační databázi – vrstva Model
- OpenSSL – šifrovací knihovna – vrstva Model
- libiconv – převod mezi kódováními řetězců – vrstva Model
- Xerces-C++ – XML parser – vrstva Controller



Obrázek 8.6: Architektura systému z pohledu komponent

- knihovna IZIP – komunikace s IZIP – vrstva Model
- Qt – grafické rozhraní – vrstva View

8.7.1 Firebird

Tato kapitola popisuje databázový server Firebird, jeho vlastnosti a důvody je užití. Při jejím psaní jsem čerpal z [13] a [8].

Firebird je multiplatformní relační databáze podporující standard ANSI SQL. Vznikla v roce 2000, kdy se rozhodla firma Borland uvolnit zdrojové texty beta-verze InterBase 6.0 pod Open Source licenci. Vývoje produktu se chopila komunita Firebird Foundation, která nedávno vydala verzi 2.1 tohoto produktu.

Firebird nabízí podporu pro databázové transakce, procedury, trigger, generátory, datové pohledy, znakové sady a konvence pro řazení, uživatelské funkce a události, a je pro něj k dispozici mnoho ovladačů a nástrojů pro přístup, jako jsou nástroje pro přístup z jazyků C++, Java, Delphi, .Net, PHP nebo ovladače OLEDB a ODBC.

Firebird je k dispozici ve třech variantách:

SuperServer – Síťová varianta, pro obsluhu uživatelských spojení se užívá vláken, která sdílí cache serveru.

Classic – Síťová varianta, pro obsluhu uživatelských spojení se užívá procesů, které se synchronizují pomocí databázových souborů.

Embedded – Lokální varianta, plná funkcionality serveru zabudovaná do několika dynamicky linkovaných knihoven, které se přilinkují ke klientské aplikaci.

Datové typy

Firebird podporuje tyto datové typy:

- SMALLINT, INTEGER, BIGINT – celočíselné datové typy.
- NUMERIC, DECIMAL – datové typy pro práci s čísly s pevnou desetinnou čárkou.
- FLOAT, DOUBLE PRECISION – číselné typy s pohyblivou desetinnou čárkou.
- DATE, TIME, TIMESTAMP – typy pro uložení data a času

- CHAR, VARCHAR, NCHAR – typy pro uložení znakových řetězců.
- BLOB – typ pro ukládání binárních dat nebo rozsáhlých textů.

Firebird neumožňuje vytvářet nové uživatelské typy, ale umožňuje vytvářet *domény*, což jsou pojmenované skupiny vlastností spojené s existujícím datovým typem.

Znakové sady a konvence pro řazení

Firebird umožňuje práci s různými způsoby kódování znaků a způsoby pro řazení. Znakovou sadu lze definovat globálně pro databázi a alternativně pro znakový sloupec tabulky.

Pro každou znakovou sadu je definován alespoň jeden způsob řazení. Řazení lze definovat při definici sloupce tabulky nebo ve výrazech ORDER BY či GROUP BY. K dispozici je i české řazení, které je definováno pro znakové sady ISO-8859-2 a CP-1250.

Databázové transakce

Firebird se liší od většiny databází v implementaci transakcí. Narozdíl od nich nepoužívá systém transakčního protokolu a zámků, ale multigenerační architekturu.

Multigenerační architektura – Každý řádek v databázi je reprezentován seznamem verzí tohoto řádku. Aktuální verze je na začátku tohoto seznamu. Transakce, která řádek zmodifikuje, vytvoří novou položku v tomto seznamu a po potvrzení transakce se tento řádek označí jako potvrzený. Jinak je brán jako nepotvrzený a je při čtení ignorován.

Výhody – rychlá operace ROLLBACK, nevyžaduje použití zámků, vyšší propustnost

Nevýhody – vyšší nároky na místo na disku

Procedury a spouště

Firebird podporuje vytváření programových bloků uložených a prováděných na serveru. Bloky jsou vytvářeny v jazyce Procedural SQL. Programové bloky mohou být v podobě procedury nebo spouště.

Spouště jsou automaticky prováděné programové bloky nad tabulkou nebo datovým pohledem spojené s operacemi INSERT, UPDATE nebo DELETE.

Procedury jsou proveditelné bloky, které se dělí na běžné a dotazovatelné.

Běžná procedura je procedura, která vrací maximálně jeden řádek hodnot.

Dotazovatelná procedura je procedura, která může vrátit více než jeden řádek hodnot.

Uživatelské funkce

Firebird má k dispozici pouze malou skupinu vestavěných funkcí, což přispívá k jeho menší velikosti a vyšší rychlosti. Tato vlastnost je kompenzována možností přidání vlastních funkcí – uživatelsky definovaných.

Uživatelsky definované funkce jsou ukládány v externích dynamicky linkovatelných knihovnách, což znamená, že mohou být napsány v libovolném programovacím jazyce, který je schopen takovéto knihovny vytvářet.

Události

Firebird umožňuje rozesílat připojeným klientům zprávy o událostech v databázi. Zprávy jsou rozesílány z uložených procedur a spouští a to v okamžiku potvrzení transakce, v rámci které byla daná procedura či spoušť spuštěna. Událost je reprezentována řetězcem až 64 znaků.

Příkladem může být zpráva o přidání řádku do určité tabulky.

Generátory

Firebird neumožňuje definovat speciální typ číselného sloupce tabulky, kterému je při vložení nového řádku tabulky automaticky přiřazena jedinečná hodnota z vzestupné řady čísel (typ sloupce `autoincrement` nebo `identita`), jak je tomu u některých jiných databází, ale nabízí obecný mechanismus pro generování sekvencí jedinečných čísel pomocí tzv. *generátorů*.

Hodnotu generátoru lze zvyšovat o definovanou hodnotu a lze ji přenastavit na definovanou hodnotu.

Generátory nemají žádnou přímou vazbu na tabulky a lze je užít libovolně. Změna hodnoty generátoru je prováděna zcela mimo transakční kontrolu, zrušení transakce nevrací generátoru jeho původní hodnotu, a proto generátory nelze použít pro generování souvislé posloupnosti čísel. Výhodou je rychlost generování.

Datové pohledy

V souladu se standardem SQL-92 nabízí Firebird datové pohledy. Datový pohled je definován jako virtuální tabulka, která odvozuje svůj obsah od obsahu jiných tabulek.

Důvody užití

Databáze Firebird byla zvolena pro výčet nabízených vlastností, Open Source licenci, přenositelnost a existenci v různých verzích. Pro pracoviště jednoho lékaře nejlépe vyhoví verze `Embedded`, pro pracoviště více lékařů, kteří chtějí sdílet data pacientů, pak verze `Classic` nebo `Super Server`.

8.7.2 IBPP

IBPP je rozhraní k databázi Firebird pro aplikace psané v jazyce C++. Toto rozhraní zapouzdřuje API databáze Firebird, které je v jazyce C, do tříd jakyka C++. IBPP je šířeno pod Open Source licenci, funguje se všemi dostupnými verzemi databáze Firebird a je tvořeno standardním C++ kódem. Proto je velmi dobře přenositelné.

IBPP tvoří tato rozhraní:

Database – rozhraní reprezentující jedno spojení s databází a umožňující operace nad ním (navázání, ukončení, vrácení informací o spojení, vytvoření databáze, ...).

Transaction – rozhraní pro práci s transakcí (vytvoření, rollback, commit, ...).

Statement – rozhraní pro provádění databázových příkazů a zpracování jejich výsledků.

Blob – rozhraní pro práci s datovým typem Blob (načtení, uložení, ...).

Array – rozhraní pro práci s poli hodnot (načtení, uložení, ...).

Service – rozhraní pro práci se systémovými funkcemi databáze (práce s uživateli, nastavování velikostí bufferů, ...).

Events – rozhraní pro práci s událostmi (vytvoření, zrušení, naslouchání, ...).

8.7.3 OpenSSL

OpenSSL je projekt, který si klade za cíl vytvořit toolkit pro protokoly SSL a TLS. Jeho součástí je nástroj pro příkazový řádek, SSL/TLS knihovna a kryptografická knihovna. Projekt je šířen pod Open Source licencí a je přenositelný na mnoho operačních systémů.

Pro potřeby projektu jsem z OpenSSL použil pouze kryptografickou knihovnu, jmenovitě hashovací funkce SHA1 a MD5.

8.7.4 Libiconv

Libiconv je knihovna pro převod textových řetězců mezi různými znakovými sadami. V projektu je využita pro převod řetězců mezi kódováním hostitelského operačního systému a kódováním znaků v databázi (tato kódování se mohou lišit).

Libiconv je šířena pod Open Source licencí a je přenositelná na mnoho operačních systémů.

8.7.5 Xerces-C++

Xerces-C++ je validační XML parser napsaný v přenositelné podmnožině jazyka C++, pomocí kterého je možné XML dokumenty načítat, generovat, manipulovat s nimi a validovat a to pomocí DOM, SAX a SAX2 API (při implementaci bylo užito pouze DOM API).

Xerces-C++ je šířen pod Open Source licencí a je přenositelný na mnoho operačních systémů.

8.7.6 Knihovna IZIP

Dodavatelům zdravotnických informačních systému je k dispozici (na požádání) komunikační modul pro komunikaci se systémem IZIP. Tento modul je distribuován pouze v binární podobě ve formě dynamicky linkované knihovny a spustitelného programu a je určen pouze pro operační systém Microsoft Windows a GNU/Linux.

Popis knihovny IZIP

Knihovna obsahuje tyto funkce:

- funkce pro zjištění verze knihovny
- funkce pro přípravu dat k odeslání
- funkce pro připojení souboru s přílohou, který je načten v paměti
- funkce pro připojení souboru s přílohou, který je na disku
- funkce pro zobrazování postupu odesílání/příjmu dat
- funkce pro odeslání dat do IZIPu
- funkce pro získání podrobného výpisu z odeslání dat

- funkce pro získání popisu nalezené chyby
- funkce pro závěrečný úklid

8.7.7 Qt

Qt je multiplatformní aplikační framework pro vytváření aplikací s grafickým uživatelským rozhraním pomocí programovacích jazyků C++, Python, Ruby, Perl, Java a dalších, a který obsahuje mimo jiné i knihovnu pro práci s XML, knihovnu pro práci s databází, knihovnu pro práci v počítačové síti a knihovnu pro práci s OpenGL a multimédií.

Qt je šířeno také pod Open Source licencí a je přenositelné na mnoho operačních systémů.

V projektu byly z knihovny Qt mimo funkce pro tvorbu dialogů také použity funkce pro práci s XML (komunikace s vrstvou Model) a funkce pro konverzi mezi různými znakovými sadami.

8.8 Demonstrace projektu

Tato kapitola má za úkol stručně ukázat, jak se pomocí programu dá pracovat se systémem IZIP.

Po spuštění programu je zobrazen dialog pro přihlášení uživatele. Ve verzi programu, která je přiložena na datovém disku, jsou v systému evidováni tři uživatelé:

- xlogin00 – uživatel *lékař* (bez hesla), který má zároveň účet v systému IZIP,
- xlogin01 – uživatel *lékař* (heslo "heslo"), bez účtu v systému IZIP,
- xsestra00 – uživatel *sestra* (bez hesla).

Po úspěšném přihlášení je zobrazeno hlavní okno programu. V menu Pacient tohoto okna vybráním položky Kartotéka otevřeme seznam pacientů v systému. Vybereme pacienta s příjmením Testovací a potvrzením dialogu otevřeme jeho zdravotní kartu.

V kartě pacienta je otevřena záložka se zdravotními záznamy a je vytvořen záznam nový pro aktuální datum. Kliknutím na tlačítko Recept otevřeme dialog pro přidání předpisu na lék. Zvolíme lék, vyplníme další potřebné údaje a dialog potvrdíme. Tím jsme přidali do záznamu předpis na lék.

Potvrzením dialogu se zdravotní kartou se záznam uloží a zároveň se zařadí do fronty pro odeslání do systému IZIP, což zkontrolujeme kliknutím na tlačítko IZIP v liště hlavního okna programu.

Nyní odešleme záznamy do systému IZIP. Klikneme na tlačítko IZIP a vybereme z nabízeného menu položku Odeslat data do IZIPu. Systém nás vyzve k zadání přístupového hesla uživatele. (Protože si vývojáři systému IZIP nepřejí, aby se hesla k testovacímu serveru volně šířila, toto heslo zde neuvedu. Návod na jeho získání bude uložen na doprovodném datovém nosiči.) Po zadání tohoto hesla se data odešlou a o výsledku jsme informováni zprávou.

Chceme-li zobrazit Zdravotní knížku pacienta, otevřeme jeho zdravotní kartu a v nástrojové liště této karty klikneme na tlačítko IZIP. V následujícím dialogu klikneme na tlačítko Otevřít ZK. Zadáme přístupové heslo a následně je spuštěn webový prohlížeč se Zdravotní knížkou pacienta.

8.9 Stav implementace projektu

Při implementaci projektu se ukázalo, že implementovat všechny navržené případy užití první iterace je velmi časově náročné a nakonec se je všechny implementovat nepodařilo. Také se nepodařilo sestavit uživatelskou příručku a návovědu. Ale protože práce na projektu diplomovou prací nekončí, bude vše dokončeno později.

Následující tabulka zobrazuje stav implementace jednotlivých navržených případů užití první iterace.

| Případ užití: | Stav: |
|--|--------------|
| Vytvoření karty pacienta | ✓ |
| Otevření karty pacienta | ✓ |
| Vypsání zdravotního záznamu (dekurzu) | ✓ |
| Připojení dokumentu ke zdravotnímu záznamu | ✓ |
| Úprava identifikačních údajů pacienta | ✓ |
| Zobrazení historie verzí identifikačních údajů pacienta | × |
| Výběr karty pacienta | ✓ |
| Zobrazení zdravotní knížky IZIP pacienta | ✓ |
| Zjištění stavu přihlášení pacienta do IZIP | ✓ |
| Přihlášení pacienta do IZIP | × |
| Hromadné zjištění stavu přihlášení pacientů do IZIP | × |
| Zobrazení historie očkování pacienta | × |
| Nové očkování pacienta | ✓ |
| Předepsání léčiv | ✓ |
| Prohlížení předepsaných léčiv pacienta | × |
| Editace rizik u pacienta | ✓ |
| Editace anamnézy pacienta | ✓ |
| Označení rodinného příslušníka pacienta | ✓ |
| Přidání dlouhodobé diagnózy do anamnézy pacienta | ✓ |
| Vymazání dlouhodobé diagnózy z diagnóz pacienta v anamnéze | ✓ |
| Zobrazení trvalých diagnóz pacienta | ✓ |
| Zaznamenání tlaku a pulsů | ✓ |
| Zobrazení historie hodnot tlaku a pulsů | × |
| Zaznamenání sedimentace | ✓ |
| Zobrazení historie hodnot sedimentace | × |
| Zaznamenání výšky a váhy | ✓ |
| Zobrazení historie hodnot výšky a váhy | × |
| Sestavení lékařské zprávy | ✓ |
| Zobrazení seznamu záznamů pro IZIP Odeslání dat do IZIP | ✓ |
| Změna osobních údajů uživatele | ✓ |
| Změna přístupového hesla uživatele | ✓ |

Tabulka 8.2: Stav implementace případů užití první iterace

Kapitola 9

Závěr

Účelem této práce byl návrh a implementace ambulantního informačního systému, který jsem po analýze požadavků zaměřil na ambulantní systém pro praktické lékaře. Hlavním účelem tohoto systému je komunikace se systémem IZIP. Tento úkol se mi podařilo splnit a implementovaný systém se systémem IZIP komunikuje.

Do budoucna počítám s rozšířením o další funkce, které by umožnily nasadit systém do skutečné ambulance, do ostrého provozu. Mezi důležité funkce patří tisk dokumentů, receptů, výkaznictví a další. Z pohledu plánu implementace se mi nepodařilo z časových důvodů implementovat všechny případy užití vybrané pro první iteraci. Tyto neimplementované případy se týkají práce se zaměstnavateli pacienta a historie identifikačních údajů pacienta.

Pro implementaci systému byly zvoleny takové komponenty, aby výsledný produkt mohl být nasazen na každý z rozšířených operačních systémů. Tato vlastnost je ale limitována komunikačním modulem IZIP, který je k dispozici pouze v binární podobě a to pro operační systémy Microsoft Windows a GNU/Linux. Pokud by se komunikační modul nahradil modulem vlastním, tato negativní vlastnost by se eliminovala.

Přínosem při návrhu tohoto systému pro mě samotného bylo seznámení se s chodem ordinace praktického lékaře, téměř všemi tiskopisy pro praktické lékaře, programy pro ordinace praktických lékařů dostupných na českém trhu, technikami návrhu informačních systémů a návrhovými vzory.

Pro značnou rozsáhlou a komplikovanou jsem se neseznámil s vyúčtováním poskytnuté zdravotní péče, ale věřím, že se při další práci na tomto projektu seznámím i s touto problematikou.

Samotná implementace byla pro mě také velkým přínosem. Využil jsem své znalosti programovacího jazyka C++ a vyzkoušel si práci v tomto jazyce na větším projektu. Tuto zkušenost jsem doposud postrádal. Seznámil jsem se knihovnamí pro práci s XML dokumenty, grafickou knihovnou Qt a databázovým systémem Firebird. Také jsem získal zkušenosti se začleňováním cizích komponent ve formě dynamicky linkovatelných knihoven do vlastního projektu.

Po dalších úpravách a implementacích bude tento program efektivně využitelný v ordinacích praktických lékařů a zaplní mezeru na trhu, kterou představují nekomerční ambulantní systémy. S velkou pravděpodobností bude mít i své webové stránky a snad se do vývoje zapojí i jiní programátoři.

Literatura

- [1] *Wikipedia – otevřená encyklopedie: Internet* [online]. [cit. 30. 12. 2007]. Dostupné z: <http://cs.wikipedia.org/wiki/Internet>.
- [2] *Wikipedia – otevřená encyklopedie: Protokol* [online]. [cit. 30. 12. 2007]. Dostupné z: http://cs.wikipedia.org/wiki/Protokol_%28informatika%29.
- [3] Zákon č. 227/2000 Sb. ze dne 29. června 2000 o elektronickém podpisu a o změně některých dalších zákonů, . §2.
- [4] Zákon č. 48/1997 Sb. ze dne 7. března 1997 o veřejném zdravotním pojištění a o změně a doplnění některých souvisejících zákonů, .
- [5] *Wikipedia – otevřená encyklopedie: MVC* [online]. [cit. 30. 12. 2007]. Dostupné z: <http://cs.wikipedia.org/wiki/MVC>.
- [6] ARLOW, J. – NEUSTADT, I. *UML 2 a unifikovaný proces vývoje aplikací*. Brno : Computer Press, a.s., 2007. ISBN 978-80-251-1503-9.
- [7] *Popis komunikačního rozhraní Portálu ZP*. Assecó Czech Republic, a.s. Dostupné z: <http://www.portalzp.cz/distribuce.ext/>. verze 1.3.2.12.
- [8] CÍSAŘ, P. *InterBase/Firebird – Tvorba, administrace a programování databází*. Brno : Computer Press, a.s., 2003. ISBN 80-7226-946-1.
- [9] COCKBURN, A. *Use Cases – Jak efektivně modelovat aplikace*. Brno : Computer Press, a.s., 2005. ISBN 80-251-0721-3.
- [10] DVOŘÁK, M. *Návrhové vzory (design patterns)* [online]. [cit. 30. 12. 2007]. poslední úpravy 16. června 2005. Dostupné z: <http://objekty.vse.cz/Objekty/Vzory>.
- [11] ECKEL, B. *Myslíme v jazyku C++*. 1. díl. Praha : Grada Publishing, a.s., 2000. ISBN 80-247-9009-2.
- [12] ECKEL, B. – ALLISON, C. *Myslíme v jazyku C++*. 2. díl. Praha : Grada Publishing, a.s., 2005. ISBN 80-247-1015-3.
- [13] FOUNDATION, F. *Firebird – The RDBMS that's going where you're going* [online]. [cit. 15. 5. 2008]. poslední úpravy 9. dubna 2008. Dostupné z: <http://www.firebirdsql.org/>.
- [14] HANÁČEK, P. – STAUDEK, J. *Bezpečnost informačních systémů*. Praha : ÚSIS, 2000. ISBN 80-238-5400-3.

- [15] *IZIP: Bezpečnost Vašich informací*. IZIP, a.s., . Dostupné z: http://www.izip.cz/index.php?p=14_5_1.
- [16] *Dokumentace IZIP*. IZIP, a.s., . Dostupné z: <http://www.izip.cz>. poslední úpravy březen 2004.
- [17] LÁZNIČKA, J. – RATHOUSKÝ, O. *Provozní řád IZIP* [online]. [cit. 30. 12. 2007]. poslední úpravy 1. listopadu 2007. Dostupné z: http://www.izip.cz/index.php?p=14_4.
- [18] *Datový standard MZ ČR 03.01.01*. Ministerstvo zdravotnictví ČR, . Dostupné z: <http://www.izip.cz/ds3/index.htm>. poslední úpravy 1. listopadu 2003.
- [19] *Datový standard MZ ČR 04.01.01*. Ministerstvo zdravotnictví ČR, . Dostupné z: http://ciselniky.dasta.mzcr.cz/CD_DS4/hypertext/DSBEX.htm. poslední úpravy 25. března 2008.
- [20] MLÁDEK, T. *Předávání informací ve zdravotnictví* [online]. [cit. 15. 5. 2008]. poslední úpravy 7. dubna 2008. Dostupné z: http://www.issz.cz/archiv/2008/download/prezentace/mladek_cnf.ppt.
- [21] MLÁDEK, T. *Přenos informací mezi pacienty a zdravotnickými pracovníky* [online]. [cit. 15. 5. 2008]. poslední úpravy 4. května 2004. Dostupné z: http://www.izip.cz/download/04.05.2004_prenosy_informaci.pps?PHPSESSID=a3.
- [22] INFOMED. *Informační systémy pro zdravotnická zařízení* [online]. [cit. 15. 5. 2008]. poslední úpravy 23. června 2005. Dostupné z: <http://www.infomed.cz/ps/article.php?arid=26>.
- [23] *Standard softwaru pro ordinace PL*. Společnost všeobecného lékařství ČLS JEP. Dostupné z: http://www.svl.cz/Files/nastenka/page_4766/Version1/Standart-softwaru.pdf.
- [24] *Datové rozhraní individuálních dokladů*. VZP ČR, . Dostupné z: http://www.vzp.cz/cms/internet/cz/Lekari/metodika/MDR/DR2_v62o.pdf. verze 6.2.
- [25] *Metodika pro pořizování a předávání dokladů VZP ČR*. VZP ČR, . Dostupné z: http://www.vzp.cz/cms/internet/cz/Lekari/metodika/MDR/MET_v62_22112006.pdf. verze 6.2.
- [26] *Tiskopisy pro komunikaci s pojišťovnou*. VZP ČR, . Dostupné z: <http://www.vzp.cz/cms/internet/cz/Lekari/metodika/tiskopisy/>.
- [27] *Číselník zdravotních pojišťoven*. VZP ČR, . Dostupné z: <http://www.vzp.cz/cms/internet/cz/Lekari/Ciselniky/640/Zdravotni-pojistovny/>. verze 640.
- [28] ZVÁROVÁ, J. et al. *Základy informatiky pro biomedicínu a zdravotnictví*. Praha : Nakladatelství Karolinum, 2002. ISBN 80-246-0609-7.

Dodatek A

Prohlášení o cílech

Cíle ambulantního informačního systému pro praktické lékaře:

- urychlení administrativní práce lékaře,
- zjišťování aktuálních nákladů za péči,
- zajištění veškeré komunikace s pojišťovnou,
- poskytování statistických výstupů,
- zajištění předávání dat s vnějšími subjekty (IZIP, laboratoře, ...).

Dodatek B

Požadavky na systém

B.1 Nefunkční požadavky

| id | požadavek | priorita |
|----|--|----------|
| 29 | jednoduchá instalace – bez potřeby instalace a nastavování dalších podpůrných služeb | možný |
| 30 | přehledné, srozumitelné grafické uživatelské rozhraní s kontextovou nápovědou | možný |
| 31 | srozumitelný manuál, příručka typu „Co dělat, když chci . . . “ | možný |
| 32 | užití osvědčených postupů při provádění úkonů v GUI, užití zavedených klávesových zkratk | možný |
| 33 | průvodce počátečním nastavením | možný |
| 34 | jednoduchá aktualizace programu | nezbytný |
| 35 | možnost rozvíjení programu v budoucnu | nezbytný |
| 36 | možnost přechodu z jiného SW a opačně | možný |
| 37 | ochrana údajů | nezbytný |
| 38 | bezpečnost dat (zálohování a rekonstrukce) | nezbytný |
| 39 | nízká cena | možný |
| 40 | přenositelnost | možný |
| 41 | minimální požadavky na HW | možný |

Tabulka B.1: Seznam nefunkčních požadavků

Hodnoty atributu priorita jsou vysvětleny v tabulce B.2, která je přejata z [6].

B.2 Funkční požadavky

Při výběru funkčních požadavků jsem čerpal z [23].

| Hodnota atributu priorita | Sémantika |
|------------------------------|---|
| Nezbytný | Povinné požadavky, jež jsou základem systému. |
| Možný | Důležité požadavky, které lze však vynechat. |
| Eventuální | Požadavky, jež jsou opravdu nepovinné (podávají se, je-li čas). |
| Chceme mít | Požadavky, které mohou být zahrnuty do dalších verzí systému. |

Tabulka B.2: Hodnoty atributu priorita v požadavcích systému

| id | požadavek | priorita |
|----|---|------------|
| 1 | vyřazování a znovuzařazování karet | nezbytný |
| 2 | řazení/vyhledávání karet podle parametrů | nezbytný |
| 3 | dělení karet do skupin | chceme mít |
| 4 | vedení evidence PN | možný |
| 5 | opravy údajů podle údajů ze ZP | eventuální |
| 6 | historie změn identifikačních údajů pacienta | možný |
| 7 | stav registrace pacienta + historie | možný |
| 8 | zviditelnění důležitých údajů o pacientovi | nezbytný |
| 9 | identifikace karet rodinných příslušníků pacienta | eventuální |
| 10 | sledování léčiv pacienta (historie předepsaných, množství, termíny dalšího předpisu) | možný |
| 11 | přikládání jiných dat (např. grafických) k zápisům v kartě pacienta | nezbytný |
| 12 | dispenzarizace pacientů | chceme mít |
| 13 | údržba číselníků (sledování platnosti, aktualizace) | možný |
| 14 | zápis zdravotních záznamů | nezbytný |
| 15 | sestavování faktur za provedenou péči | chceme mít |
| 16 | vykazování provedené péče (dávky pro pojišťovnu), kontrola nepovolených kombinací a frekvencí výkonů, archivace | eventuální |
| 17 | tisky přehledů a seznamů | eventuální |
| 18 | tisky obálek, štítků, podacích lístků, pozvánek | chceme mít |
| 19 | podpora při vyplňování zdravotnických dokladů | eventuální |
| 20 | zvýraznění nově zjištěných dat z jiných zdrojů | eventuální |
| 21 | příjem dat z laboratoří | eventuální |
| 22 | výměna dat se systémem IZIP | nezbytný |
| 23 | výměna dat s pojišťovnou | možný |
| 24 | import-export s jinými programy | možný |
| 25 | tvorba přehledů a statistik | eventuální |
| 26 | sledování nákladů | chceme mít |
| 27 | vedení zdravotní dokumentace v elektronické podobě | možný |
| 28 | urychlení pro psaní často užívaných textů | možný |

Tabulka B.3: Seznam funkčních požadavků

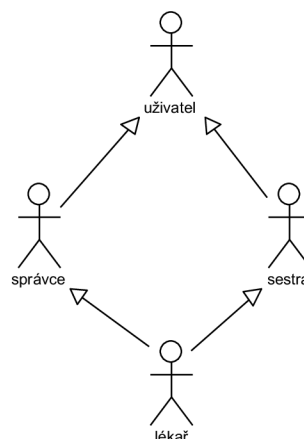
Dodatek C

Analýza požadavků na systém

C.1 Diagram případů užití

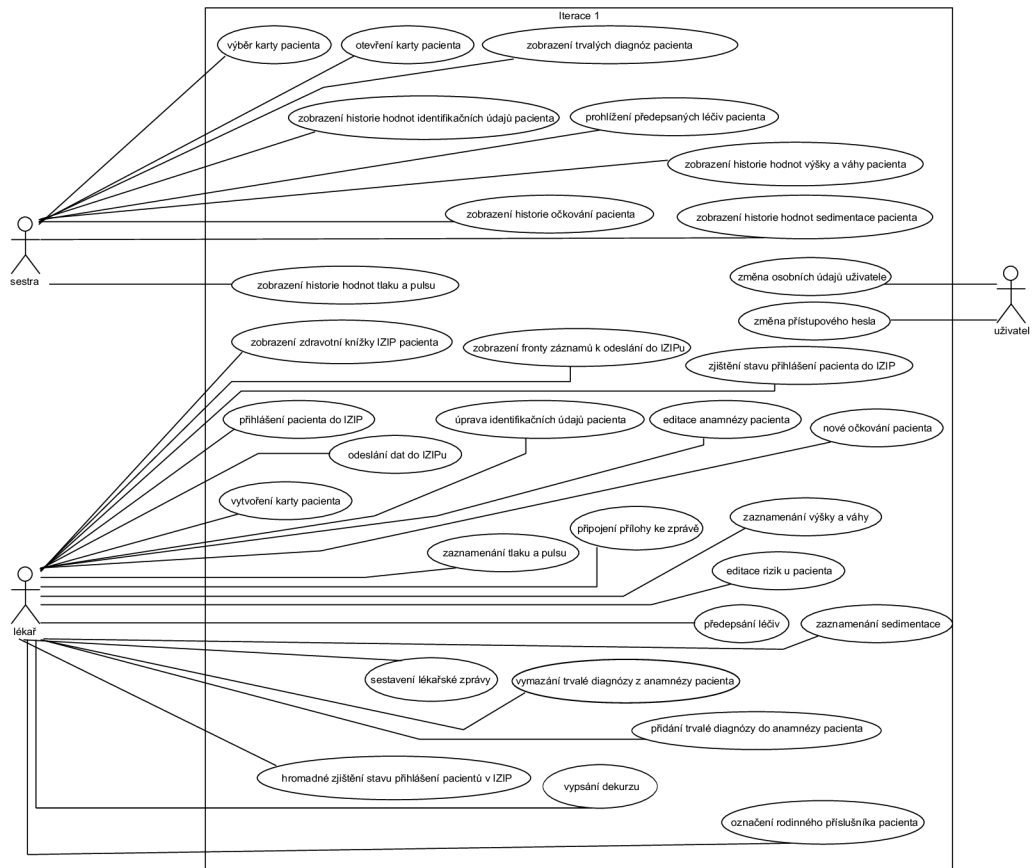
Diagramy případů užití byly rozděleny do čtyř iterací. V první iteraci jsou případy užití, které umožní zápis dat do systému, která následně mohou být odeslána do systému IZIP. Ve druhé iteraci jsou případy užití pro správu uživatelů, správu pacientů a různé administrační úkoly lékaře. Třetí iterace zahrnuje vyúčtování, poslední iterace zahrnuje import a export dat, komunikaci s pojišťovnou a laboratoří.

Na obrázku C.1 je diagram s aktéry systému. Aktér „uživatel“ je pouze abstraktní.

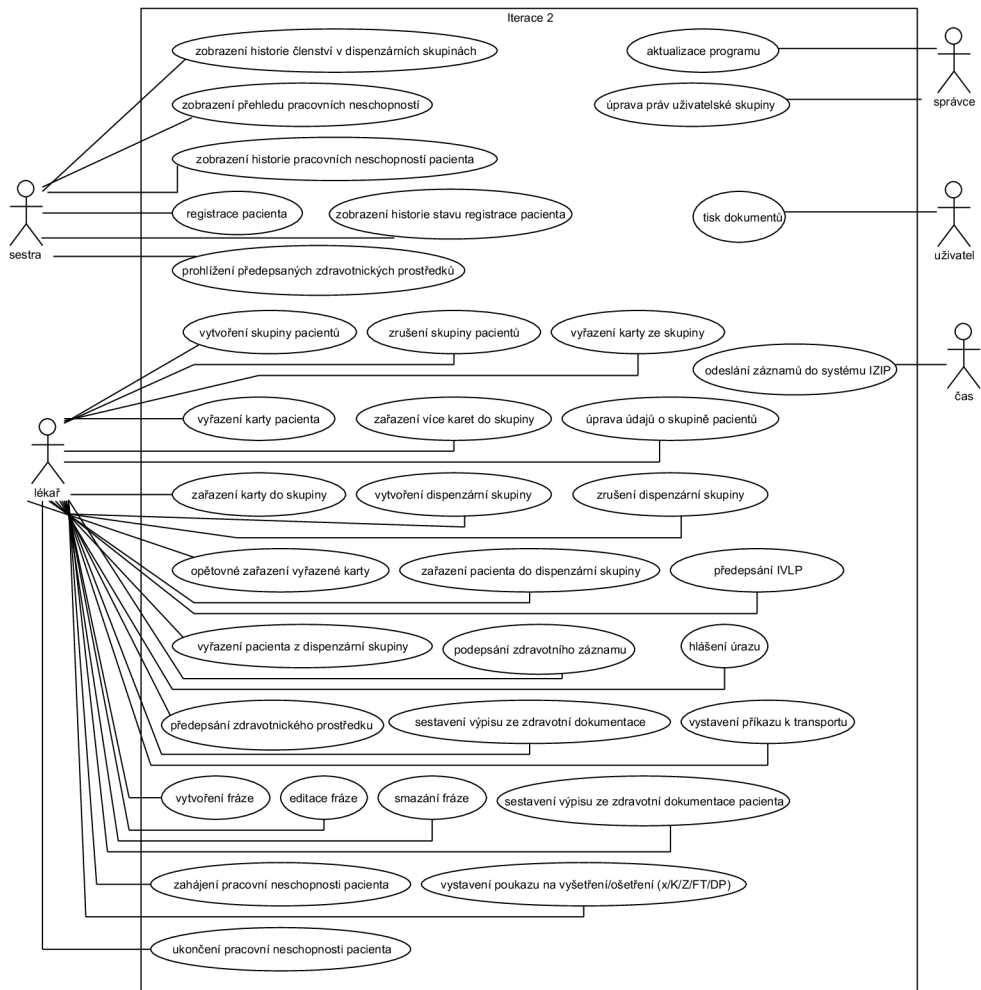


Obrázek C.1: Diagram aktérů užívajících systém

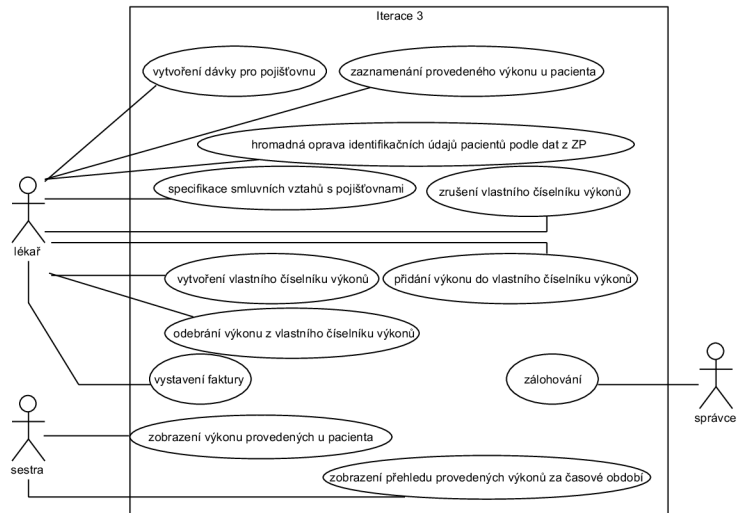
Následují jednotlivé diagramy užití.



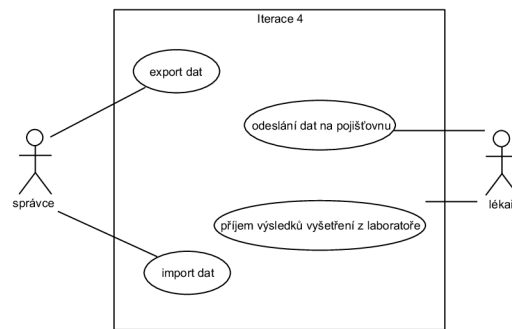
Obrázek C.2: Diagram případů užití první iterace



Obrázek C.3: Diagram případů užití druhé iterace



Obrázek C.4: Diagram případů užití třetí iterace



Obrázek C.5: Diagram případů užití čtvrté iterace

C.2 Popis diagramu případů užití první iterace

| | |
|------------------------------|---|
| Případ užití: | Vytvoření karty pacienta |
| ID: | UC1I01 |
| Stručný popis: | Vytvoření karty pacienta, který doposud není v systému evidován. Případ užití je specializován na vytvoření většího počtu karet, požaduje zadání pouze identifikačních údajů pacienta. |
| Hlavní aktéři: | lékař |
| Vedlejší aktéři: | žádní |
| Vstupní podmínky: | žádné |
| Hlavní scénář: | Případ užití je spuštěn lékařem příkazem „vytvořit novou kartu pacienta“. Dokud jsou identifikační údaje pacienta neplatné, žádá systém uživatele, aby zadal identifikační údaje pacienta. Systém vytvoří kartu pacienta. |
| Výstupní podmínky: | Pro pacienta byla vytvořena karta. |
| Alternativní scénáře: | Neplatný některý z identifikačních údajů. Karta pacienta již je v systému. Storno. |

| | |
|------------------------------|---|
| Případ užití: | Otevření karty pacienta |
| ID: | UC1I02 |
| Stručný popis: | Otevření karty pacienta se záznamy |
| Hlavní aktéři: | sestra |
| Vedlejší aktéři: | žádní |
| Vstupní podmínky: | vybrána karta pacienta v seznamu karet v systému |
| Hlavní scénář: | Případ užití je sestrou spuštěn příkazem „otevřít kartu pacienta“ na zvolené kartě v seznamu karet obsažených v systému. Systém zobrazí dialog s kartou pacienta a všemi záznamy. |
| Výstupní podmínky: | dialog s kartou pacienta a všemi záznamy |
| Alternativní scénáře: | žádné |

| | |
|------------------------------|--|
| Případ užití: | Vypsání zdravotního záznamu (dekurzu) |
| ID: | UC1I03 |
| Stručný popis: | Zapsání záznamu z návštěvy pacienta v ordinaci, či lékaře u pacienta. |
| Hlavní aktéři: | lékař |
| Vedlejší aktéři: | žádní |
| Vstupní podmínky: | otevřená karta pacienta |
| Hlavní scénář: | Případ užití je spuštěn otevřením karty pacienta. Je tedy vždy vytvořen nový záznam. Tento záznam je strukturovaný a skládá se z dalších připojených dokumentů. Má-li pacient aktivovanou knížku v IZIP, označí systém po jeho dokončení tento záznam jako nový záznam pro odeslání. |
| Výstupní podmínky: | nový záznam v kartě pacienta |
| Alternativní scénáře: | žádné |

| | |
|------------------------------|---|
| Případ užití: | Připojení dokumentu ke zdravotnímu záznamu |
| ID: | UC1I04 |
| Stručný popis: | Připojení jiného dokumentu (grafická data z vyšetření, zpráva jiného lékaře, ...) |
| Hlavní aktéři: | lékař |
| Vedlejší aktéři: | žádní |
| Vstupní podmínky: | otevřená karta pacienta |
| Hlavní scénář: | Případ užití je lékařem spuštěn příkazem „připojit dokument“. Systém zobrazí dialog pro vyhledání připojovaného dokumentu. Systém připojí uživatelem vybraný dokument a zobrazí ve zdravotním záznamu informaci o připojeném dokumentu. |
| Výstupní podmínky: | připojený dokument ve zdravotním záznamu pacienta |
| Alternativní scénáře: | Storno. |

| | |
|------------------------------|---|
| Případ užití: | Úprava identifikačních údajů pacienta |
| ID: | UC1I05 |
| Stručný popis: | Oprava některého z identifikačních údajů pacienta z důvodu zjištění chyby nebo změny |
| Hlavní aktéři: | lékař |
| Vedlejší aktéři: | žádní |
| Vstupní podmínky: | otevřená karta pacienta |
| Hlavní scénář: | Případ užití je spuštěn změnou některého z identifikačních údajů v kartě pacienta. Dokud je měněný údaj neplatný, systém žádá uživatele, aby zadal platný údaj. Systém porovná novou hodnotu s původní, liší-li se tyto hodnoty, přidá původní hodnotu do historie. |
| Výstupní podmínky: | Změněný údaj u pacienta, nová hodnota v historii měněné. |
| Alternativní scénáře: | Neplatná hodnota měněného identifikačního údaje. Storno. |

| | |
|------------------------------|--|
| Případ užití: | Zobrazení historie verzí identifikačních údajů pacienta |
| ID: | UC1I06 |
| Stručný popis: | Zobrazení historie verzí identifikačních údajů pacienta |
| Hlavní aktéři: | sestra |
| Vedlejší aktéři: | žádní |
| Vstupní podmínky: | otevřená karta pacienta |
| Hlavní scénář: | Případ užití je sestrou v otevřené kartě pacienta spuštěn příkazem „zobrazit historii verzí identifikačních údajů pacienta“. Systém zobrazí dialog s identifikačními údaji pacienta, ve kterém lze přepínat mezi jednotlivými verzemi. |
| Výstupní podmínky: | dialog s historií verzí identifikačních údajů |
| Alternativní scénáře: | Storno. |

| | |
|------------------------------|---|
| Případ užití: | Výběr karty pacienta |
| ID: | UC1I07 |
| Stručný popis: | Vyhledání konkrétní karty pacienta podle různých kritérií. |
| Hlavní aktéři: | sestra |
| Vedlejší aktéři: | žádní |
| Vstupní podmínky: | žádné |
| Hlavní scénář: | Případ užití je sestrou spuštěn pomocí příkazu „vyhledat kartu pacienta“. Systém zobrazí dialog, ve kterém uživatel specifikuje, podle kterého kritéria bude vyhledávat. Uživatel postupně zadává řetězec prohledávacího kritéria, systém postupně upravuje seznam karet, které tomuto kritériu vyhovují. Uživatel zvolí hledanou kartu pacienta. |
| Výstupní podmínky: | Vybraná karta pacienta |
| Alternativní scénáře: | Změna vyhledávacího kritéria. Storno. |

| | |
|------------------------------|---|
| Případ užití: | Zobrazení zdravotní knížky IZIP pacienta |
| ID: | UC1I08 |
| Stručný popis: | Zdravotní pracovník si může prohlédnout pacientovy záznamy v systému IZIP. |
| Hlavní aktéři: | lékař |
| Vedlejší aktéři: | IZIP |
| Vstupní podmínky: | otevřená zdravotní karta pacienta, pacient má aktivní zdravotní knížku v systému IZIP, zdravotní pracovník je registrován v systému IZIP, zdravotní pracovník má právo nahlížet do zdravotní knížky pacienta |
| Hlavní scénář: | Případ užití je lékařem spuštěn pomocí příkazu „zobrazit záznamy IZIP“ na kartě zvoleného pacienta s aktivovanou zdravotní knížkou. Systém se pokusí spojit z branou IZICHECK. Systém zkontroluje, zda pacient má stále aktivní ZK a zda je pro zdravotního pracovníka přístupná. Systém se spojí s pranou IZIVIEW a zobrazí ZK pacienta. |
| Výstupní podmínky: | Zobrazení ZK pacienta. |
| Alternativní scénáře: | Nemožnost navázat spojení z IZICHECK. Nemožnost přihlásit se na IZICHECK. Nemožnost odeslat dotaz na IZICHECK. Žádná odpověď od IZICHECK. Nahlášena chyba z IZICHECK. Pacient není registrován v systému IZIP. Pacient nemá aktivní ZK. Zdravotní pracovník nemá právo nahlížet do ZK pacienta. |

| | |
|------------------------------|---|
| Případ užití: | Zjištění stavu přihlášení pacienta do IZIP |
| ID: | UC1I09 |
| Stručný popis: | Systém zjistí, zda je pacient v IZIP registrován, zda má aktivní zdravotní knížku a zda ji může zdravotnický pracovník prohlížet. |
| Hlavní aktéři: | lékař |
| Vedlejší aktéři: | IZIP |
| Vstupní podmínky: | otevřená karta pacienta, zdravotní pracovník je registrován v systému IZIP |
| Hlavní scénář: | Případ užití je lékařem spuštěn pomocí příkazu „zkontrolovat stav pacienta v IZIP“ na kartě zvoleného pacienta. Systém se pokusí spojit z branou IZICHECK. Systém zkontroluje, zda je pacient v IZIP registrován. Systém zjistí, zda má pacient aktivní ZK. Systém zjistí, zda může zdravotnický pracovník do knížky nahlížet. Systém zobrazí a uloží zjištěné skutečnosti. |
| Výstupní podmínky: | upravený IZIP stav pacienta v systému |
| Alternativní scénáře: | Nemožnost navázat spojení z IZICHECK. Nemožnost přihlásit se na IZICHECK. Nemožnost odeslat dotaz na IZICHECK. Žádná odpověď od IZICHECK. Nahlášena chyba z IZICHECK. Storno. |

| | |
|------------------------------|--|
| Případ užití: | Přihlášení pacienta do IZIP |
| ID: | UC1I10 |
| Stručný popis: | Pacient sdělí zájem mít elektronickou zdravotní knížku. Systém pomůže s vyplněním přihlášky. |
| Hlavní aktéři: | lékař |
| Vedlejší aktéři: | žádní |
| Vstupní podmínky: | vybraná karta pacienta, který není registrován v systému IZIP; zdravotní pracovník je registrován v systému IZIP |
| Hlavní scénář: | Případ užití je lékařem spuštěn pomocí příkazu „registrovat pacienta do IZIP“ na kartě zvoleného pacienta. Systém zkontroluje, zda již nebyla přihláška vytištěna. Systém zkontroluje, zda má všechny potřebné údaje pro přihlášku. Systém vytiskne přihlášku pro pacienta do systému IZIP. Systém uloží k pacientovi poznámku, že byla vytištěna přihláška do IZIP. |
| Výstupní podmínky: | poznámka u pacienta o vytištění přihlášky do IZIP |
| Alternativní scénáře: | Nedávno vytištěná přihláška pacienta do IZIP. Storno. |

| | |
|------------------------------|--|
| Případ užití: | Hromadné zjištění stavu přihlášení pacientů do IZIP |
| ID: | UC1I11 |
| Stručný popis: | Hromadné zjištění, zda je pacienti jsou v IZIP registrováni, zda mají aktivní zdravotní knížku a zda do nich může zdravotnický pracovník prohlížet. |
| Hlavní aktéři: | lékař |
| Vedlejší aktéři: | IZIP |
| Vstupní podmínky: | zdravotní pracovník je registrován v systému IZIP |
| Hlavní scénář: | Případ užití je lékařem spuštěn pomocí příkazu „zkontrolovat stav pacientů v IZIP“. Systém zobrazí uživateli dialog s výběrem pacientů. Dokud uživatel nevybere alespoň jednoho pacienta, vyzývá systém uživatele k vybrání alespoň jednoho pacienta. Systém se spojí z branou IZICHECK. Systém odešle na bránu IZICHECK seznam uživatelem vybraných pacientů. IZICHECK odešle v odpovědi stavy registrace jednotlivých pacientů. Systém uloží nové stavy registrací pacientů v IZIP. Systém zobrazí uživateli zjištěné skutečnosti. |
| Výstupní podmínky: | Nové stavy registrací pacientů v systému IZIP. |
| Alternativní scénáře: | Nemožnost navázat spojení z IZICHECK. Nemožnost přihlásit se na IZICHECK. Nemožnost odeslat dotaz na IZICHECK. Žádná odpověď od IZICHECK. Nahlášena chyba z IZICHECK. Storno. |

| | |
|------------------------------|---|
| Případ užití: | Zobrazení historie očkování pacienta |
| ID: | UC1I12 |
| Stručný popis: | Systém zobrazí dialog se seznamem všech očkování pacienta s daty provedení. |
| Hlavní aktéři: | sestra |
| Vedlejší aktéři: | žádní |
| Vstupní podmínky: | otevřená karta pacienta |
| Hlavní scénář: | Případ užití je sestrou spuštěn pomocí příkazu „zobrazit očkování“ v otevřené kartě pacienta. Systém zobrazí dialog se seznamem všech očkování pacienta s daty provedení. |
| Výstupní podmínky: | Dialog s historií očkování vybraného pacienta. |
| Alternativní scénáře: | žádné |

| | |
|------------------------------|--|
| Případ užití: | Nové očkování pacienta |
| ID: | UC1I13 |
| Stručný popis: | Zdravotnický pracovník provede očkování pacienta a zanesse tuto skutečnost do systému. |
| Hlavní aktéři: | lékař |
| Vedlejší aktéři: | žádní |
| Vstupní podmínky: | otevřený dialog s historií očkování vybraného pacienta nebo otevřená karta pacienta |
| Hlavní scénář: | Případ užití je lékařem spuštěn pomocí příkazu „přidat očkování“ v otevřené dialogu s historií očkování pacienta. Systém zobrazí dialog s poli pro zadání typu očkování, očkovací látky, data a času očkování a diagnózy (nepovinná). Po potvrzení dialogu uloží systém nové očkování a je-li v kartě pacienta otevřen nový záznam (dekurz), připojí systém očkování k tomuto záznamu, jinak systém založí záznam nový. Má-li pacient aktivovanou knížku v IZIP, označí systém očkování jako nový záznam pro odeslání. |
| Výstupní podmínky: | nové očkování pacienta |
| Alternativní scénáře: | Storno. |

| | |
|------------------------------|--|
| Případ užití: | Předepsání léčiv |
| ID: | UC1I15 |
| Stručný popis: | Systém ulehčí lékaři předepsání léčiv. |
| Hlavní aktéři: | lékař |
| Vedlejší aktéři: | žádní |
| Vstupní podmínky: | otevřená karta pacienta |
| Hlavní scénář: | Případ užití je lékařem spuštěn v otevřené kartě pacienta pomocí příkazu „předepsat léky“. Systém zobrazí dialog s poli pro vyplnění receptu. Dokud nejsou informace kompletní, vyzývá systém uživatele k doplnění informací (uživatel vybere způsob úhrady, přípravek, počet balení, dávkování, návod k užití, datum předepsání a také, zda se jedná o trvale užívaný lék). Po ukončení dialogu vytiskne systém recept. Systém uloží předepsaná léčiva do otevřeného záznamu v pacientově kartě a zařadí je také do seznamu předepsaných léčiv. |
| Výstupní podmínky: | vytištěný recept, uložená předepsaná léčiva v pacientově kartě |
| Alternativní scénáře: | Předepsání léčivého přípravku s preskripčním omezením. Storno. |

| | |
|------------------------------|--|
| Případ užití: | Prohlížení předepsaných léčiv pacienta |
| ID: | UC1I16 |
| Stručný popis: | Zobrazení seznamu s předepsanými léčivy pacienta a daty jejich předpisu. Lze se přepínat mezi trvale užívanými léčivy a ostatními léčivy. |
| Hlavní aktéři: | sestra |
| Vedlejší aktéři: | žádní |
| Vstupní podmínky: | otevřená karta pacienta |
| Hlavní scénář: | Případ užití je sestrou spuštěn v otevřené kartě pacienta pomocí příkazu „zobrazit léky“. Systém zobrazí seznam s předepsanými léčivy pacienta a daty jejich předpisu. V dialogu je možné přepnutí mezi trvale předepisovanými léčivy a ostatními léčivy. V případě, že jsou vybraná trvale užívaná léčiva, informuje systém o době, kdy pacientovi léčiva dojdou. |
| Výstupní podmínky: | dialog s informacemi o léčivech |
| Alternativní scénáře: | žádné |

| | |
|------------------------------|--|
| Případ užití: | Editace rizik u pacienta |
| ID: | UC1I17 |
| Stručný popis: | Ke každému pacientovi je možné poznamenat různá fakta, která mají vliv na léčbu a která budou vždy dobře viditelné (např. alergie, ...) |
| Hlavní aktéři: | lékař |
| Vedlejší aktéři: | žádní |
| Vstupní podmínky: | otevřená karta pacienta |
| Hlavní scénář: | Případ užití je lékařem spuštěn v otevřené kartě pacienta pomocí příkazu „editovat rizika“. Systém zobrazí dialog se zaškrťávacím polem s vyjmenovanými riziky a textovým polem pro specifikaci jiných rizik. Po potvrzení dialogu systém uloží specifikovaná rizika. Má-li pacient aktivovanou knížku v IZIP označí systém rizika jako nové záznamy pro odeslání do IZIP. |
| Výstupní podmínky: | specifikovaná rizika u pacienta |
| Alternativní scénáře: | Storno. |

| | |
|------------------------------|--|
| Případ užití: | Editace anamnézy pacienta |
| ID: | UC1I18 |
| Stručný popis: | Případ užití popisuje zápis anamnézy do karty pacienta |
| Hlavní aktéři: | lékař |
| Vedlejší aktéři: | žádní |
| Vstupní podmínky: | otevřená karta pacienta |
| Hlavní scénář: | Případ užití je lékařem spuštěn pomocí příkazu „editovat anamnézu“. Systém zobrazí dialog pro editaci anamnézy. Anamnéza je rozdělena do více částí (např. na osobní, rodinnou a další). Uživatel může také označit rodinné příslušníky pacienta. Anamnéza je po ukončení zápisu systémem uložena. Má-li pacient aktivovanou knížku v IZIP označí systém anamnézu jako nový záznam pro odeslání do IZIP. |
| Výstupní podmínky: | anamnéza pacienta v systému |
| Alternativní scénáře: | Označení rodinného příslušníka pacienta. Přidání dlouhodobé diagnózy do anamnézy. Storno. |

| | |
|------------------------------|---|
| Případ užití: | Označení rodinného příslušníka pacienta |
| ID: | UC1I19 |
| Stručný popis: | Případ užití popisuje označení rodinného příslušníka pacienta, jehož karta je také v systému. |
| Hlavní aktéři: | lékař |
| Vedlejší aktéři: | žádní |
| Vstupní podmínky: | otevřený dialog s editací anamnézy pacienta |
| Hlavní scénář: | Případ užití je lékařem spuštěn pomocí příkazu „označit rodinného příslušníka“ v dialogovém okně s editací anamnézy pacienta. Systém zobrazí dialog s výběrem pacienta a textovým polem pro určení rodinného vztahu. Po potvrzení dialogu je rodinný příslušník přidán mezi rodinné příslušníky pacienta. |
| Výstupní podmínky: | označený rodinný příslušník pacienta |
| Alternativní scénáře: | Storno. |

| | |
|------------------------------|--|
| Případ užití: | Přidání dlouhodobé diagnózy do anamnézy pacienta |
| ID: | UC1I20 |
| Stručný popis: | Přidání dlouhodobé diagnózy mezi diagnózy pacienta v anamnéze |
| Hlavní aktéři: | lékař |
| Vedlejší aktéři: | žádní |
| Vstupní podmínky: | otevřený dialog s anamnézou pacienta |
| Hlavní scénář: | Případ užití je lékařem spuštěn příkazem „přidat dlouhodobou diagnózu“ v dialogu s anamnézou pacienta. Systém zobrazí dialog s výběrem diagnóz. Dokud uživatel nevybere jednu diagnózu, vyzývá systém uživatele k zadání dlouhodobé diagnózy. Uživatel může specifikovat dobu počátku diagnózy. Po potvrzení dialogu systém uloží diagnózu mezi trvalé diagnózy pacienta. Má-li pacient aktivovanou knížku v IZIP a je-li toto jediná změna v anamnéze pacienta, označí systém anamnézu jako nový záznam pro odeslání do IZIP. |
| Výstupní podmínky: | nová diagnóza v seznamu dlouhodobých diagnóz pacienta |
| Alternativní scénáře: | Storno. |

| | |
|------------------------------|--|
| Případ užití: | Vymazání dlouhodobé diagnózy z diagnóz pacienta v anamnéze |
| ID: | UC1I21 |
| Stručný popis: | Při chybném zadání diagnózy je potřeba diagnózu se seznamu odstranit. |
| Hlavní aktéři: | lékař |
| Vedlejší aktéři: | žádní |
| Vstupní podmínky: | otevřený dialog s anamnézou pacienta, vybraná dlouhodobá diagnóza |
| Hlavní scénář: | Případ užití je lékařem spuštěn příkazem „odstranit dlouhodobou diagnózu“ na vybrané diagnóze pacienta. Systém zobrazí potvrzovací dialog. Má-li pacient aktivovanou knížku v IZIP a je-li toto jediná změna v anamnéze pacienta, označí systém anamnézu jako nový záznam pro odeslání do IZIP. Systém diagnózu odstraní ze seznamu. |
| Výstupní podmínky: | odstraněná diagnóza se seznamu dlouhodobých diagnóz pacienta |
| Alternativní scénáře: | Storno. |

| | |
|------------------------------|--|
| Případ užití: | Zobrazení trvalých diagnóz pacienta |
| ID: | UC1I22 |
| Stručný popis: | Zobrazení trvalých diagnóz pacienta. |
| Hlavní aktéři: | sestra |
| Vedlejší aktéři: | žádní |
| Vstupní podmínky: | otevřená karta pacienta |
| Hlavní scénář: | Případ užití je sestrou spuštěn příkazem „zobrazit trvalé diagnózy“ na kartě pacienta. Systém zobrazí dialog s chronologicky seřazenými trvalými diagnózami pacienta. Ke každé diagnóze je zobrazen datum jejího počátku, pokud jsou tyto údaje k dispozici. |
| Výstupní podmínky: | zobrazený dialog s trvalými diagnózami |
| Alternativní scénáře: | žádné |

| | |
|------------------------------|---|
| Případ užití: | Zaznamenání tlaku a pulsu |
| ID: | UC1I23 |
| Stručný popis: | Případ užití popisuje zaznamenání tlaku a pulsu a jejich připojení ke zdravotnímu záznamu. |
| Hlavní aktéři: | lékař |
| Vedlejší aktéři: | žádní |
| Vstupní podmínky: | otevřená karta pacienta |
| Hlavní scénář: | Případ užití je lékařem spuštěn pomocí příkazu „zaznamenat tlak a puls“. Systém zobrazí dialog pro zadání tlaku a pulsu. Dokud nejsou hodnoty tlaku a pulsu korektní, vyzývá systém uživatele k zadání korektních údajů. Po potvrzení dialogu uloží systém zadané hodnoty do otevřeného záznamu v pacientově kartě a zařadí je také do historie hodnot tlaku a pulsu. |
| Výstupní podmínky: | hodnota tlaku a pulsu ve zdravotním záznamu pacienta |
| Alternativní scénáře: | Storno. |

| | |
|------------------------------|--|
| Případ užití: | Zobrazení historie hodnot tlaku a pulsu |
| ID: | UC1I24 |
| Stručný popis: | Názorné zobrazení vývoje tlaku a pulsu u pacienta |
| Hlavní aktéři: | sestra |
| Vedlejší aktéři: | žádní |
| Vstupní podmínky: | otevřená karta pacienta |
| Hlavní scénář: | Případ užití je sestrou spuštěn pomocí příkazu „zobrazit vývoj tlaku a pulsu“ v otevřené kartě pacienta. Systém zobrazí dialog s grafickým znázorněním vývoje tlaku a pulsu. |
| Výstupní podmínky: | zobrazený dialog s grafem vývoje tlaku a pulsu |
| Alternativní scénáře: | žádné |

| | |
|------------------------------|--|
| Případ užití: | Zaznamenání sedimentace |
| ID: | UC1I25 |
| Stručný popis: | Zaznamenání výsledku sedimentace krve. |
| Hlavní aktéři: | lékař |
| Vedlejší aktéři: | žádní |
| Vstupní podmínky: | otevřená karta pacienta |
| Hlavní scénář: | Případ užití je lékařem spuštěn pomocí příkazu „zaznamenat výsledky sedimentace“ v otevřené kartě pacienta. Systém zobrazí dialog pro zadání výsledku sedimentace. Dokud není hodnota korektní, vyzývá systém uživatele k zadání korektní hodnoty. Po potvrzení dialogu uloží systém zadanou hodnotu do otevřeného záznamu v pacientově kartě a zařadí ji také do historie hodnot sedimentace. |
| Výstupní podmínky: | výsledek sedimentace ve zdravotním záznamu pacienta |
| Alternativní scénáře: | Storno. |

| | |
|------------------------------|--|
| Případ užití: | Zobrazení historie hodnot sedimentace |
| ID: | UC1I26 |
| Stručný popis: | Názorné zobrazení vývoje sedimentace u pacienta |
| Hlavní aktéři: | sestra |
| Vedlejší aktéři: | žádní |
| Vstupní podmínky: | otevřená karta pacienta |
| Hlavní scénář: | Případ užití je sestrou spuštěn pomocí příkazu „zobrazit vývoj sedimentace“ v otevřené kartě pacienta. Systém zobrazí dialog s grafickým znázorněním vývoje sedimentace. |
| Výstupní podmínky: | zobrazený dialog s grafem vývoje sedimentace |
| Alternativní scénáře: | žádné |

| | |
|------------------------------|--|
| Případ užití: | Zaznamenání výšky a váhy |
| ID: | UC1I27 |
| Stručný popis: | Zaznamenání měření výšky a váhy pacienta. |
| Hlavní aktéři: | lékař |
| Vedlejší aktéři: | žádní |
| Vstupní podmínky: | otevřená karta pacienta |
| Hlavní scénář: | Případ užití je lékařem spuštěn pomocí příkazu „zaznamenat výšku a váhu“ v otevřené kartě pacienta. Systém zobrazí dialog pro zadání výšky a váhy. Dokud nejsou hodnoty korektní, vyzývá systém uživatele k zadání korektních hodnot. Po potvrzení dialogu uloží systém zadané hodnoty do otevřeného záznamu v pacientově kartě a zařadí je také do historie hodnot výšky a váhy |
| Výstupní podmínky: | hodnoty výšky a váhy ve zdravotním záznamu pacienta |
| Alternativní scénáře: | Storno. |

| | |
|------------------------------|--|
| Případ užití: | Zobrazení historie hodnot výšky a váhy |
| ID: | UC1I28 |
| Stručný popis: | Názorné zobrazení vývoje výšky a váhy pacienta. |
| Hlavní aktéři: | sestra |
| Vedlejší aktéři: | žádní |
| Vstupní podmínky: | otevřená karta pacienta |
| Hlavní scénář: | Případ užití je sestrou spuštěn pomocí příkazu „zobrazit vývoj výšky a váhy“ v otevřené kartě pacienta. Systém zobrazí dialog s grafickým znázorněním vývoje výšky a váhy. |
| Výstupní podmínky: | zobrazený dialog s grafem vývoje sedimentace |
| Alternativní scénáře: | žádné |

| | |
|------------------------------|--|
| Případ užití: | Sestavení lékařské zprávy |
| ID: | UC1I29 |
| Stručný popis: | Sestavení lékařské zprávy. |
| Hlavní aktéři: | lékař |
| Vedlejší aktéři: | žádní |
| Vstupní podmínky: | otevřená karta pacienta |
| Hlavní scénář: | Případ užití je lékařem spuštěn příkazem „napsat lékařskou zprávu“. Systém zobrazí dialog s editorem pro sestavení zprávy. Uživatel sestaví zprávu. Systém zprávu uloží a připojí ke zdravotnímu záznamu pacienta. |
| Výstupní podmínky: | uložená lékařská zpráva |
| Alternativní scénáře: | Storno. |

| | |
|------------------------------|--|
| Případ užití: | Zobrazení seznamu záznamů pro IZIP |
| ID: | UC1I31 |
| Stručný popis: | Zobrazení seznamu záznamů, které byly odeslány do systému IZIP a které odeslané teprve budou. |
| Hlavní aktéři: | lékař |
| Vedlejší aktéři: | žádní |
| Vstupní podmínky: | žádné |
| Hlavní scénář: | Případ užití je lékařem spuštěn příkazem „zobrazit frontu záznamů pro IZIP“. Systém zobrazí dialog tabulkou všech odeslaných/připravených k odeslání záznamů pro IZIP. |
| Výstupní podmínky: | dialog s přehledem záznamů pro IZIP |
| Alternativní scénáře: | žádné |

| | |
|------------------------------|--|
| Případ užití: | Odeslání dat do IZIP |
| ID: | UC1I32 |
| Stručný popis: | Záznamy pro IZIP se v systému shromažďují do té doby, než jsou automaticky či manuálně odeslány. Záznamy, které IZIP nepřijme, zůstávají ve frontě. |
| Hlavní aktéři: | čas, lékař |
| Vedlejší aktéři: | IZIP |
| Vstupní podmínky: | zdravotní pracovník je registrován v systému IZIP |
| Hlavní scénář: | Případ užití je vyvolán příkazem „odeslat data do IZIP“. Systém vybere z fronty záznamů k odeslání do IZIP ty záznamy, které mají být odeslány (podle data odeslání). Systém sestaví seznam pacientů, jejichž záznamy se budou odesílat. Systém zkontroluje pomocí IZICHECK možnost zápisu do knížek pacientů, jejichž záznamy se budou odesílat. Systém podle přijaté odpovědi upraví frontu záznamů, které se budou odesílat. Systém odešle záznamy na IZIGATE. Systém zpracuje odpověď. |
| Výstupní podmínky: | odeslané záznamy do IZIP, změněná fronta záznamů k odeslání, změněný seznam odeslaných záznamů |
| Alternativní scénáře: | Žádné záznamy k odeslání Nemožnost navázat spojení z IZICHECK Nemožnost přihlásit se na IZICHECK. Nemožnost odeslat dotaz na IZICHECK. Žádná odpověď od IZICHECK. Nahlášena chyba z IZICHECK. Do žádné knížky z pacientů, jejichž záznamy se mají odeslat, není možný zápis. Nemožnost navázat spojení z IZIGATE. Nemožnost přihlásit se na IZIGATE. Nemožnost odeslat data na IZIGATE. Žádná odpověď od IZIGATE. Nahlášena chyba z IZIGATE. |

| | |
|------------------------------|---|
| Případ užití: | Změna osobních údajů uživatele |
| ID: | UC1I33 |
| Stručný popis: | Případ užití slouží ke změně/zadání informací o uživateli. Množství zadavatelných informací je závislé na typu uživatelského účtu. |
| Hlavní aktéři: | uživatel |
| Vedlejší aktéři: | žádní |
| Vstupní podmínky: | žádné |
| Hlavní scénář: | Případ užití uživatelem spuštěn příkazem „zobrazit informace o uživateli“. Systém zobrazí dialog pro zobrazení/editaci informací o uživateli. Množství informací se liší podle typu uživatelského účtu. |
| Výstupní podmínky: | změněné údaje o uživateli |
| Alternativní scénáře: | žádné |

| | |
|------------------------------|--|
| Případ užití: | Změna přístupového hesla uživatele |
| ID: | UC1I34 |
| Stručný popis: | Případ užití slouží ke změně uživatelského hesla |
| Hlavní aktéři: | uživatel |
| Vedlejší aktéři: | žádní |
| Vstupní podmínky: | žádné |
| Hlavní scénář: | Případ užití uživatelem spuštěn příkazem „změnit přístupové heslo“ Systém zobrazí dialog pro zadání dosavadního a nového hesla. Poté nastaví nové heslo. |
| Výstupní podmínky: | změněné přístupové heslo uživatele |
| Alternativní scénáře: | chybné dosavadní heslo uživatele |

Dodatek D

Návrh systému

D.1 Návrh obrazovek systému

Z důvodu značné prostorové náročnosti jsou navržené obrazovky pouze na doprovodném disku CD-ROM. Jejich seznam s jednoduchým popisem zachycuje tabulka D.1.

| Název souboru: | Popis: |
|----------------|---|
| snap001.png | Dialog pro zadání/editaci adresy pacienta |
| snap002.png | Dialog se seznamem adres pacienta |
| snap003.png | Dialog s výběrem typem aktualizace |
| snap004.png | Dialog pro výběr součástí aktualizace |
| snap005.png | Dialog se seznamem dispenzárních skupin pacienta |
| snap006.png | Dialog se seznamem skupin pacienta |
| snap007.png | Dialog pro zadávání fráze |
| snap009.png | Dialog pro zadání přístupového hesla do IZIP |
| snap010.png | Dialog s historií členství v dispenzárních skupinách |
| snap011.png | Dialog s historií změn identifikačních údajů pacienta |
| snap012.png | Dialog s přehledem sedimentací pacienta |
| snap013.png | Dialog s přehledem tlaků a pulzů pacienta |
| snap014.png | Dialog s přehledem měření výšek a vah pacienta |
| snap015.png | Dialog pro vyplnění tiskopisu Hlášení úrazu |
| snap016.png | Dialog s informacemi o lékaři – identifikace |
| snap017.png | Dialog s informacemi o lékaři – vyúčtování |
| snap018.png | Dialog s informacemi o lékaři – nastavení |
| snap019.png | Dialog s informacemi o uživateli |
| snap020.png | Dialog se zdravotní kartou pacienta – identifikace |
| snap021.png | Dialog se zdravotní kartou pacienta – anamnéza |
| snap022.png | Dialog se zdravotní kartou pacienta – dekurz |
| snap023.png | Dialog pro zápis lékařské zprávy |
| snap024.png | Dialog pro vytvoření nové karty pacienta |
| snap026.png | Dialog pro zápis očkování |
| snap027.png | Dialog pro opravy neshod v údajích pacienta |
| snap028.png | Dialog pro vyplnění poukazu DP (1) |
| snap029.png | Dialog pro vyplnění poukazu DP (2) |

pokračování na následující stránce

Tabulka D.1 – pokračování z předchozí strany

| Název souboru: | Popis: |
|----------------|--|
| snap030.png | Dialog pro vyplnění poukazu DP (3) |
| snap031.png | Dialog pro vyplnění poukazu FT |
| snap032.png | Dialog pro vyplnění poukazu K (1) |
| snap033.png | Dialog pro vyplnění poukazu K (2) |
| snap034.png | Dialog pro vyplnění poukazu Z (1) |
| snap035.png | Dialog pro vyplnění poukazu Z (2) |
| snap036.png | Dialog pro vyplnění poukazu na vyšetření |
| snap037.png | Dialog pro vyplnění poukazu pro pracovní neschopnost |
| snap038.png | Dialog pro nastavení práv skupin |
| snap039.png | Dialog pro výběr příbuzného pacienta |
| snap040.png | Dialog pro vyplnění předpisu na IVLP |
| snap041.png | Dialog pro vyplnění předpisu na léčivý přípravek |
| snap042.png | Dialog pro vyplnění předpisu na pomůcku |
| snap043.png | Přihlašovací dialog |
| snap045.png | Dialog pro vyplnění poukazu na transport |
| snap046.png | Dialog pro připojení přílohy |
| snap047.png | Dialog pro zápis rizik pacienta |
| snap048.png | Dialog pro zápis sedimentace pacienta |
| snap049.png | Dialog s číselníkem diagnóz |
| snap050.png | Dialog se seznamem dispenzárních skupin |
| snap051.png | Dialog s číselníkem léčivých přípravků |
| snap052.png | Dialog se seznamem nalezených neshod v údajích pacientů |
| snap053.png | Dialog se seznamem očkování pacienta |
| snap054.png | Dialog se seznamem pacientů |
| snap055.png | Dialog se seznamem neschopností pacienta |
| snap056.png | Dialog se seznamem všech lékařem předepsaných neschopností |
| snap057.png | Dialog se seznamem všech lékařem předepsaných léčiv |
| snap058.png | Dialog se seznamem předepsaných zdrav. prostředků pacienta |
| snap060.png | Dialog se seznamem skupin pacientů |
| snap061.png | Dialog s číselníkem druhů očkování |
| snap062.png | Dialog se seznamem provedených výkonů u pacienta |
| snap063.png | Dialog se seznamem všech provedených výkonů lékařem za časové období |
| snap064.png | Dialog s číselníkem výkonů |
| snap065.png | Dialog se seznamem frází |
| snap066.png | Dialog s informacemi o stavu pacienta v systému IZIP |
| snap067.png | Dialog pro zadání tlaku a pulzu pacienta |
| snap068.png | Dialog pro zadání trvalé diagnózy pacienta |
| snap069.png | Dialog s výběrem dat pro export |
| snap070.png | Dialog s výběrem pacientů pro export |
| snap071.png | Dialog s výběrem dat pro import |
| snap072.png | Dialog pro vyplnění formuláře Výpis ze zdravotní dokumentace |

pokračování na následující stránce

Tabulka D.1 – pokračování z předchozí strany

| Název souboru: | Popis: |
|----------------|---|
| snap073.png | Dialog pro ukončení dispenzarizace pacienta |
| snap074.png | Dialog pro zadání výšky a váhy pacienta |
| snap075.png | Dialog pro vytvoření dispenzární skupiny |
| snap076.png | Dialog pro vytvoření skupiny pacientů |
| snap077.png | Dialog se seznamem zpráv pro IZIP |
| snap078.png | Dialog pro změnu uživatelského hesla |
| snap079.png | Dialog pro vyřazení karty pacienta |

Tabulka D.1: Seznam navržených obrazovek systému

D.2 Návrh schématu databáze

Pro implementaci systému bylo zvoleno užití relační databáze.

D.2.1 Popis schématu databáze

| | |
|-----------------|--|
| Tabulka: | Uzivatel |
| Popis: | Tabulka s uživateli, kteří jsou ve skupině Správce nebo Sestra. Tabulka obsahuje sloupce s informacemi o uživateli (přihlašovací jméno, heslo, jméno a příjmení, . . .). Primární klíč idUzivatel a login jsou unikátní v rámci celého systému. Sloupec urole rozlišuje skupinu uživatele. |

| | |
|-----------------|--|
| Tabulka: | Lekar |
| Popis: | Tabulka s uživateli, kteří jsou ve skupině Lékař. Tabulka obsahuje sloupce s identifikací uživatele (přihlašovací jméno, heslo, jméno a příjmení, tituly), adresou ordinace, bankovním spojením (cizí klíč idBanka do tabulky Banka), přístupovými údaji k systému IZIP a nastavením komunikace, identifikačními čísly IČP, IČPP, IČZ a IČO, systémovou cestou k osobnímu certifikátu a příznakem pro přihlašování do systému tímto certifikátem. Dále zde jsou sloupce pro nastavení spojení s autoritou časových razítek. Primární klíč idUzivatel a login jsou unikátní v rámci celého systému. |

| | |
|-----------------|---|
| Tabulka: | VerzeCiselniku |
| Popis: | Tabulka k udržování verzí číselníků obsažených v systému. Primárním klíčem je idVerzeCiselniku. Další sloupce slouží k uložení popisu verze a období platnosti. |

| | |
|-----------------|---|
| Tabulka: | Banka |
| Popis: | Tabulka s číselníkem peněžních ústavů v ČR. Primárním klíčem je idBanka, následuje sloupec s kódem peněžního ústavu a sloupec s názvem ústavu. Je verzovaný, cizí klíč idVerzeCiselniku do tabulky VerzeCiselniku udává verzi číselníku. |
| Tabulka: | Pojistovna |
| Popis: | Tabulka s číselníkem zdravotních pojišťoven. Primárním klíčem je idPojistovna, následují sloupce s kódem a názvem zdravotní pojišťovny. Je verzovaný, cizí klíč idVerzeCiselniku do tabulky VerzeCiselniku udává verzi číselníku. |
| Tabulka: | Pacient |
| Popis: | Tabulka s pacienty. Tabulka obsahuje sloupce s identifikačními údaji (jméno a příjmení, tituly, rodné číslo), stavem registrace v systému IZIP, posledním zjištěným počtem zpráv v systému IZIP, přístupem, datem registrace v systému IZIP, datem úmrtí (pro již nežijící), příznak vyřazení z kartotéky a důvod vyřazení. Cizí klíč idPojistovna z tabulky Pojistovna udává, u které pojišťovny je pacient pojištěn. Cizí klíč idPrislusost z tabulky Státní příslušnost udává státní příslušnost pacienta a cizí klíč idStav z tabulky Stav udává rodinný stav pacienta. |
| Tabulka: | Adresa |
| Popis: | Tabulka s adresami pacienta. Cizí klíč idPacient udává pacienta, o jehož adresu se jedná, atribut priorita udává prioritu adresy. |
| Tabulka: | RodinnýStav |
| Popis: | Tabulka s číselníkem rodinných stavů (svobodný/ženatý/rozvedený...) Je verzovaný, cizí klíč idVerzeCiselniku do tabulky VerzeCiselniku udává verzi číselníku. |
| Tabulka: | StatniPrislusnost |
| Popis: | Tabulka číselníkem státních příslušností. Je verzovaný, cizí klíč idVerzeCiselniku do tabulky VerzeCiselniku udává verzi číselníku. |
| Tabulka: | Pacient_Rodina |
| Popis: | K pacientovi lze přiřadit rodinné členy. První sloupec identifikuje pacienta, druhý pak jeho rodinného příslušníka. Oba sloupce jsou cizími klíči idPacient z tabulky Pacient. Atributem relace je Vztah – textový popis vztahu pacienta a rodinného příslušníka. |

| | |
|-----------------|--|
| Tabulka: | vazební tabulka Registrace |
| Popis: | Tabulka pro vyjádření vztahu M:N mezi pacientem a lékařem. Pacient může mít více registrací u stejného lékaře, avšak pouze jediná je aktuálně platná. V danou chvíli smí být pacient registrovaný pouze u jediného lékaře. Atributy vztahu jsou datum registrace a stav registrace. |
| Tabulka: | Diagnoza |
| Popis: | Číselník s diagnózami MKN 10. Primárním klíčem je idDiagnoza, dalšími sloupci jsou kód diagnózy a popis dané diagnózy. Je verzovaný, cizí klíč idVerzeCiselniku do tabulky VerzeCiselniku udává verzi číselníku. |
| Tabulka: | Lek |
| Popis: | Číselník s hromadně vyráběnými léčivými přípravky a potravinami pro zvláštní lékařské účely. Datové rozhraní určuje VZP. Je verzovaný, cizí klíč idVerzeCiselniku do tabulky VerzeCiselniku udává verzi číselníku. |
| Tabulka: | VyskaVaha |
| Popis: | Tabulka s hodnotami výšky a váhy pacienta. Primárním klíčem je idVyskaVaha, cizím klíčem je idDekurz z tabulky Dekurz. Pomocí tohoto klíče lze zjistit, kterému pacientovi míry patří a kdy byly pořízeny. Další sloupce tabulky slouží pro uložení hodnot výšky a váhy. |
| Tabulka: | TlakPulz |
| Popis: | Tabulka s hodnotami tlaku a pulzu pacienta. Primárním klíčem je idTlakPulz, cizím klíčem je idDekurz z tabulky Dekurz. Pomocí tohoto klíče lze zjistit, kterému pacientovi hodnoty patří a kdy byly pořízeny. Další sloupce tabulky slouží pro uložení hodnot tlaku a pulzu. |
| Tabulka: | Sedimentace |
| Popis: | Tabulka s hodnotami sedimentací pacienta. Primárním klíčem je idSedimentace, cizím klíčem je idDekurz z tabulky Dekurz. Pomocí tohoto klíče lze zjistit, kterému pacientovi hodnoty patří a kdy byly pořízeny. Další sloupce tabulky slouží pro uložení hodnot sedimentací a doby provádění sedimentace. |

| | |
|-----------------|--|
| Tabulka: | Dekurz |
| Popis: | Tabulka pro uložení záznamů o průběhu péče o pacienta. Primárním klíčem je idDekurz, další sloupce obsahují datum a čas návštěvy pacienta u lékaře, zápis zjištěných skutečností, kvalifikované časové razítko (pokud systém spolupracuje s autoritou časových razítek), elektronický podpis záznamu (pokud uživatel tuto možnost požaduje) cizí klíč idPacient do tabulky Pacient, který identifikuje pacienta, kterému daný záznam patří a cizí klíč idUzivatel do tabulky Lékař, který identifikuje lékaře, který záznam provedl. |

| | |
|-----------------|--|
| Tabulka: | Anamneza |
| Popis: | Tabulka pro uložení anamnézy pacienta. Primárním klíčem je idAnamneza. Další sloupce slouží k uložení textu anamnézy. Text je rozčleněn do 5 kategorií. Cizí klíč idPacient přiřazuje anamnézu k pacientovi. |

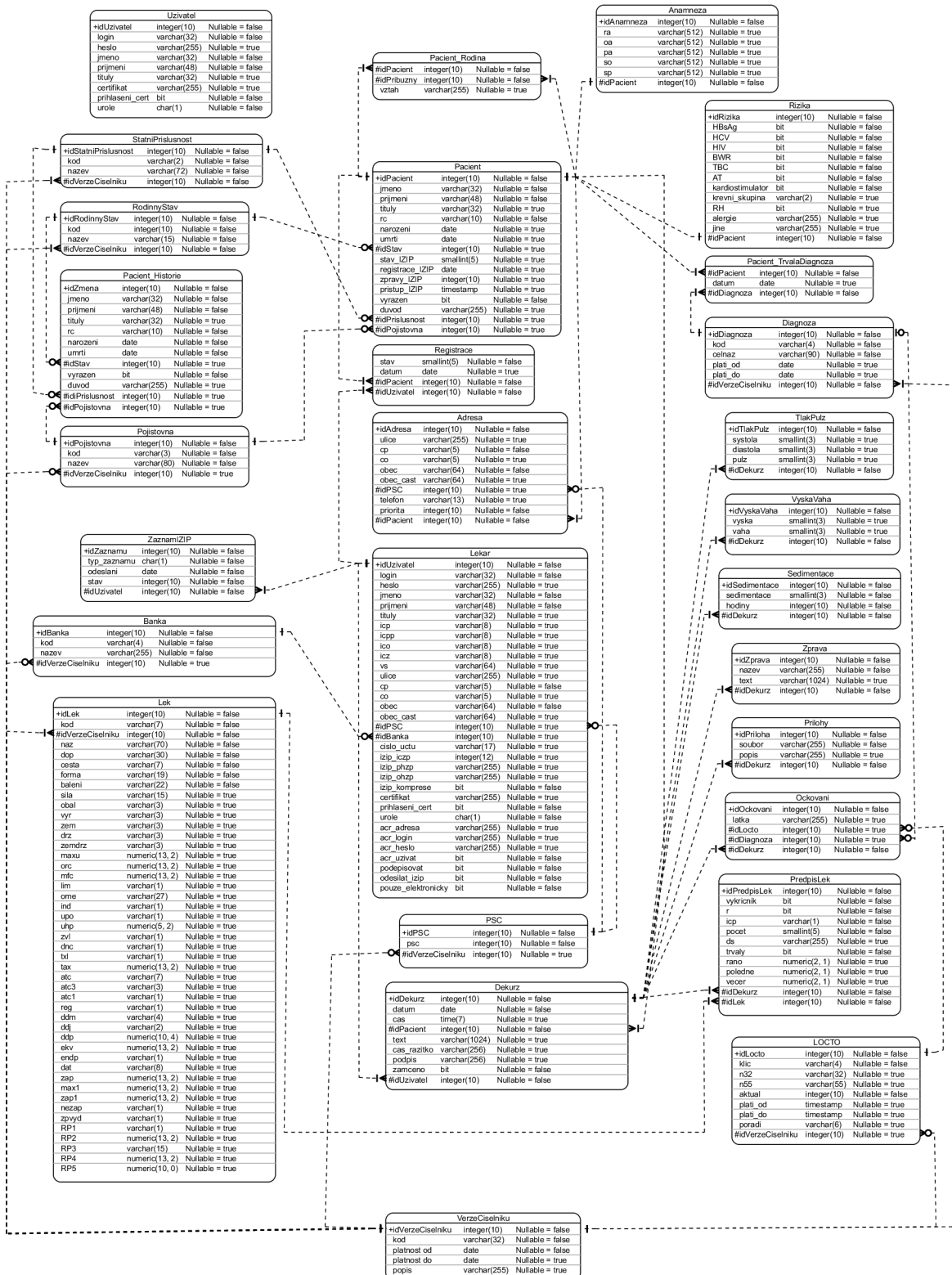
| | |
|-----------------|---|
| Tabulka: | Rizika |
| Popis: | Tabulka s riziky ovlivňující péči o pacienta. Primárním klíčem je idRizika. Následují sloupce s booleovskými hodnotami pro různé nemoci pacienta (TBC, HIV, ...), sloupec pro krevní skupinu a RH faktor a dva sloupce s textovými řetězci pro zápis alergií a jiných rizik. Cizím klíčem je idPacient z tabulky Pacient, který udává, kterému pacientovi rizika patří. |

| | |
|-----------------|---|
| Tabulka: | Ockovani |
| Popis: | Tabulka s očkováními aplikovanými pacientovi. Primárním klíčem je idOckovani, následuje očkovací látka. Cizí klíč idDekurz z tabulky Dekurz udává pacienta. Cizí klíč idDiagnoza z tabulky Diagnóza udává diagnózu (je volitelná), cizí klíč idLocto z tabulky LOCTO udává druh očkování podle datového standardu Ministerstva zdravotnictví české republiky. |

| | |
|-----------------|--|
| Tabulka: | Locto |
| Popis: | Číselník s typy očkování dle datového standardu Ministerstva zdravotnictví české republiky. Je verzovaný, cizí klíč idVerzeCiselniku do tabulky VerzeCiselniku udává verzi číslníku. |

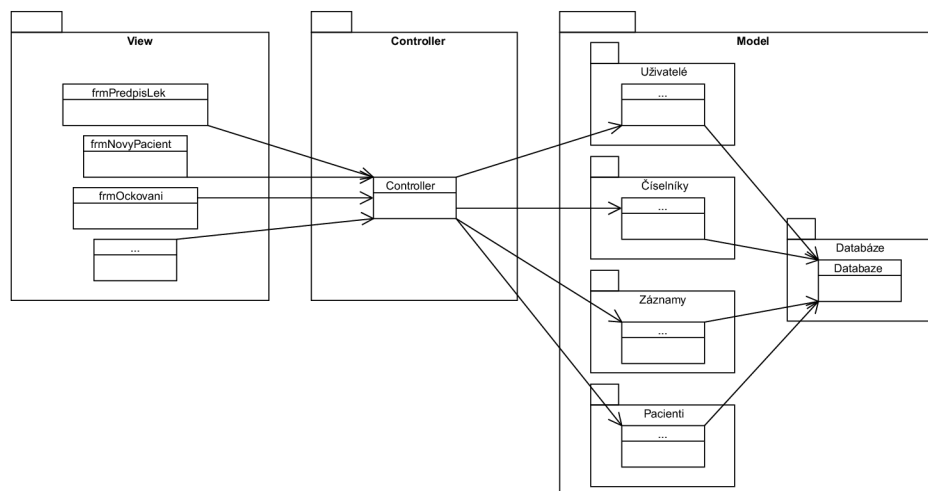
| | |
|-----------------|--|
| Tabulka: | PredpisLek |
| Popis: | Tabulka pro uložení informací o předepsaném léčivém přípravku. Primárním klíčem je idPredpisLek, následují sloupce pro uložení způsobu úhrady, počtu balení, návodu k užití. Cizí klíč idDekurz z tabulky Dekurz udává datum vystavení a pacienta, kterému byl předpis vydán, cizí klíč idLek z tabulky Lek označuje předepsaný léčivý přípravek. Sloupec Trvalý udává, zda byl přípravek označen jako pacientovi trvale předepisovaný. Sloupce Rano, Poledne, Vecer slouží pro uložení dávkování. (pro vyplnění tiskopisu VZP-10) |
| Tabulka: | ZaznamIZIP |
| Popis: | Tabulka se záznamy, které budou/byly odeslány do systému IZIP. Primárním klíčem je idZaznam. Tento sloupec obsahuje identifikaci záznamu, který bude odeslán – jeho hodnota je unikátní v rámci tabulek se zdravotními záznamy (Anamneza, Rizika, Ockovani, Zprava, ...). Cizí klíč idUzivatel z tabulky Lekar označuje lékaře, který záznam pořídil a přes jehož účet bude záznam do IZIP odeslán. Sloupec Datum odeslání označuje, kdy bude/byl záznam odeslán, Stav označuje, zda se záznam bude odesílat, kolik pokusů o odeslání se již konalo a podobně. |
| Tabulka: | Zprava |
| Popis: | Tabulka pro uložení sepsané lékařské zprávy. Primárním klíčem je idZprava, následuje sloupec s textem zprávy. Cizí klíč idDekurz označuje datum a pacienta, o kterém zpráva pojednává. |
| Tabulka: | Prilohy |
| Popis: | Tabulka pro uložení názvů souborů, které byly přiloženy k lékařskému záznamu (dekurzu). Primárním klíčem je idPriloha, následují sloupce pro uložení popisu přílohy a názvu souboru s přílohou. Cizí klíč idDekurz z tabulky Dekurz označuje záznam, ke kterému příloha patří. |
| Tabulka: | vazební tabulka Pacient_TrvalaDiagnoza |
| Popis: | Tabulka pro vyjádření vztahu M:N mezi pacientem a trvalou diagnózou. Atributem vztahu je zahájení diagnózy. |
| Tabulka: | Racient_Historie |
| Popis: | Tabulka pro ukládání změn v identifikačních údajích pacienta. Primárním klíčem je idZmena, pomocí něhož se dá vysledovat pořadí změn údajů. Cizí klíč idPacient do tabulky Pacient označuje pacienta, o jehož změny údajů se jedná. |

D.2.2 Diagram schématu databáze



Obrázek D.1: Schéma databáze

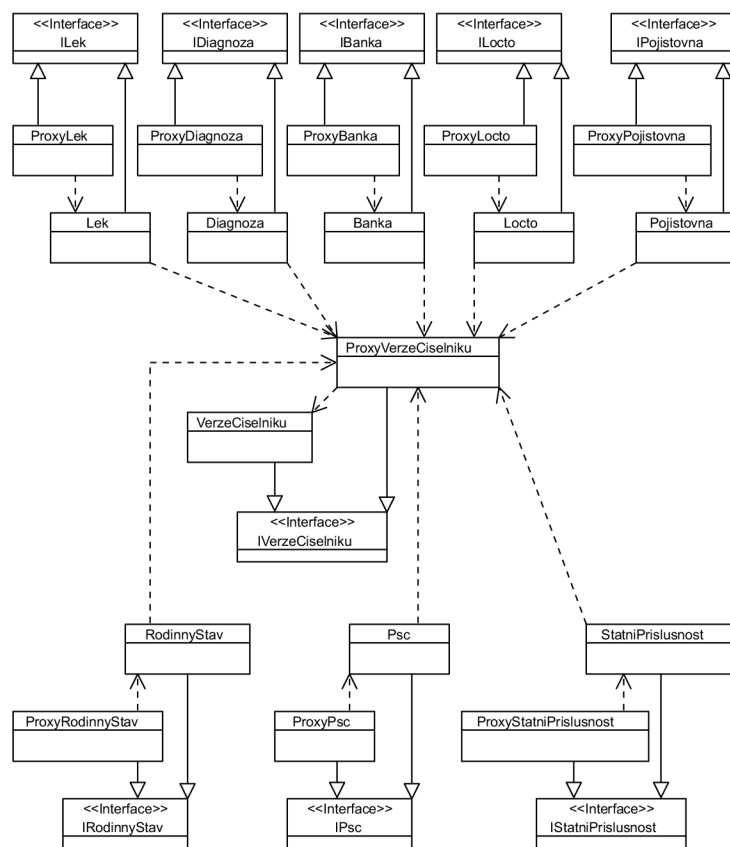
D.3 Návrh architektury systému



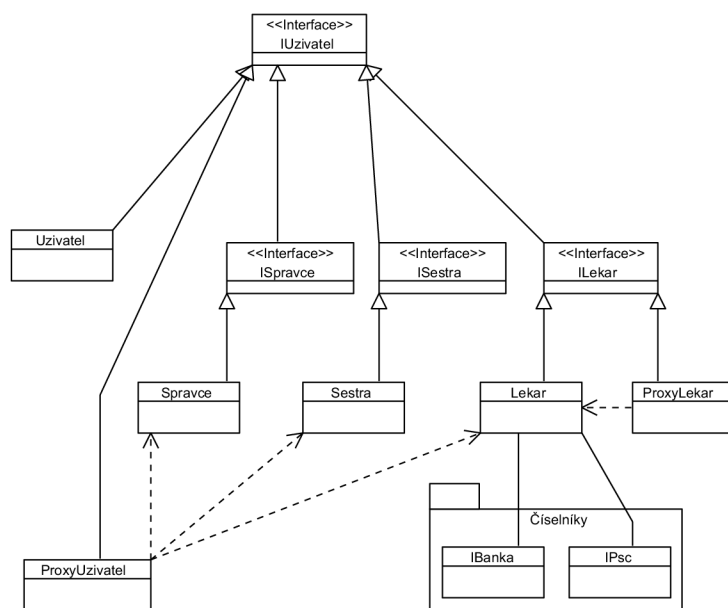
Obrázek D.2: Schéma architektury aplikace

D.3.1 Návrh jednotlivých vrstev architektury

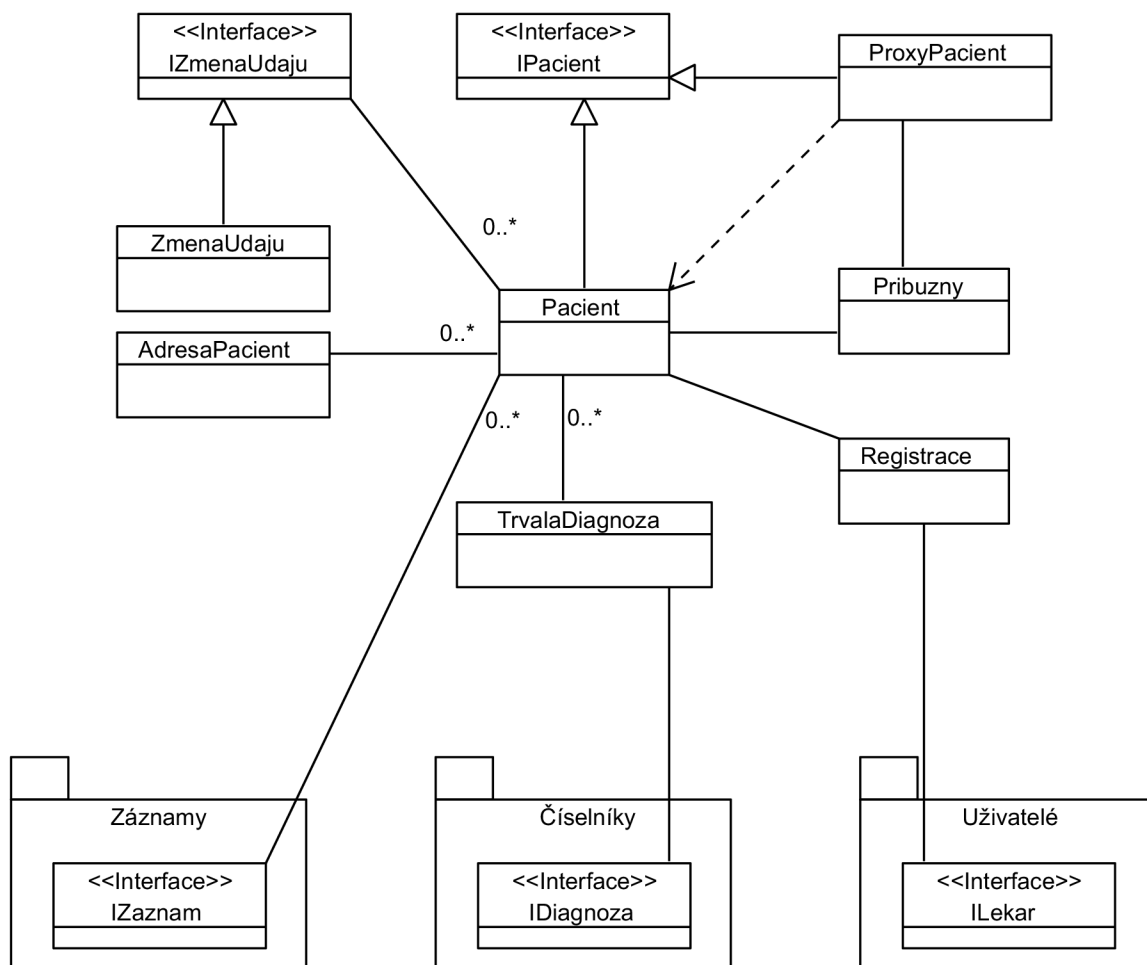
Návrh vrstvy Model



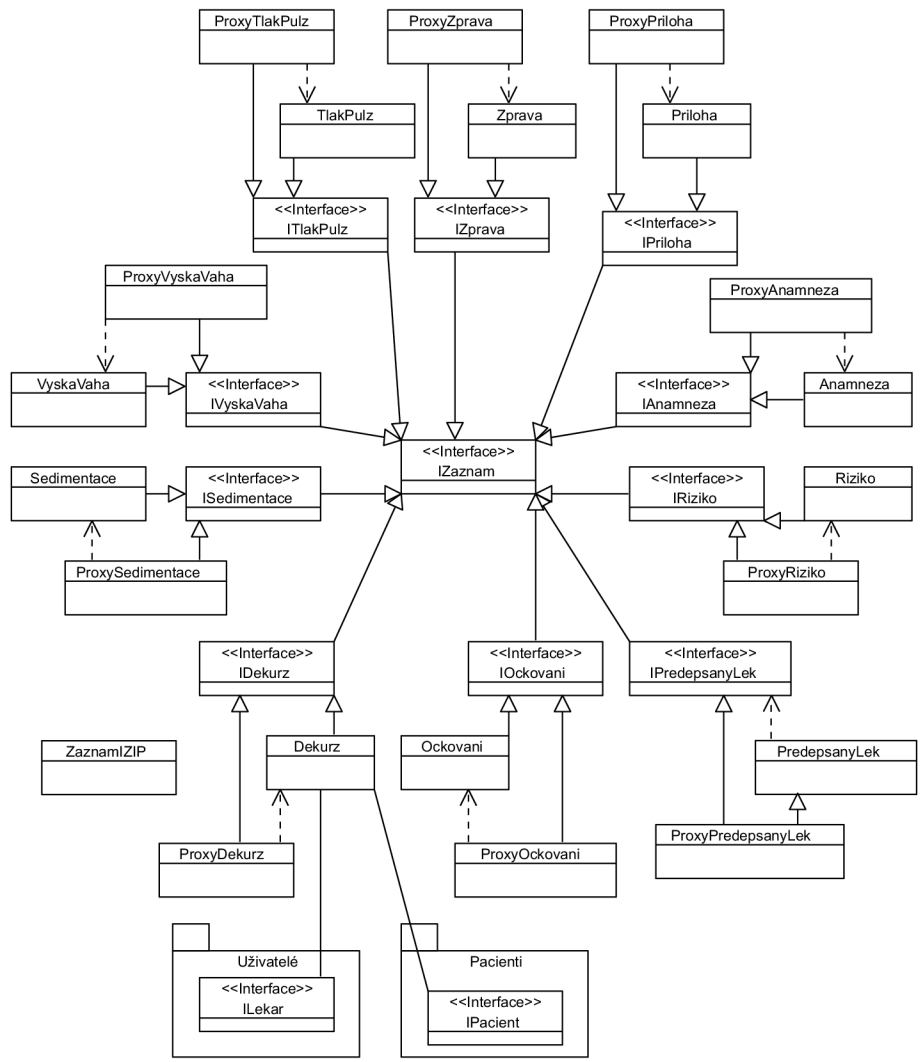
Obrázek D.3: Balíček Číselníky



Obrázek D.4: Balíček Uživatelé



Obrázek D.5: Balíček Pacienti



Obrázek D.6: Balíček Záznamy

Dodatek E

Obsah přiloženého disku

| Umístění: | Obsah: |
|----------------------|---|
| technicka_zprava | V tomto adresáři je umístěna technická zpráva (diplomová práce). |
| technicka_zprava/src | V tomto adresáři jsou umístěny zdrojové kódy pro vysázení technické zprávy v systému L ^A T _E X. |
| obrazovky program | V tomto adresáři jsou umístěny navržené dialogy systému. V tomto adresáři se nachází soubory vztahující se k programu. |
| program/doc | V tomto adresáři se nachází programová dokumentace (dokumentace k aplikačním třídám) vygenerovaná programem Doxygen. |
| program/bin | V tomto adresáři se nachází binární (spustitelné) formy programu. |
| program/bin/win32 | V tomto adresáři se nachází 32 bitová spustitelná verze programu pro operační systémy Microsoft Windows. |
| program/bin/linux32 | V tomto adresáři se nachází spustitelná verze programu pro 32 bitové systémy GNU/Linux. |
| program/src | V tomto adresáři se nachází zdrojové kódy programu. |