



**FAKULTA  
INFORMAČNÍCH  
TECHNOLOGIÍ  
ČVUT V PRAZE**

## ZADÁNÍ DIPLOMOVÉ PRÁCE

<b>Název:</b>	Revitalizace diagnostického nástroje v medicíně
<b>Student:</b>	Bc. Petra Krnáčová
<b>Vedoucí:</b>	Ing. Jakub Novák
<b>Studijní program:</b>	Informatika
<b>Studijní obor:</b>	Webové a softwarové inženýrství
<b>Katedra:</b>	Katedra softwarového inženýrství
<b>Platnost zadání:</b>	Do konce zimního semestru 2019/20

### Pokyny pro vypracování

Prozkoumejte existující webovou aplikaci pro detekci onemocnění. Aplikace umožňuje zadat výsledky testů pacienta, vyhodnotit onemocnění a zobrazovat výsledky pacientů.

Cílem práce je prozkoumat historickou implementaci algoritmu na detekci poruch štítné žlázy, identifikovat její nedostatky a navrhnout vlastní řešení formou modulu do webové aplikace.

Cíle práce:

- 1) Seznamte se s jinými aplikacemi podobného charakteru.
- 2) Seznamte se s algoritmem pro vyhodnocení poruch štítné žlázy a popište ho.
- 3) Prozkoumejte jeho historickou implementaci v jazyce Borland Pascal.
- 4) Identifikujte nedostatky implementace ve vztahu k aktuálním programovacím přístupům.
- 5) Navrhněte architekturu vlastního řešení, které těmito nedostatky netrpí.
- 6) Implementujte řešení moderním programovacím přístupem formou modulu do webové aplikace.
- 7) Otestujte vlastní řešení a porovnejte ho s historickou implementací.

### Seznam odborné literatury

Dodá vedoucí práce.

Ing. Michal Valenta, Ph.D.  
vedoucí katedry

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

V Praze dne 16. února 2018





**FAKULTA  
INFORMAČNÍCH  
TECHNOLOGIÍ  
ČVUT V PRAZE**

Diplomová práce

## **Revitalizace diagnostického nástroje v medicíně**

*Bc. Petra Krnáčová*

Katedra softwarového inženýrství

Vedúci práce: Ing. Jakub Novák

6. mája 2018



---

## Pod'akovanie

Pod'akovanie patrí vedúcemu mojej diplomovej práce Ing. Jakubovi Novákovi za dobré vedenie mojej práce a absolvované konzultácie, ďalej doc. RNDr. Ladislavovi Pecnovi, CSc. za konzultácie ohľadom algoritmu, mojej rodine za neustálu podporu počas štúdia a priateľom, ktorí mi spríjemňovali štúdium na ČVUT.



---

# Prehlásenie

Prehlasujem, že som predloženú prácu vypracoval(a) samostatne a že som uviedol(uviedla) všetky informačné zdroje v súlade s Metodickým pokynom o etickej príprave vysokoškolských záverečných prác.

Beriem na vedomie, že sa na moju prácu vzťahujú práva a povinnosti vyplývajúce zo zákona č. 121/2000 Sb., autorského zákona, v znení neskorších predpisov. V súlade s ustanovením § 46 odst. 6 tohoto zákona týmto udeľujem bezvýhradné oprávnenie (licenciu) k užívaniu tejto mojej práce, a to vrátane všetkých počítačových programov ktoré sú jej súčasťou alebo prílohou a tiež všetkej ich dokumentácie (ďalej len „Dielo“), a to všetkým osobám, ktoré si prajú Dielo užívať. Tieto osoby sú oprávnené Dielo používať akýmkoľvek spôsobom, ktorý neznižuje hodnotu Diela (vrátane komerčného využitia). Toto oprávnenie je časovo, územne a množstevne neobmedzené.

V Prahe 6. mája 2018

.....

České vysoké učení technické v Praze

Fakulta informačních technologií

© 2018 Petra Krnáčová. Všetky práva vyhradené.

*Táto práca vznikla ako školské dielo na FIT ČVUT v Prahe. Práca je chránená medzinárodnými predpismi a zmluvami o autorskom práve a právach súvisiacich s autorským právom. Na jej využitie, s výnimkou bezplatných zákonných licencií, je nutný súhlas autora.*

### **Odkaz na túto prácu**

Krnáčová, Petra. *Revitalizace diagnostického nástroje v medicíně*. Diplomová práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2018.



---

# Abstrakt

Táto diplomovaná práca sa zaoberá revitalizáciou programu DITHYRON určeného na diagnostiku porúch štítnej žľazy, ktorý vznikol v 90. rokoch minulého storočia pre operačný systém MS-DOS, do formy modulu modernej webovej aplikácie Alma Diagnostics. V rámci práce je uvedená podrobná analýza nedostatkov historickej implementácie, ktoré neumožňovali program v súčasnosti používať, rozširovať a vylepšovať. V práci sú zároveň analyzované existujúce aplikácie na pomoc s diagnostikou ochorení a problémy, ktoré limitujú ich rozšírenie. S využitím týchto informácií je navrhnutý a následne implementovaný modul na pomoc s diagnostikou porúch štítnej žľazy.

**Kľúčová slova** klinické systémy na podporu rozhodovania, poruchy štítnej žľazy, DITHYRON, Alma Diagnostics

---

# Abstract

The diploma thesis concerns with the renewal of the DITHYRON program designed for thyroid disorder diagnostics during 1990s for MS-DOS operating system into a module form of contemporary Alma Diagnostics web application. The thesis contains detailed analysis of historical implementation shortcomings, which caused the program to be unsuitable for present usage, expansion and improvements. The thesis also analyzes present applications that are used to help with the disease diagnostics and problems that limit their expansion. Based on these findings a module for help with thyroid disorders diagnostics was designed and implemented.

**Keywords** clinical decision support systems, thyroid disorders, DITHYRON, Alma Diagnostics

---

# Obsah

<b>Úvod</b>	<b>1</b>
Ciele práce . . . . .	1
<b>1 Analýza domény</b>	<b>3</b>
1.1 Medicínsky systém . . . . .	3
1.2 Diagnostika porúch štítnej žľazy . . . . .	8
1.3 CDSS, ktoré pomáhajú s diagnostikou porúch štítnej žľazy . . . . .	11
1.4 Komerčne používané CDSS . . . . .	15
<b>2 Algoritmus na detekciu porúch štítnej žľazy</b>	<b>21</b>
2.1 Potrebné pojmy . . . . .	21
2.2 Algoritmus . . . . .	30
<b>3 Analýza historickej implementácie programu DITHYRON</b>	<b>43</b>
3.1 Technológie . . . . .	43
3.2 Architektúra . . . . .	44
3.3 Business proces používania aplikácie – AS IS . . . . .	46
3.4 Nedostatky historickej implementácie . . . . .	47
<b>4 Analýza súčasnej webovej aplikácie Alma Diagnostics</b>	<b>57</b>
4.1 Architektúra systému . . . . .	57
4.2 Technológie . . . . .	58
4.3 Funkcionalita Alma Diagnostics aplikácie . . . . .	61
4.4 Komponenty Alma Diagnostics aplikácie . . . . .	62
4.5 Analýza požiadaviek na modul DITHYRON . . . . .	63
<b>5 Návrh modulu</b>	<b>69</b>
5.1 Zapojenie algoritmu do celkovej architektúry aplikácie . . . . .	69
5.2 Návrh častí . . . . .	71

<b>6</b>	<b>Realizácia</b>	<b>85</b>
6.1	Verzovanie . . . . .	85
6.2	Fázy implementácie . . . . .	85
<b>7</b>	<b>Testovanie, nasadenie a porovnanie</b>	<b>89</b>
7.1	Jednotkové testy . . . . .	89
7.2	Testovanie na reálnych dátach . . . . .	89
7.3	Integračné testovane . . . . .	90
7.4	Testovanie užívateľského rozhrania . . . . .	90
7.5	Nasadenie . . . . .	95
7.6	Záverečné porovnanie s historickou verziou . . . . .	95
<b>8</b>	<b>Diskusia</b>	<b>97</b>
	<b>Záver</b>	<b>99</b>
	<b>Literatúra</b>	<b>101</b>
<b>A</b>	<b>Zoznam použitých skratiek</b>	<b>107</b>
<b>B</b>	<b>Zoznam vstupných markerov a výstupných diagnóz programu DITHYRON</b>	<b>109</b>
<b>C</b>	<b>Užívateľské rozhranie modulu</b>	<b>111</b>
<b>D</b>	<b>Scenáre integračného testovania</b>	<b>113</b>
<b>E</b>	<b>Obsah priloženého CD</b>	<b>117</b>

---

## Zoznam obrázkov

1.1	GUI systému v Matlabe . . . . .	12
1.2	GUI ESTDD . . . . .	13
1.3	Architektúra OBESTDD . . . . .	14
1.4	GUI OBESTDD . . . . .	14
1.5	GUI Isabel – zadávanie symptómov . . . . .	16
1.6	GUI Isabel – prepojenie s medicínskymi encyklopédiami . . . . .	17
1.7	GUI DXplain . . . . .	17
1.8	GUI MDCalc . . . . .	18
2.1	Grafické znázornenie funkcie príslušnosti(2.18) . . . . .	26
2.2	Príklad bayesovskej siete pre náhodný vektor $(U,V,X,Y,Z)$ . . . . .	27
2.3	Naivná bayesovská sieť . . . . .	28
2.4	Rôzne odtiene azúrovej . . . . .	29
2.5	Diagram, zobrazujúci jednotlivé fázy algoritmu . . . . .	31
2.6	Ukážka, do ktorých intervalov môže patriť hodnota merania markeru, pričom farebné obdĺžniky znázorňujú okolie cut-off hodnoty . . . . .	34
2.7	Diagram, zobrazujúci kroky fuzzy naivného bayesovského klasifikátora . . . . .	36
2.8	Diagram, zobrazujúci proces aplikácie indexov . . . . .	38
2.9	Diagram, zobrazujúci proces zhodnotenia progresu ochorenia . . . . .	40
3.1	Diagram nasadenia historickej verzie programu DITHYRON . . . . .	45
3.2	Kompozícia funkcií a procedúr historickej verzie programu DITHYRON . . . . .	46
3.3	Proces nastavenia programu DITHYRON . . . . .	46
3.4	Proces vyšetrenia pacienta programom DITHYRON . . . . .	47
4.1	Diagram nasadenia aktuálnej webovej aplikácie Alma Diagnostics . . . . .	58
4.2	Procesný diagram vyšetrenia pacienta pomocou webovej aplikácie Alma Diagnostics . . . . .	61
4.3	Databázový model webovej aplikácie Alma Diagnostics . . . . .	63

4.4	Ukážka užívateľského rozhrania webovej aplikácie Alma Diagnostics	64
4.5	Prípady použitia: Vyšetrovanie a správa pacientov	66
4.6	Prípady použitia: Konfigurácia algoritmu	67
4.7	Prípady použitia: Užívateľské nastavenie	67
5.1	Diagram tried: Prepojenie triedy obalujúcej algoritmus s ostatnými časťami systému	70
5.2	Diagram tried: Komponenty algoritmu	72
5.3	Diagram tried: Podrobný popis triedy DithyronComputation	78
5.4	Databázový model modulu DITHYRON	78
5.5	Návrh riešenia problémov pri zadávaní meraní	81
5.6	Nevhodne zobrazený formulár na zadávanie meraní v aplikácii Alma Diagnostics	81
5.7	Návrh riešenia problému s nevhodne zobrazovaným formulárom	81
5.8	Návrh komponentu na nastavenie hodnôt referenčných intervalov	82
5.9	Návrh zobrazenia chovania markerov pomocou tabuľky	83
5.10	Návrh zobrazenia chovania markerov pomocou stĺpcového grafu	83
5.11	Návrh zobrazenia chovania markerov pomocou kruhového grafu	83
5.12	Finálny návrh zobrazenia chovania markerov pomocou kruhového grafu	84
5.13	Návrh rozhrania na zobrazovanie výsledkov vyšetrení	84
C.1	Ukážka rozhrania modulu Dithyron: Voľba modulu	111
C.2	Ukážka rozhrania modulu Dithyron: Voľba akcie	111
C.3	Ukážka rozhrania modulu Dithyron: Správa vyšetrení	111
C.4	Ukážka rozhrania modulu Dithyron: Vytvorenie nového pacienta	112
C.5	Ukážka rozhrania modulu Dithyron: Nastavenie hodnôt referenčných intervalov	112
C.6	Ukážka rozhrania modulu Dithyron: Výstup	112

---

## Zoznam tabuliek

2.1	Znalosti pre zhodnotenie priebehu liečby predchádzajúcej diagnózy	40
2.2	Znalosti pre zhodnotenie priebehu liečby predchádzajúcej diagnózy z meraní markerov . . . . .	41
3.1	Úkony nutné na odstránenie nedostatkov identifikovaných v historickej implementácii Dithyron . . . . .	55
7.1	Porovnanie historickej a súčasnej implementácie algoritmu na pomoc s diagnostikou porúch štítnej žľazy . . . . .	96
D.1	Kroky a vyhodnotenie scenára správy vyšetrení . . . . .	113
D.2	Kroky a vyhodnotenie scenára vyšetrenia pacienta . . . . .	114
D.3	Kroky a vyhodnotenie scenára importu údajov zo súboru a zmeny jazyka . . . . .	115
D.4	Kroky a vyhodnotenie scenára nastavenia hodnôt referenčných intervalov . . . . .	116
D.5	Kroky a vyhodnotenie scenára nastavenia výstupu administrátorom	116





---

# Úvod

Zdravie je jednou z najdôležitejších vecí v živote človeka. Dobré zdravie prináša človeku možnosť plnohodnotne prežívať život a venovať sa aktivitám, o ktoré má človek záujem. Liečenie množstva existujúcich ochorení je úloha, ktorá na špecialistov kladie vysoké nároky na znalosti, ktoré by mali poznať i aktívne využívať pri stanovovaní diagnózy.

Pri tomto procese by im mohli pomôcť počítače, konkrétne expertné systémy. Bohužiaľ, v rutínnej medicínskej praxi sa stále používa len minimum systémov na pomoc s diagnostikou, aj keď existuje množstvo validovaných algoritmov.

Takýmto algoritmom je aj algoritmus na detekciu porúch štítnej žľazy vytvorený doc. RNDr. Ladislavom Pecnom, CSc., ktorého zastaraná implementácia v podobe programu DITHYRON [47] pre operačný systém MS-DOS nie je v súčasnosti použiteľná.

## Ciele práce

Hlavným cieľom práce je revitalizácia programu DITHYRON vo forme modulu webovej aplikácie Alma Diagnostics [59].

Cieľom teoretickej časti práce je získanie náhľadu do problematiky medicínskych systémov, identifikácia problémov týchto systémov a analýza existujúcich systémov na pomoc s diagnostikou ochorení (najmä ochorení štítnej žľazy). Zároveň je nutné podrobne zanalyzovať historický program DITHYRON, identifikovať hlavné nedostatky vo vzťahu k použiteľnosti, rozšíriteľnosti a údržbe programu, a vytvoriť podrobný popis algoritmu použitého v programe, nakoľko v súčasnosti podrobný popis algoritmu chýba. Ďalším cieľom teoretickej časti je analýza webovej aplikácie Alma Diagnostics a formulácia požiadaviek na nový modul.

Cieľom praktickej časti je návrh a implementácia modulu na diagnostiku porúch štítnej žľazy do aplikácie Alma Diagnostics, ktorý nebude trpieť ne-

## ÚVOD

---

dostatkami historickej implementácie, jeho testovanie z hľadiska funkčnosti i používateľského rozhrania a porovnanie súčasnej a historickej implementácie.

---

# Analýza domény

Nasledujúca kapitola slúži ako úvod do problematiky medicínskych systémov a diagnostiky ochorení štítnej žľazy. Definuje potrebné pojmy, prináša prehľad používaných typov medicínskych systémov, pričom sa podrobnejšie zaoberá klinickými systémami na podporu rozhodovania (clinical decision support systems), popisuje spôsoby diagnostiky ochorení štítnej žľazy a analyzuje existujúce systémy, ktoré s diagnostikou ochorení pomáhajú.

## 1.1 Medicínsky systém

*Medicínske systémy pomáhajú doktorom a sestram diagnostikovať ochorenia a liečiť pacientov. Nemocnice a kliniky sa vo výraznej miere spoliehajú na počítačové technológie pri vyšetrovaní pacientov, liečení ochorení, výskume a podnikaní v zdravotníctve [1, vlastný preklad].*

### 1.1.1 Typy medicínskych systémov

Mnohé medicínske systémy ponúkajú viacero typov funkcionality, preto nie je jednoduchá úloha ich jednoznačne zaradiť do kategórií. Rovnako nie je jednoduché vymenovať všetky existujúce kategórie, pretože nové poznatky v počítačovej vede, ktoré preukážu technický či vedecký potenciál, sú rýchlo zakomponované aj do oblasti medicíny [2].

Rozdelenie systémov podľa práce [2] Hachaja a Ogiela je nasledovné:

- **Simulácie a modelovanie** – systémy tohto typu sa využívajú na skúmanie javov bez nutnej priamej interakcie s pacientom. Napríklad pomocou simulácie operácie je možné vyučovať mladých lekárov.
- **Vizualizácia dát** – v procese rozhodovania o pacientovej diagnóze lekári často využívajú rôzne obrazové dáta, napríklad snímky z počítačovej

tomografie (CT) a magnetickej rezonancie (MR). Systémy pre vizualizáciu dát sú určené na jednoduché a prehľadné zobrazovanie obrazových dát.

- **Pokročilé spracovanie dát** – pred samotnou vizualizáciou dát je častokrát nutné dáta spracovať. Príkladom môže byť algoritmus na detekciu nádoru na snímke, ktorý používa metódy spracovania obrazu.
- **Klinické systémy na podporu rozhodovania (CDSS)** – pomáhajú lekárovi pri práci. Môžu byť prospešné pri stanovovaní diagnózy pacienta či predpisovaní lieku.
- **Systémy na archiváciu snímok a komunikáciu (Picture Archiving and Communication Systems)** – poskytujú úložisko pre obrazové snímky, ktoré je jednoducho prístupné z rôznych typov zariadení.
- **Vzdialené monitorovanie** – systémy zložené zo senzorov umiestnených na tele pacienta a aplikácie, ktorá spracováva dáta zo senzorov.
- **Systémy na vzdelávanie pacientov** – môžu byť použité na interaktívne vysvetlenie testov čakajúcich pacienta.
- **Vedecké databázy** – slúžia na uchovávanie poznatkov. Príkladom je databáza liekov a liečiv.
- **Systémy na riešenie zložitých problémov** – niektoré problémy sú príliš výpočtovo zložité a nie je možné ich vyriešiť na bežnom počítači lekára. Pre tieto problémy sú navrhnuté špecializované algoritmy využívajúce výkonné paralelné platformy.
- **Rozhrania človek - počítač** – pomocou rozpoznávania gest a pohybu ruky chirurga umožňujú napríklad vykonanie operácie na diaľku.
- **Objavovanie liekov** – systémy, ktoré skúmajú chemické interakcie a molekulovú štruktúru. Pomáhajú pri vývoji nových liekov.

Okrem vyššie uvedených typov systémov je vhodné spomenúť aj **medicínske informačné systémy**, ktoré využívajú uvedené systémy a zároveň ponúkajú nástroje na správu pacientov, vyšetrení a komunikáciu v rámci inštitúcie.

Fenoménom dnešnej doby je prechod z papierovej formy dokumentácie o pacientovi, prípadne lokálnej počítačovej databázy u lekára, na centralizované elektronické záznamy o pacientoch tzv. **electronic health records (EHR)**. EHR zjednodušujú výmenu informácií o pacientoch medzi inštitúciami a zlepšujú informovanosť pacienta o jeho zdravotnom stave [3].

### 1.1.2 Klinické systémy na podporu rozhodovania (CDSS)

CDSS sú expertné systémy, ktoré sú špeciálne navrhnuté tak, aby pomáhali v procese rozhodovania sa v medicínskej oblasti [4].

*Expertné systémy* sa snažia nahradiť praktické znalosti a uvažovanie špecialistu v nejakom obore. Na tento účel využívajú vhodne formalizované a počítačovo zakódované znalosti prevzaté od špecialistu. Znalosti sú aplikované na riešenie zložitých problémov pomocou metód počítačovej vedy [5].

#### 1.1.2.1 História

S rozvojom počítačovej vedy v skorých 60. rokoch minulého storočia, vyvolala myšlienka, že počítače budú využívané na riešenie zložitých problémov medicíny, vlnu záujmu a vzrušenia a bola jedným z hnacích síl pre používanie počítačov. Mnoho vízií ohľadom automatickej diagnostiky, robotických doktoroch či strate súkromia bolo formulovaných [3].

Počas obdobia entuziazmu v rokoch 1960 – 1980 boli formulované a skúšané metódy, ktoré by mohli pomôcť s rozhodovaním v medicínskej oblasti [3]. Boli vyvinuté prototypy systémov, napríklad MYCIN a INTERNIST-1. MYCIN pomáhal s diagnostikou a liečbou bakteriálnych infekcií. INTERNIST-1 bol navrhnutý na diagnostiku ochorení v oblasti internej medicíny, pričom jeho databáza zahrňovala vyše 500 druhov ochorení [6].

V rokoch 1980 – 2000 vznikalo mnoho ďalších CDSS používajúcich pokročilejšie metódy, zároveň sa uskutočňovali rozsiahle štúdie na vyhodnotenie úspešnosti existujúcich CDSS. Najvýraznejšou bola snaha zakomponovať CDSS do rutínnej medicínskej praxe. Tento cieľ sa však nepodarilo naplniť (aj kvôli problémom popísaným v podkapitole 1.1.2.3) [3].

V tomto období vzniklo aj niekoľko systémov, ktoré sa podarilo komercionalizovať [3]. Takýmto zástupcom je systém DxPlain (viac v podkapitole 1.4.2). Rovnako vznikli programy BIANTA a CRACTES, ktoré tvoria súčasť novodobej webovej aplikácie Alma Diagnostics (viac v kapitole 4), a program DITHYRON, ktorého revitalizácia je cieľom tejto diplomovej práce.

Od roku 2000 až po súčasnosť sa znovu obnovuje entuziazmus do vývoja nových CDSS vďaka rozmachu technológii umožňujúcich využívanie cloudových riešení, webových aplikácií, mobilných telefónov a tabletov, zavádzaniu EHR a celkovej transformácii zdravotníctva za pomoci grantov na jeho informatizáciu [3].

#### 1.1.2.2 Typy CDSS

CDSS je možné deliť podľa metód, ktoré využívajú na [6]:

- **CDSS založené na znalostiach**
- **CDSS využívajúce štatistické metódy**

- **Inteligentné výpočtové systémy**
- **Hybridné CDSS**

Typickým zástupcom **CDSS založených na znalostiach** je systém, ktorého znalosti sú vyjadrené pomocou if-then pravidiel. Pravidlá môžu byť dynamicky vytvárané alebo extrahované zo stromovej štruktúry či tabuľky [6].

Iný typ funguje na základe porovnávania informácií týkajúcich sa súčasného prípadu s informáciami obsiahnutými v predchádzajúcich prípadoch, pričom lekárovi navrhuje postupy, ktoré fungovali v prípadoch s najvyššou zhodou [6].

CDSS môže využívať aj znalosti získané z biochemických alebo biofyzických modelov, napríklad molekulárnu štruktúru látok [6].

**CDSS využívajúce štatistické metódy** pracujú so znalosťami získanými z predchádzajúcich prípadov. Aplikujú štatistické metódy na určenie pravdepodobnosti javov (napríklad diagnóz) alebo sa snažia o klasifikáciu nového pozorovania do vopred určených kategórií. Typicky sa využívajú metódy založené na bayesovských sieťach (viac v podkapitole 2.1.3) [6]. Tiež je možné použiť klasifikátory ako support vector machines (SVM) alebo klasifikátor založený na metóde  $k$ -najbližších susedov ( $k$ -nn).

**Inteligentné výpočtové systémy** napodobňujú procesy inšpirované prírodou na riešenie problémov, ktoré majú príliš vysokú zložitosť a použitie exaktných matematických (štatistických) metód je nepostačujúce vzhľadom na nelinearitu, mnohorozmernosť, neočakávané či neznáme javy procesu generujúceho dáta problému. Používané metódy zahŕňajú genetické algoritmy, neurónové siete alebo fuzzy logiku (viac v podkapitole 2.1.2) [7].

**Hybridné CDSS** nepoužívajú jeden prístup, ale kombinujú rôzne metódy [6].

### 1.1.2.3 Problémy CDSS

Navzdory tomu, že bolo na akademickej pôde vyvinutých mnoho CDSS, veľká väčšina konceptov neprenikla do komerčnej sféry [3].

Medicínske informačné systémy majú v sebe zakomponovanú funkcionálnu jednoduchosť CDSS najmä v oblasti liekových interakcií, dávkovaní liekov, odkazov smerujúcich na lekárske encyklopédie, výpočtov, pripomenutí a upozornení (napríklad na možné vynechanie dávky) [3].

Napriek rokom akademického výskumu (viď podkapitola 1.1.2.1) je však v praxi používaných minimum komplexnejších CDSS, ktoré pomáhajú s diagnostikou ochorení [3]. Príčin je niekoľko.

CDSS bývajú v akademickej sfére implementované vo forme prototypu, ktorý má preukázať efektivitu zvolených metód. Projekty sú často vedené

odborníkom z oboru medicíny, ktorého znalosti sú základom systému a nie špecialistami z oblasti informačných technológií [3].

Pozornosť nebýva zameraná na dizajn, architektúru a použiteľnosť systému, ale na metódy samotné [3].

Nevhodne navrhnutá, prípadne neexistujúca, architektúra komplikuje prepojenie s existujúcimi medicínskymi informačnými systémami. Nedostatočná fáza plánovania a návrhu aplikácie spôsobuje problémy s udržateľnosťou programu. Zároveň znalosti bývajú často zakomponované priamo do výkonného kódu a nie je možné ich jednoducho pozmeniť alebo doplniť [3].

Prepojenie s existujúcimi medicínskymi informačnými systémami je náročná úloha aj pre odborníkov z oblasti informačných technológií, vzhľadom na rôznorodosť formátov, ktoré tieto systémy používajú. S plným prechodom na EHR by sa táto situácia mohla zlepšiť, nakoľko výmena informácií by mohla prebiehať pomocou EHR, do ktorých by oba systémy mali prístup [3].

*Použiteľnosť* programu je možné definovať ako metriku, ktorá skúma, ako jednoducho sa program používa. Skúma ako rýchlo a efektívne sa ho užívatelia naučia ovládať a či jeho používanie môže viesť ku chybám. Nedostatok použiteľnosti zabraňuje väčšiemu rozšíreniu CDSS v praxi. Mnohé akademické prototypy nemajú užívateľské rozhranie, prípadne je veľmi jednoduché a do produkčnej prevádzky sa nehodí [8].

Problémom býva aj získanie mohutnej a spoľahlivej základne znalostí, pretože dáta väčšinou pochádzajú od jedného pracoviska spojeného s akademickou inštitúciou a prístup k širšiemu okruhu dát je z dôvodu ochrany údajov obmedzený [3].

Prechod od akademického prototypu do komerčnej sféry je vzhľadom na vyššie uvedené časovo náročná a drahá záležitosť [3].

Tento prechod komplikujú aj ďalšie dôvody. Problémy s určením diagnózy nenastávajú v klinickej praxi tak často, takže sú CDSS zväčša používané len na kritické prípady a veľkým spoločnostiam sa viac oplatí investovať do iných komponentov medicínskych informačných systémov. Zároveň samostatným CDSS vyvíjaným inými spoločnosťami chýba efektívny spôsob, ako ich s veľkými medicínskymi informačnými systémami prepojiť. Lekári teda musia sami používať externú aplikáciu, čo narušuje ich pracovný postup. Používanie externej aplikácie zväčša zahŕňa aj nutnosť znovu zadávať informácie o pacientovi, čo môže byť zdĺhavé, frustrujúce a náchylné k chybám [3].

Pri tvorbe CDSS je teda nutné venovať zvýšenú pozornosť použiteľnosti. Existuje niekoľko štúdií, ktoré sa použiteľnosti CDSS priamo venujú.

Práca [9] poskytuje rady získané z experimentov vo Brigham and Women's Hospital, ktoré pomáhajú zlepšiť použiteľnosť CDSS.

Jedným z najdôležitejších faktorov je rýchlosť odozvy aplikácie. Lekári pociťujú diskomfort pokiaľ aplikácia na ich vstupy odpovedá pomaly, prípadne sú nútení pre uskutočnenie svojho cieľa prejsť cez množstvo obrazoviek. Práca

tvrdí, že lekári nemali záujem používať pomalý systém, aj keď informácie, ktoré poskytoval boli kvalitné a čitateľne prezentované.

Dôležité je tiež klásť dôraz na minimalitu a viditeľnosť stavu systému. Je vhodné zobrazovať informácie vtedy, keď sú aktuálne potrebné, nenútiť lekára si ich pamätať alebo ich hľadať na inom mieste v systéme. Zle dostupné informácie sú lekármi ignorované. Systém by tiež nemal nútiť vyplňať lekára údaje, ktoré nie sú potrebné. Naopak, tiež nie je vhodné označiť na začiatku procesu niektoré údaje ako voliteľné, aby potom lekár na konci procesu zistil, že ich nezadaním prišiel o dôležité znalosti a musel celý proces opakovať.

Zároveň je cenné poskytovať užívateľovi menšie množstvo relevantných informácií. Vo veľkom množstve obecných informácií je ťažké sa orientovať a podstatné informácie sa strácajú.

Práca tiež nabáda, aby autori CDSS udržiavali svoje systémy, monitorovali chovanie užívateľov, zisťovali, ktoré funkcionality sú využívané a ktoré ignorované a zakomponovali spätnú väzbu od lekárov. Pravidelné aktualizácie znalostí sú nemenej dôležité.

Práca [8] analyzuje 4 CDSS systémy používané v Canadian Emergency Department a popisuje ako problémy s použiteľnosťou vedú k chybám.

Poukazuje na to, že ak v znalostnej databáze chýba symptóm, lekári chýbajúci symptóm ignorujú alebo zadajú iný podobný symptóm, čo môže viesť k nepresným výsledkom.

Až 57 % detegovaných chýb má spojitost s navigáciou v aplikácii. Problémy sa týkajú informácií nachádzajúcich sa na neočakávaných miestach, zadávaní informácií cez množstvo obrazoviek alebo nemožnosti vrátiť sa k predchádzajúcemu kroku.

Problémom sú aj chýbajúce potvrdzovacie dialógy pri nevratných akciách či malé prehliadnutelné zaškrťavacie tlačítka, ktorých ignorovanie môže spôsobiť diametrálne odlišné chovanie systému.

## 1.2 Diagnostika porúch štítnej žľazy

Nasledujúca kapitola sa zaoberá diagnostikou porúch štítnej žľazy, pričom pojmy v nej definované tvoria teoretický základ algoritmov na diagnostiku porúch štítnej žľazy.

### 1.2.1 Základné pojmy

**Epidemiológia** *Epidemiológia* je vedná disciplína skúmajúca výskyt a rozloženie so zdravím súvisiacich javov (napr. ochorenia) v definovaných populáciách [10].

**Prevalencia** *Prevalencia* je miera výskytu určitého ochorenia (prípadne akéhokoľvek javu spojeného so zdravím) v určitej populácii v určitom čase.



Vypočíta sa ako pomer všetkých evidovaných osôb s ochorením k určenému dátumu zisťovania voči celkovej populácii v riziku ochorenia. Prevalencia sa udáva na 1 000, 10 000 alebo 100 000 ľudí v určitom čase a mieste [10].

**Krvný marker** *Krvný marker* je známka ochorenia alebo zdravotného stavu, ktorú je možné izolovať z krvného séra pacienta [11]. Môže sa jednať napríklad o koncentráciu určitého hormónu v krvi či prítomnosť špecifickej protilátky.

Ďalšie časti diplomovej práce sa na krvný marker odkazujú zjednodušeným názvom *marker*.

Pojem marker býva v medicíne použitý aj v iných kontextoch ako región DNA alebo nádorový marker. Týmito kontextami sa diplomová práca nezaobrá.

**Referenčný interval** Pojem *referenčný interval* sa používa pri laboratórnych vyšetreniach. Je tvorený takými hodnotami, medzi ktorými leží väčšina hodnôt získaná meraním referenčnej (zdravej) populácie. Hodnoty ležiace v referenčnom intervale sa označujú za normálne. Hodnoty ležiace mimo referenčný interval sa označujú za znížené alebo zvýšené [12].

**Cut-off hodnoty** *Cut-off hodnoty* sú rozhodovacie medzné hodnoty ležiace na kraji referenčného intervalu [12]. Napríklad výrazné prekročenie hornej cut-off hodnoty referenčného intervalu môže znamenať patologicky zvýšenú hodnotu laboratórneho merania signalizujúcu ochorenie.

### 1.2.2 Štítna žľaza

Štítna žľaza je životne dôležitý orgán zasahujúci do takmer všetkých metabolických dejov. Nachádza sa v podkoží na prednej strane krku. Je zložená z folikul vyplnených tekutinou, tzv. koloidom. Folikuly sú tvorené bunkami, ktoré produkujú hormóny štítnej žľazy tyroxín (TT4) a trijódyronín (TT3). Dôležitou zložkou hormónov štítnej žľazy je jód, ktorý je aktívne získavaný štítnou žľazou z krvi. TT4 obsahuje 4 atómy jódu a TT3 tri. Vyprodukované hormóny sú uskladnené v koloide v podobe molekuly tyreoglobulínu (TG) a v prípade potreby sú uvoľňované do krvi [13].

Väčšina hormónov prítomných v krvi (99 %) je viazaná na transportné bielkoviny, najčastejšie na tyroxín viažúci globulín (TBG). Viazané hormóny sú biologicky neaktívne. Nositeľom biologickej aktivity sú ich voľné frakcie fT3 a fT4 [14].

Regulácia funkcie štítnej žľazy je vykonávaná tyreoidálnym stimulačným hormónom (TSH), ktorý sa tvorí v adenohypofýze vďaka hormónu tyreoliberínu (TRH), ktorý je produkovaný hypotalamom. Princípom regulácie je negatívna spätná väzba. Pri poklese hormónov štítnej žľazy v krvi je zvýšená produkcia TSH [13].

Hlavné účinky hormónov štítnej žľazy sú dva [13]:

1. Obehové – hormóny štítnej žľazy zvyšujú rýchlosť srdečnej činnosti. Ich nadbytok aj nedostatok môže viesť k zlyhaniu srdca.
2. Metabolické – hormóny štítnej žľazy zvyšujú energetický výdaj organizmu a ovplyvňujú metabolizmus živín (cukrov, tukov a bielkovín). Nedostatok aj nadbytok hormónov vedie k poruchám metabolizmu napríklad zhoršenie cukrovky, zvýšenie cholesterolu či rednutie kostí.

### 1.2.3 Ochorenia štítnej žľazy [13]

Ochorením štítnej žľazy trpí 5 % až 10 % populácie. Ak je včas diagnostikované, je väčšina prípadov dobre liečiteľná, bez trvalých následkov.

Ochorenie štítnej žľazy sa môže prejavovať zníženou funkciou (hypotyreóza), zvýšenou funkciou (hypertyreóza) alebo pri niektorých ochoreniach ako sú uzly, nádory či zápal, môže funkcia ostať normálna (eutyreóza).

Ďalším častým príznakom ochorenia štítnej žľazy môže byť jej zväčšenie, tzv. struma. V niektorých prípadoch môže byť štítna žľaza aj zmenšená.

Typickými symptómami hypotyreózy sú napríklad: ospalosť, únava, suchá koža, zápcha či poruchy menštruačného cyklu.

Medzi symptómy hypertyreózy patria napríklad: potenie, nespavosť, únava, nervozita, búšenie srdca, svalová slabosť, váhový úbytok či poruchy menštruačného cyklu.

### 1.2.4 Diagnostické metódy

Prvotná diagnostika ochorenia štítnej žľazy sa vykonáva pomocou **analýzy symptómov** pacienta. Avšak príznaky typické pre ochorenia štítnej žľazy sa objavujú aj pri ochoreniach nesúvisiacich so štítnou žľazou a niektoré ochorenia prebiehajú celkom bez príznakov [13].

**Diagnostika uzlov** na štítnej žľaze sa vykonáva palpačným vyšetrením (prehmataním) oblasti krku alebo pomocou ultrazvuku [13].

Aj bezpríznakové ochorenia štítnej žľazy môžu spôsobiť výrazné zdravotné komplikácie, preto sú **laboratórne vyšetrenia** krvného séra pacienta veľmi dôležitou metódou diagnostiky porúch štítnej žľazy [13].

Pri laboratórnom vyšetrení sa stanovujú hodnoty želaných markerov vzhľadom na referenčné intervaly [14].

Základným tyreoidálnym laboratórnym vyšetrením je stanovenie hladiny TSH. Stanovenie hladiny TSH je v 99 % prípadov spoľahlivou metódou pre detekciu hypertyreóz a hypotereóz, vrátane latentných foriem.

Ak sa hodnoty TSH nachádzajú blízko hraníc referenčného intervalu, je možné použiť TRH stimulačný test podaním 200 mg TRH a sledovať odpoveď

TSH. Nesprávna odpoveď TSH môže indikovať ochorenie. TRH stimulačný test tiež slúži na odhalenie porúch hypofýzy a hypotalamu.

Časté je aj stanovovanie fT3 a fT4, celkové TT3 a TT4 sa kvôli svojej biologickej neaktivite stanovujú menej často.

Pre odlišenie pôvodu ochorenia sa stanovujú aj ďalšie hladiny markerov. Typicky sa jedná TG, ktorý je za normálnych podmienok syntetizovaný folikulami a jeho zvýšené sérové hladiny môžu indikovať ochorenie.

Ochorenia štítnej žľazy majú často autoimunitný charakter, preto je dôležitým laboratórnym vyšetrením aj stanovovanie protilátok. Protilátky proti TG (antiTG) a protilátky proti tyreoperoxidáze (antiTPO) bývajú napríklad prítomné u pacientov s tyreoiditídou Hashimotovho typu. Stanovujú sa tiež protilátky proti TSH receptoru (antiTSHR).

Pre odlišenie niektorých ochorení s podobným chovaním markerov sa tiež porovnávajú pomery markerov (tzv. indexy) voči sebe.

## 1.3 CDSS, ktoré pomáhajú s diagnostikou porúch štítnej žľazy

Existujúce CDSS, ktoré pomáhajú s diagnostikou porúch štítnej žľazy sa snažia čerpať znalosti z výsledkov všetkých troch diagnostických metód.

Práca [15] využíva metódy spracovania obrazu na automatickú detekciu uzlov na snímkach z ultrazvuku. Typické je použitie výsledkov laboratórných vyšetrení [16], symptómov pacienta [17] alebo ich kombinácie [18].

Spektrum používaných metód je široké, zahŕňajúce napríklad využitie pravidiel [17], fuzzy pravidiel [16],[19], bayesovských sietí [20],[21], rozhodovacích stromov [20], SVM [22], neurónových sietí [20],[21] a fuzzy neurónových sietí [23],[24].

Najlepšie výsledky na datasete získanom z UCI Machine Learning Repository, ktorý sa bežne používa na porovnanie jednotlivých metód, má práca z roku 2017 [24], využívajúca metódy informačného zisku na selekciu atribútov,  $k$ -nn klasifikátora a fuzzy neurónových sietí. Dosahuje presnosť 99,13 % (pomer správne klasifikovaných prípadov ku všetkým prípadom).

Výsledky ohľadom úspešnosti algoritmov sú sľubné a mohli by pomôcť zlepšiť diagnostický proces, avšak takmer žiadna z prác sa nezaobrá transformáciou metód do reálnej aplikácie, nenavrhuje možnú architektúru systému a nezaobrá sa použiteľnosťou. Diplomová práca v nasledujúcich podkapitolách analyzuje 3 systémy, ktorých návrhy sa aspoň sčasti zaoberajú týmito prvkami.

### 1.3.1 Systém s grafickým rozhraním v Matlabe [25]

Systém využíva 5 druhov neurónových sietí, ktorých znalosti vychádzajú z laboratórných vyšetrení 7 markerov a veku 244 pacientov so známymi diagnó-

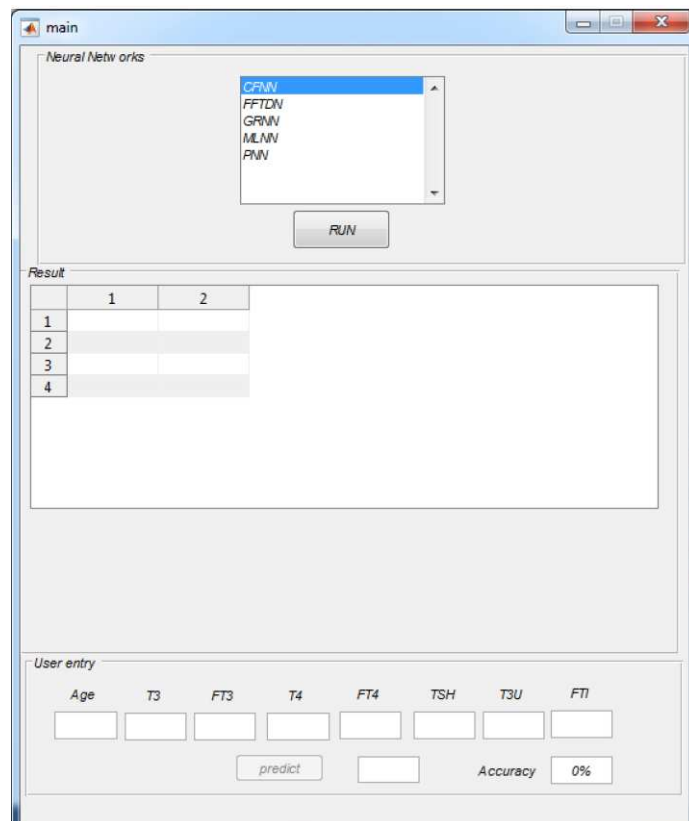
## 1. ANALÝZA DOMÉNY

---

zami patriacimi do 8 kategórií.

Pri testovaní na 40 nových pacientoch dosiahol 100 % úspešnosť zaradenia do kategórií.

Jedná sa o prototyp napísaný v jazyku Matlab a architektúra systému nie je diskutovaná. Systém však navyše k implementovaným metódam ponúka aj grafické užívateľské rozhranie (GUI) vhodné na testovanie systému.



Obr. 1.1: GUI systému v Matlabe, obrázok prevzatý z [25]

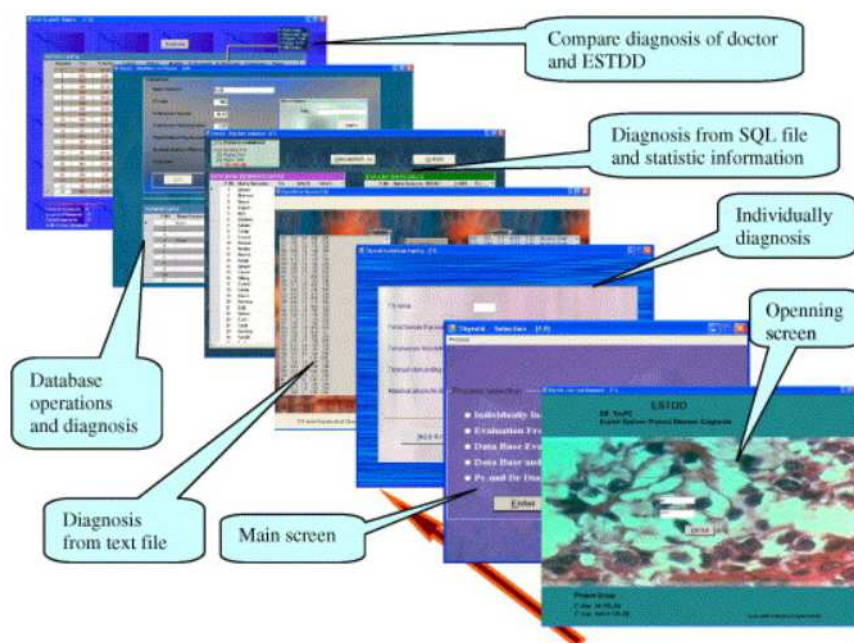
GUI pozostáva z jednej obrazovky, ktorá slúži na tréning neurónových sietí a zároveň na zadávanie nových vyšetrení pacientov.

Primárnym cieľom pri vytváraní GUI nebola použiteľnosť systému v praxi, ale zjednodušenie testovania metód odborníkmi. Pričom prototyp v tejto forme nie je nasaditeľný v praxi.

### 1.3.2 ESTDD

ESTDD [26] pracuje s laboratórnymi vyšetreniami pacienta a metódy, ktoré používa zahŕňajú množinu fuzzy pravidiel a neurónové siete. Na datasete získanom z UCI Machine Learning Repository dosahuje úspešnosti 95,33 %.

### 1.3. CDSS, ktoré pomáhajú s diagnostikou porúch štítnej žľazy



Obr. 1.2: GUI ESTDD, obrázok prevzatý z [26]

Architektonicky sa jedná o desktopovú aplikáciu napísanú v jazyku Visual C#, zloženú zo štyroch častí: GUI, lokálnej SQL databázy slúžiacej na ukládanie výsledkov vyšetrení, databázy znalostí (pravidiel) a výkonného programu obsahujúceho algoritmy, ktorý komunikuje so zvyšnými 3 časťami systému.

Užívateľovi umožňuje zadávať nové vyšetrenie pomocou rozhrania alebo pomocou importu zo súboru, a ukladať ich do lokálnej databázy. Rovnako mu umožňuje manipuláciu s predchádzajúcimi vyšetreniami, ich úpravu, mazanie či zopakovanie.

Oddelenie databázy znalostí od algoritmov je vhodným prístupom, ktorý umožňuje jednoduchú správu znalostí.

Aplikácia je, oproti prototypu popísanému v podkapitole 1.3.1, použiteľná v praxi, avšak obsahuje niekoľko nedostatkov.

GUI je v súčasnej dobe už zastarané a využíva netypické farebné kombinácie a fonty, aktualizácie lokálnej databázy znalostí alebo algoritmov vyžadujú inštaláciu novej verzie programu a lokálna SQL databáza neumožňuje zdieľať výsledky medzi rôznymi zariadeniami.

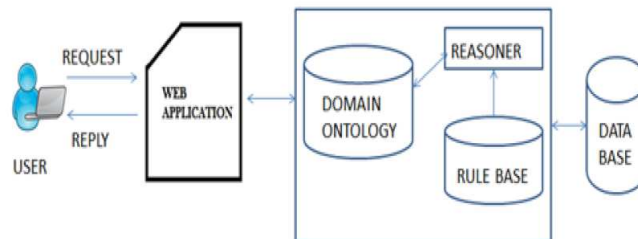
#### 1.3.3 OBESTDD

Systém OBESTDD [17] na diagnostiku používa symptómy pacienta, pričom je schopný vyhodnotiť či symptómy svedčia o hypotyreóze, hypertyreóze alebo

## 1. ANALÝZA DOMÉNY

---

normálnej funkcii štítnej žľazy. Jedná sa o klient-server aplikáciu (kompletná architektúra sa nachádza na obrázku 1.3).



Obr. 1.3: Architektúra OBESTDD, obrázok prevzatý z [17]

Name  
Rucha

Gender and Age

Female (01-12 years)

Female (13-19 years)

Female (more than 20 years)

Male (01-12 years)

Male (13-19 years)

Male (more than 20 years)

SELECT THYROID SYMPTOMS

Insomnia

Heat Intolerance and More Sweating

Sudden weight loss despite an increased appetite

Feeling panicked and nervous, Increased Heart Rate, Hand Tremor

Lighter Menstrual Periods/ Missed Periods

High Libido(high estrogen level)

Feeling Tired and More Sleepiness

Cold Intolerance and Less Sweating

Sudden weight gain despite low appetite

Low Heart Rate

More menstrual bleeding

Low libido (low estrogen level)

Submit Reset

Obr. 1.4: GUI OBESTDD, obrázok prevzatý z [17]

Webová aplikácia má model-view-controller (MVC) architektúru a využíva technológie JavaServer Pages.

Backend systému je zložený z:

- MySQL databázy určenej na ukladanie záznamov o pacientoch,
- základne znalosti v podobe ontológií vytvorenej editorom Protégé, pričom ontológie reprezentujú entity reálneho sveta ako osoba, diagnóza a symptómy,
- kolekcie pravidiel napísanej v jazyku Semantic Web Rule Language,
- odvodzovacieho systému Jena, ktorý používa ontológie a pravidlá na určenie diagnózy.

Všetky časti systému sú od seba oddelené, čo zjednodušuje údržbu a rozširovanie systému. Znalosti a pravidlá sú jednoducho nahraditeľné a doplniteľné a môžu byť poskytnuté aj iným systémom.

Klient-server architektúra aplikácie umožňuje širokú dostupnosť programu užívateľom, nakoľko podmienkou na jej používanie je mať webový prehliadač. Užívatelia majú k dispozícii vždy aktuálnu verziu, bez nutnosti manuálnych aktualizácií ako v prípade desktopových aplikácií.

Použitelnosť systému bohužiaľ nie je možné otestovať, vzhľadom na to, že aplikácia nie je verejne dostupná. Malá ukážka GUI sa nachádza na obrázku 1.4.

Úspešnosť aplikácie na testovacej vzorke 60 pacientov je 90 %.

Nevýhodou systému je stručnosť výstupu, ktorý ponúka. Jedná sa iba o strohú informáciu či pacient má hypertyreózu, hypotyreózu alebo je zdravý. Neponúka žiadne doplňujúce informácie ako príčiny (napr. aký typ ochorenia spôsobuje hypertyreózu) týchto stavov alebo rady ako ich liečiť. Systém lekárovi teda neponúka výraznú pridanú hodnotu.

## 1.4 Komerčne používané CDSS

Nasledujúca časť prináša analýzu 3 CDSS vo forme webových aplikácií, ktorým sa podarilo komerčne presadiť, pričom sa zameriava hlavne na použiteľnosť systémov, podporu EHR a integráciu s medicínskymi informačnými systémami.

### 1.4.1 Isabel

Systém Isabel je určený pre pediatrov na diagnostiku ochorení zo symptómov pacienta. Jeho vývoj začal v roku 1999 a prvá verzia bola dostupná v roku 2002 [27].

V súčasnosti je dostupná moderná webová aplikácia optimalizovaná aj pre zariadenia ako tablety či mobily.

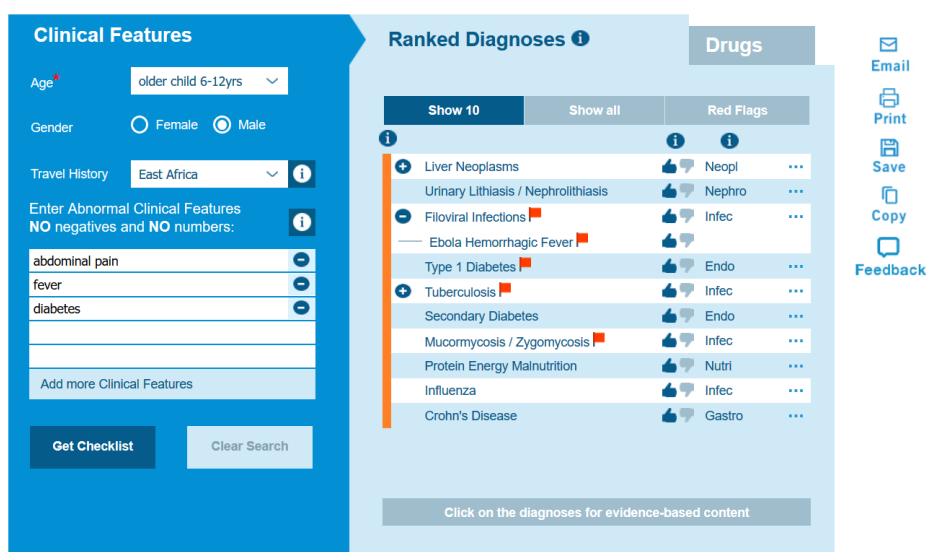
## 1. ANALÝZA DOMÉNY

Isabel vie pracovať s niekoľkými EHR formátmi [28], z ktorých automaticky extrahuje potrebné informácie o pacientovi bez nutnej interakcie lekára. Navyše, informačný systém Emergency Department Information System (EDIS) od spoločnosti T-System obsahuje modul, ktorý priamo komunikuje s Isabel a zobrazuje odporúčenia lekárovi bez nutnosti prepínania do externej aplikácie [28].

Použitelnosť aj úspešnosť systému bola validovaná niekoľkými štúdiami, pričom presnosť diagnostiky lekárov sa za použitia Isabel zvýšila o 33 % [29].

Samotná aplikácia v 96 % prípadoch zobrazí správnu diagnózu v zozname možných diagnóz [29].

Aplikácia má intuitívne grafické rozhranie (viď obrázok 1.5) pričom umožňuje lekárovi zadávať symptómy voľným textom a po zadaní troch znakov mu napovedá o aký symptóm by sa mohlo jednať. Aplikácia zobrazuje zoznam možných diagnóz a liekov, ktoré môžu spôsobovať zadané symptómy. Po kliknutí na diagnózu alebo liek sa zobrazí rozcestník (viď obrázok 1.6), ktorý umožňuje konzultáciu stavu s bežne používaným medicínskymi encyklopédiami alebo liekovými databázami.



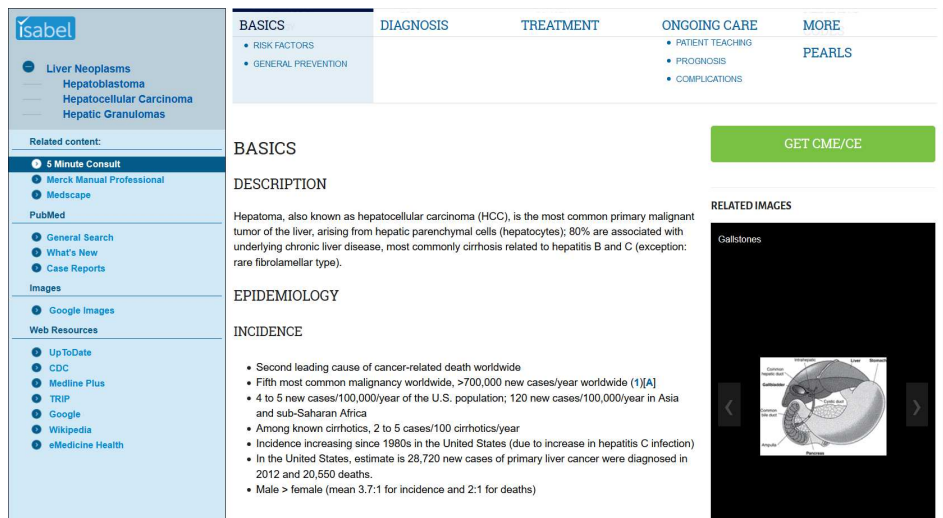
Obr. 1.5: GUI Isabel [30] – zadávanie symptómov

Odozva aplikácie je rýchla, v prípade krátkeho čakania algoritmus zobrazuje ukazateľ postupu. Rovnako je možné zadať spätnú väzbu pokiaľ sa špecialistovi používajúcemu systém nezdá výstup.

Aplikácia obsahuje aj niekoľko chýb z pohľadu používateľnosti. Štatistiky používania aplikácie sa nachádzajú v sekcii nastavenie, pričom so samotným nastavením aplikácie nemajú nič spoločné. Odkaz v menu pomenovaný „visit website“ vedie na adresu spôsobujúcu odhlásenie užívateľa zo systému a pre-



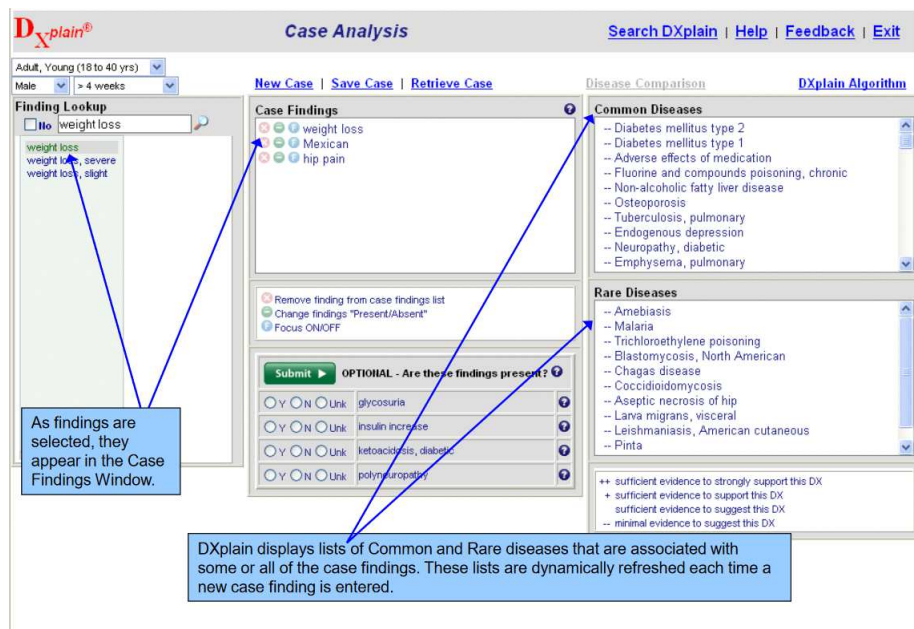
## 1.4. Komerčne používané CDSS



Obr. 1.6: GUI Isabel [30] – prepojenie s medicínskymi encyklopédiami

smerovanie na stránku popisujúcu Isabel. Pre návrat do systému je nutné sa znovu prihlásiť.

### 1.4.2 DXplain



Obr. 1.7: GUI DXplain, obrázok prevzatý z [31]

## 1. ANALÝZA DOMÉNY

---

DXplain je, rovnako ako Isabel, systém určený pre pediatrov na pomoc s diagnostikou ochorení zo symptómov pacienta. Proces používania oboch programov je rovnaký a tieto dva systémy sú si navzájom priami konkurenti.

Vývoj systému DxPlain začal v roku 1984, pričom prvá verzia bola dostupná v roku 1986 [32].

V súčasnosti je systém dostupný v podobe webovej aplikácie, ktorej rozhranie však nie je responzívne a použiteľnosť na zariadeniach ako mobily a tablety je obmedzená. Rozhranie (viď obrázok 1.7) je, aj napriek zastaranejšiemu vzhľadu, prehľadné a ovládacie prvky sú dobre viditeľné.

Systémy majú podľa porovnávajúcej štúdie [33] porovnateľnú úspešnosť, avšak veľkou nevýhodou systému DXplain je, že neposkytuje možnosť pracovať s obecnými dostupnými EHR a je prepojený len s informačným systémom Massachusetts General Hospital, ktorá ho vlastní.

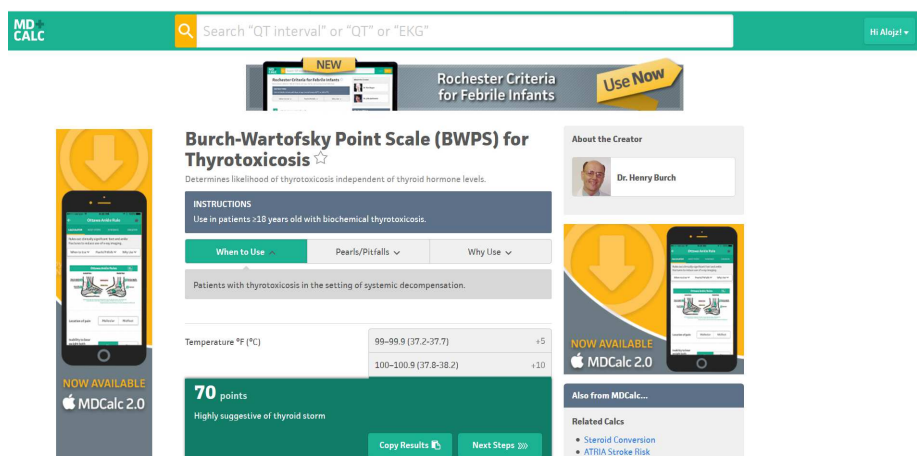
### 1.4.3 MDCalc

MDCalc je systém poskytujúci platformu pre diagnostické kalkulačky – rovnice, modely alebo testy, ktoré sú navrhnuté a validované špecialistami [34].

Autori platformy tvrdia, že systém je používaný v 196 krajinách viac ako miliónom užívateľov mesačne [34].

U každej kalkulačky systém uvádza autora a informácie o ňom, kedy a prečo je vhodné kalkulačku použiť, aké sú jej silné a slabé stránky, použitú metodológiu a publikované články.

Rozhranie (viď obrázok 1.8) každej kalkulačky obsahuje formulár na zadanie potrebných symptómov, box na zobrazenie výsledku a ďalšie kroky, ktoré je vhodné vykonať vzhľadom na výsledok.



Obr. 1.8: GUI MDCalc [35]

Samotná aplikácia má intuitívne rozhranie, pričom umožňuje vyhľadávanie jednotlivých kalkulačiek podľa názvu, filtrovanie podľa kategórií či uloženie

oblúbených.

Problémom z hľadiska použiteľnosti sú reklamy, ktoré sa zobrazujú na niekoľkých miestach na stránke a pôsobia rušivo. Ďalej nezobrazovanie, kde sa aktuálne v rámci aplikácie užívateľ nachádza. Posledným odhaleným problémom je menu, ktoré sa neposúva a nie je vždy viditeľné.

MDCalc je dostupný vo forme webovej aplikácie, aj aplikácií pre zariadenia s operačným systémom Android alebo iOS.

Aplikácia nie je integrovaná so žiadnymi informačnými systémami ani EHR a je voľne dostupná.



# Algoritmus na detekciu porúch štítnej žľazy

Kapitola na úvod vymedzuje pojmy potrebné pre popísanie a vysvetlenie algoritmu na detekciu porúch štítnej žľazy. Po vymedzení pojmov sa zaoberá popisom samotného algoritmu.

## 2.1 Potrebné pojmy

Nasledujúce podkapitoly obsahujú pojmy, ktoré sú použité na popis algoritmu, poskytujú preň teoretický základ a vďaka nim je možné ho pochopiť.

### 2.1.1 Pojmy z teórie pravdepodobnosti

Táto podkapitola definuje a vysvetľuje potrebné pojmy z teórie pravdepodobnosti.

**Náhodný pokus** *Náhodný pokus* je pokus, ktorého výsledok nie je jednoznačne určený podmienkami za ktorých prebieha, a zároveň je (teoreticky) opakovateľný za rovnakých podmienok [36],[37].

**Elementárny náhodný jav** Každý možný výsledok náhodného pokusu sa nazýva *elementárny jav*, značí sa  $\omega$ .

Pri vykonávaní pokusu žiadne dva elementárne javy nemôžu nastať zároveň a súčasne jeden z nich nastane vždy [37].

Elementárny jav sa dá chápať ako výsledok náhodného pokusu, ktorý už ďalej nejde rozložiť [36].

**Základný priestor elementárnych javov** Všetky elementárne javy tvoria *základný priestor elementárnych javov*, značí sa  $\Omega$  [37].

**Náhodný jav** *Náhodný jav*  $A$  je ľubovoľná podmnožina základného priestoru elementárnych javov  $\Omega$ ,  $A \subseteq \Omega$  [37].

**Pravdepodobnosti javov** Javy je možné hodnotiť podľa toho, akú veľkú majú *pravdepodobnosť*, že pri náhodnom pokuse nastanú. Pravdepodobnosť javu  $A \subseteq \Omega$  sa značí  $P(A)$  a platí pre ňu [37]:

$$P(A) \geq 0 \quad (2.1)$$

$$P(\Omega) = 1 \quad (2.2)$$

$$P(\bar{A}) = 1 - P(A) \quad (2.3)$$

Ak sú javy  $A_1, A_2, \dots, A_n$  navzájom nezlučiteľné potom [37]:

$$P(A_1, A_2, \dots, A_n) = \sum_{i=1}^n P(A_i) \quad (2.4)$$

*Podmienená pravdepodobnosť* javu  $A$  za predpokladu, že nastal jav  $B$  je daná vzťahom [37]:

$$P(A|B) = \frac{P(A \cap B)}{P(B)}, \quad P(B) > 0 \quad (2.5)$$

Zo vzťahu (2.5) je možné vyjadriť pravdepodobnosť, že nastal jav  $A$  a zároveň nastal jav  $B$  ako:

$$P(A \cap B) = P(A|B)P(B), \quad P(B) > 0 \quad (2.6)$$

Pomocou *Bayesovej vety* je možné opačnú podmienenú pravdepodobnosť  $P(B|A)$  vyjadriť ako [36]:

$$P(B|A) = \frac{P(A|B)P(B)}{P(A)}, \quad P(A) > 0 \quad (2.7)$$

Pre ľubovoľný počet javov  $A_1, A_2, \dots, A_n$  platí *veta o násobení pravdepodobnosti* [37]:

$$P\left(\bigcap_{i=1}^n A_i\right) = P(A_1)P(A_2|A_1) \cdots P(A_n|P(A_1 \cap A_2 \cap \dots, \cap A_{n-1})) \quad (2.8)$$

$$\text{za predpokladu, že } P\left(\bigcap_{i=1}^k A_i\right) > 0, \quad k = 1, 2, \dots, n-1$$

Javy  $A$  a  $B$  sú *nezávislé* ak platí [36]:

$$P(A \cap B) = P(A|B)P(B) = P(B|A)P(A) = P(A)P(B) \quad (2.9)$$

Nezávislosť javov  $A$  a  $B$  znamená, že fakt, že nastal jav  $B$  nijakým spôsobom neovplyvňuje pravdepodobnosť, že nastane jav  $A$  a naopak.

Zo vzťahov (2.8) a (2.9) je možné odvodiť, že pre nezávislé javy  $A_1, \dots, A_n$  platí:

$$P\left(\bigcap_{i=1}^n A_i\right) = \prod_{i=1}^n P(A_i) \quad (2.10)$$

Nezávislé javy  $A$  a  $B$  sa môžu stať *podmienene závislé*, v prípade, že nastal nejaký ďalší jav  $C$ . Rovnako, závislé javy sa za prítomnosti iného javu môžu stať *podmienene nezávislé*.

Javy  $A$  a  $B$  sú podmienene nezávislé, podmienené javom  $C$ , ak platí [38]:

$$P(A \cap B|C) = P(A|C)P(B|C) \text{ alebo } P(C) = 0 \quad (2.11)$$

Ekvivalentne je možné podmienenú nezávislosť javov  $A$  a  $B$  podmienenú javom  $C$  vyjadriť ako [38]:

$$P(A|B \cap C) = P(A|C) \text{ alebo } P(B \cap C) = 0 \quad (2.12)$$

**Náhodná veličina** *Náhodná veličina*  $X(\omega)$  je reálna funkcia definovaná na priestore elementárnych javov  $\Omega$ , ktorá každému javu priraduje práve jedno reálne číslo  $X(\omega) = x$ ,  $x \in \mathbb{R}$  [37].

Čísla priradené elementárnym javom tvoria obor hodnôt  $M$  náhodnej veličiny,  $M = \{x = X(\omega) : \omega \in \Omega\}$  [37].

Pojem náhodná veličina sa dá chápať ako číselné ohodnotenie výsledku náhodného pokusu.

**Diskrétna náhodná veličina** *Diskrétna náhodná veličina* je náhodná veličina, ktorej obor hodnôt  $M$  je konečná alebo spočetná množina a platí [36]:

$$\sum_{\forall x \in M} P(X = x) = 1 \quad (2.13)$$

Funkcia  $P(X = x)$  sa nazýva pravdepodobnostná funkcia diskkrétnej náhodnej veličiny  $X$ , skrátene sa zapisuje  $P(x)$  [36].

**Náhodný vektor** *Náhodný vektor* (prípadne  $n$ -rozmerná náhodná veličina)  $X = (X_1, \dots, X_n)$  je usporiadaná  $n$ -tica náhodných veličín  $X_1, \dots, X_n$  [36].

Náhodný vektor sa používa v prípade, keď je nutné pri pokuse sledovať viac výsledkov. Napríklad pri analyzovaní výsledkov laboratórneho testu je nutné sledovať hodnoty koncentrácie viacerých látok.

**Náhodný vektor s rozdelením diskrétného typu** Náhodný vektor má *rozdelenie diskrétného typu* pokiaľ existuje konečná alebo spočetná množina  $n$ -tíc reálnych čísel  $x_1, \dots, x_n$ , každá tvoriaca obor hodnôt jednej náhodnej veličiny a platí [36]:

$$\sum_{\forall x_1} \cdots \sum_{\forall x_n} P(X_1 = x_1, \dots, X_n = x_n) = 1 \quad (2.14)$$

Funkcia  $P(X_1 = x_1, \dots, X_n = x_n)$  sa nazýva *združená pravdepodobnostná funkcia náhodných veličín*  $X_1, \dots, X_n$ . Skráteno sa zapisuje  $P(x_1, \dots, x_n)$  [36].

Pravdepodobnosti javov týkajúce sa jednej náhodnej veličiny je možné vypočítať pomocou *marginálnej pravdepodobnostnej funkcie* [36]:

$$P(x_1) = \sum_{\forall x_2} \cdots \sum_{\forall x_n} P(x_1, x_2, \dots, x_n) \quad (2.15)$$

Pre náhodný vektor  $(X_1, \dots, X_n)$ , ktorý sa riadi rozdelením diskrétného typu platí, že náhodné veličiny  $X_1, \dots, X_n$  sú nezávislé práve vtedy, keď:

$$P(x_1, \dots, x_n) = \prod_{\forall x_1 \cdots \forall x_n} P(x_1) \cdots P(x_n) \quad (2.16)$$

Vzťah (2.16) vychádza zo vzťahu pre nezávislosť javov (2.9). Analogicky je možné vyjadriť aj ďalšie vzorce.

Napríklad pre náhodný vektor dvoch náhodných veličín  $(X_1, X_2)$  s rozdelením diskrétného typu, je možné pravdepodobnostnú funkciu náhodnej veličiny  $X_1$  za podmienky, že veličina  $X_2$  získala hodnotu  $x_2$ , vyjadriť ako (viď (2.5)):

$$P(x_1|x_2) = \frac{P(x_1, x_2)}{P(x_2)}, \quad P(x_2) > 0 \quad (2.17)$$

### 2.1.2 Fuzzy množiny

Táto podkapitola definuje a vysvetľuje potrebné pojmy z teórie fuzzy množín.

**Prvok** *Prvok* je jeden z objektov, s ktorými je pracované. Môže sa jednať o číselnú hodnotu, ale aj o akýkoľvek reálny objekt [39]. Prvky sa musia dať od seba odlíšiť.

**Univerzum** *Univerzum* tvoria všetky prvky, s ktorými je pracované, pričom v jeden okamih môže byť súbežne definované aj viacero univerz [39].



**Množina** *Množina* je súbor prvkov z univerza.

Bud univerzum  $U$  obsahujúce prvky *jablko, kiwi, mango, zemiak a mrkva*. Množinu  $M$  na univerze  $U$ , ktorá obsahuje iba ovocie je možné zapísať vymenovaním prvkov nasledovne:

$$M = \{jablko, kiwi, mango\}$$

Samotné univerzum je tiež množinou:

$$U = \{jablko, kiwi, mango, zemiak, mrkva\}$$

Množina  $M$  je podmnožinou  $U$ ,  $M \subset U$ .

Alternatívne je množinu  $M$  možné popísať s využitím *stupňa príslušnosti k množine*, značí sa  $\mu$  [39].

$\mu$  môže pre klasické množiny nadobúdať hodnôt 0 – pre prvok, ktorý do množiny nepatrí a 1 – pre prvok, ktorý do množiny patrí.

Množina  $M$  zapísaná pomocou stupňa príslušnosti vyzerá nasledovne:

$$M = \{1/jablko, 1/kiwi, 1/mango, 0/zemiak, 0/mrkva\}$$

**Fuzzy množina** Existujú situácie, kedy nestačí jednoznačné zaradenie patrí alebo nepatrí do množiny. Platí to napríklad pre univerzum  $U$  obsahujúce študentov Fakulty informačných technológií, kde je úlohou rozdeliť študentov do dvoch množín na dobrých a zlých. V tomto prípade je náročné určiť hranicu, kto je ešte dobrý študent a kto už nie.

*Fuzzy množina* je rovnako ako klasická množina tvorená prvkami  $x$  univerza. Avšak v prípade fuzzy množín, je stupeň príslušnosti k množine  $\mu$  definovaný na intervale  $\langle 0,1 \rangle$ , kde hodnota 1 značí, že prvok  $x$  do množiny plne náleží a 0 značí, že prvok do množiny nenáleží vôbec [39].

Fuzzy množiny  $Z$  pre zlých študentov a  $D$  pre dobrých študentov môžu vyzeráť nasledovne:

$$\begin{aligned} Z &= \{0,95/Peter, 0,75/Martin, 0,74/Jitka, 0,1/Marek, \dots\} \\ D &= \{0,05/Peter, 0,25/Martin, 0,26/Jitka, 0,9/Marek, \dots\} \end{aligned}$$

Teória fuzzy množín je zobecnením klasickej teórie množín. Klasická množina so stupňami príslušnosti 0 a 1 je špeciálnym prípadom fuzzy množiny, takáto množina sa nazýva aj *ostrá množina*.

**Funkcia príslušnosti** *Funkcia príslušnosti*  $\mu_A(x)$  fuzzy množiny  $A$  definovanej na univerze  $U$  je funkcia, ktorá prvkom  $x \in U$  priraduje ich stupeň príslušnosti k množine  $A$  [39].

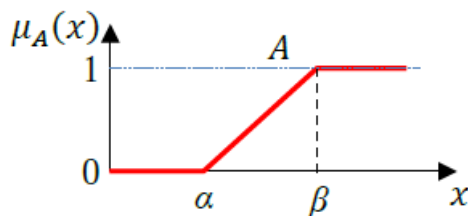
Funkcia príslušnosti býva zväčša popísaná rovnicou alebo sadou rovníc, v prípade konečného univerza môže byť popísaná aj tabuľkou [39].

Pre ilustráciu je možné si predstaviť fuzzy množinu  $A$ , do ktorej patria vysokí muži. Jej funkcia príslušnosti  $\mu_A(x)$  je zapísaná nasledovne:

$$\mu_A(x) = \begin{cases} 0 & x < \alpha \\ \frac{x-\alpha}{\beta-\alpha} & x \in \langle \alpha, \beta \rangle \\ 1 & x > \beta \end{cases} \quad (2.18)$$

kde  $\alpha = 170$  cm a  $\beta = 190$  cm.

Muži s výškou nad 190 cm do množiny  $A$  plne náležia, naopak muži s výškou pod 170 cm do nej nenáležia vôbec. Stupeň príslušnosti pre mužov, ktorí merajú viac ako 170 cm, ale menej ako 190 cm, je možné vypočítať pomocou funkcie  $\frac{x-170}{20}$ . Funkcia príslušnosti  $\mu_A(x)$  má tvar znázornený na obrázku 2.1.



Obr. 2.1: Grafické znázornenie funkcie príslušnosti (2.18), obrázok prevzatý z [39]

### 2.1.3 Bayesovská sieť

*Bayesovská sieť* slúži na reprezentáciu vzťahov (podmienených závislostí a nezávislostí) medzi náhodnými veličinami a ponúka kompaktný spôsob pre výpočet ich združenej pravdepodobnosti [40].

Bayesovská sieť pre vektor náhodných veličín  $X = (X_1, \dots, X_n)$  je dvojica  $(G, \Theta)$ , kde [38],[41]:

- $G = (V, E)$  – je orientovaný acyklický graf, ktorého uzly  $V = \{1, \dots, n\}$  reprezentujú náhodné veličiny  $X_v$ , a ktorého hrany  $E$  reprezentujú závislosti medzi náhodnými veličinami. Graf  $G$  sa nazýva štruktúra siete.
- $\Theta$  – je množina tabuliek podmienených pravdepodobností, jedna pre každú náhodnú veličinu  $X_v \in X$ .  $\Theta$  sa nazýva parametrizácia siete.

Zároveň musí platiť, že združená pravdepodobnostná funkcia celej siete  $P(x)$  sa dá vyjadriť ako:

$$P(x) = \prod_{v=1}^n P(x_v | x_{pa(v)}) \quad (2.19)$$

kde  $pa(v)$  je množina rodičov uzla  $v$ .

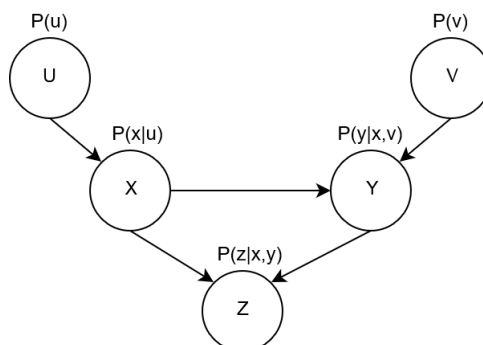
Pre lepšie pochopenie definície bayesovskej siete je nutné doplniť definíciu ešte pre niekoľko pojmov:

- rodičia uzla  $v$  – uzly, z ktorých priamo vedie hrana do  $v$
- predkovia uzla  $v$  – uzly, z ktorých je možné sa dostať nasledovaním hrán do uzla  $v$
- potomkovia uzla  $v$  – uzly, do ktorých je možné sa dostať nasledovaním hrán z uzla  $v$

Definícia bayesovskej siete pomocou vzťahu (2.19) sa nazýva definícia pomocou *faktorizácie* a vychádza z vety o násobení pravdepodobnosti (2.8), pričom sa v úvahu berie to, že uzol  $v$  je podmienene nezávislý (viď vzťah (2.12)) na uzloch, ktoré nie sú jeho potomkami, podmienené jeho rodičmi [41].

V praxi to znamená, že napríklad pre bayesovskú sieť zobrazenú na obrázku 2.2 je možné združenú pravdepodobnostnú funkciu vyjadriť ako:

$$P(u,v,x,y,z) = P(u)P(v)P(x|u)P(y|x,v)P(z|x,y)$$



Obr. 2.2: Príklad bayesovskej siete pre náhodný vektor  $(U,V,X,Y,Z)$

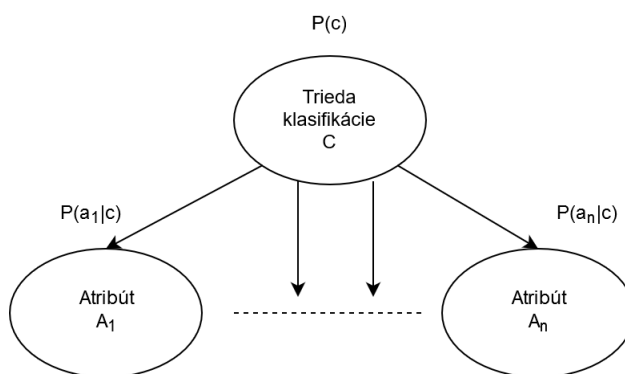
#### 2.1.4 Naivný bayesovský klasifikátor

*Naivný bayesovský klasifikátor* je jednoduchým typom bayesovskej siete použitej na *klasifikačný problém*.

Klasifikačný problém je problém, pri ktorom je nutné zaradiť nové pozorovanie do vopred definovaných kategórií, popísaných náhodnou veličinou  $C$ . Proces klasifikácie využíva znalosti získané z množiny pozorovaní, ktoré majú priradenú kategóriu  $c$ . Pozorovanie je inštanciou náhodného vektora atribútov  $A = (A_1, \dots, A_n)$ .

Naivný bayesovský klasifikátor sa naučí zo známych pozorovaní podmienené pravdepodobnosti  $P(a_i|c), \forall i$ .

Tieto pravdepodobnosti sú použité na zaradenie nového pozorovania do triedy. Slúžia na výpočet pravdepodobností  $P(c|a_1 \dots a_n)$ , tj. aká je pravdepodobnosť, že pozorovanie patrí do triedy  $c$ , ak je tvorené hodnotami atribútov  $(a_1, \dots, a_n)$ . Trieda  $c$  s maximálnou  $P(c|a_1 \dots a_n)$  je zvolená [42]. Grafická reprezentácia bayesovskej siete tvoriacej naivný bayesovský klasifikátor sa nachádza na obrázku 2.3.



Obr. 2.3: Naivná bayesovská sieť

Pravdepodobnosť  $P(c|a_1 \dots a_n)$  je za použitia Bayesovej vety (2.7) možné vyjadriť ako [43]:

$$P(c|a_1, \dots, a_n) = \frac{P(a_1, \dots, a_n|c)P(c)}{P(a_1, \dots, a_n)} \quad (2.20)$$

Naivný bayesovský klasifikátor predpokladá, že náhodné veličiny reprezentujúce jednotlivé atribúty sú podmienene nezávislé, podmienené náhodnou veličinou reprezentujúcou triedu klasifikácie  $C$ .  $P(c|a_1, \dots, a_n)$  je teda možné vyjadriť ako [43]:

$$P(c|a_1, \dots, a_n) = \frac{P(c) \prod_{i=1}^n P(a_i|c)}{P(a_1, \dots, a_n)} \quad (2.21)$$

V praxi sa často  $P(a_1, \dots, a_n)$  zanedbáva a na záver sa pravdepodobnosti  $P(c|a_1, \dots, a_n)$  normalizujú, aby  $\sum_{\forall c} P(c|a_1, \dots, a_n) = 1$ .  $P(c|a_1, \dots, a_n)$  je teda možné zjednodušiť ako:

$$P(c|a_1, \dots, a_n) = P(c) \prod_{i=1}^n P(a_i|c) \quad (2.22)$$

### 2.1.5 Fuzzy naivný bayesovský klasifikátor

Táto podkapitola porušuje definíciu náhodnej veličiny danú v podkapitole 2.1.1, a z dôvodu lepšej čitateľnosti nasledujúceho výkladu, rozširuje obor hodnôt  $M$  náhodnej veličiny aj na reťazce.

*Fuzzy naivný bayesovský klasifikátor* využíva toho, že častokrát hodnota atribútu nie je úplne istá [42]. Napríklad atribút reprezentovaný diskretnou náhodnou veličinou  $X$  určujúci farbu objektu,  $M_x = \{\text{modrá, zelená, červená}\}$ . Objekt azúrovej farby môže byť vnímaný ako modrý aj zelený, niekedy viac modrý, niekedy viac zelený, v závislosti na jeho odtieni (viď obrázok 2.4).



Obr. 2.4: Rôzne odtiene azúrovej

S určitým stupňom príslušnosti patrí do fuzzy množiny *modrá* obsahujúcej modré predmety, aj do fuzzy množiny *zelená* obsahujúcej zelené predmety.

Nech

- $C$  – náhodná veličina predstavujúca triedu klasifikácie nadobúdajúca hodnôt  $C = c, \forall c$
- $A$  –  $n$ -rozmerný náhodný vektor všetkých atribútov  $A = (A_1, \dots, A_n)$ , ktoré nadobúdajú hodnôt  $A_1 = a_1, \dots, A_n = a_n, \forall a_1, \dots, \forall a_n$
- hodnoty  $a_i$  atribútu  $A_i$  tvoria fuzzy množiny
- $\underline{a_{ik}}$  – fuzzy atribút, ktorý môže patriť do fuzzy množín  $a_i$
- $\mu_{a_i}(\underline{a_{ik}})$  – funkcia príslušnosti, určujúca stupeň príslušnosti  $\underline{a_{ik}}$  do  $a_i$

potom sa pravdepodobnosť  $P(\underline{a_{ik}}|c)$  vypočíta ako [43]:

$$P(\underline{a_{ik}}|c) = \sum_{\forall a_i} P(a_i|c) \mu_{a_i}(\underline{a_{ik}}) \quad (2.23)$$

Nech  $\underline{A}$  je množina všetkých fuzzy atribútov  $\underline{A} = \{\underline{a_{ik}}\}$ , potom sa pravdepodobnosť  $P(c|\underline{A})$  vypočíta analogicky ako v prípade klasického naivného bayesovského klasifikátora (2.21) [43]:

$$P(c|\underline{A}) = P(c) \prod_{\forall \underline{a_{ik}}} \frac{P(\underline{a_{ik}}|c)}{P(\underline{a_{ik}})} \quad (2.24)$$

Menovateľ je opäť možné zanedbať (viď vzťah (2.22)), ak sa na záver pravdepodobnosti  $P(c|\underline{A})$  normalizujú tak, aby  $\sum_{\forall c} P(c|\underline{A}) = 1$ .  $P(c|\underline{A})$  je teda možné vypočítať ako [43]:

$$P(c|\underline{A}) = P(c) \prod_{\forall a_{ik}} P(a_{ik}|c) \quad (2.25)$$

## 2.2 Algoritmus

Algoritmus na detekciu porúch štítnej žľazy na svoju prácu využíva vek, pohlavie a výsledky laboratórných vyšetrení pacienta. Jeho hlavnou úlohou je pomoc s primárnou diagnostikou ochorení. Výstupom sú pravdepodobnosti možných diagnóz a stavov pacienta. Rovnako je ho možné použiť na monitorovanie efektov terapie u niektorých typov ochorení štítnej žľazy [44].

Táto podkapitola sa zaoberá podrobným popisom fungovania algoritmu. Popis vychádza najmä zo skúmania historickej implementácie algoritmu v jazyku Pascal (viď kapitola 3) a následnej konzultácií nejasností s autorom algoritmu doc. RNDr. Ladislavom Pecnom, CSc., nakoľko publikovaná literatúra sa podrobnému popisu algoritmu nevenuje a ponúka len strohé informácie ako „*algoritmus je založený na teórii bayesovských sietí* [45, vlastný preklad]“ alebo „*na základe koncentrácií hormónov a protilátok proti nim, epidemiologickej incidencie jednotlivých ochorení (v závislosti na veku a pohlaví) je pomocou fuzzy pravdepodobnostnej siete určená pravdepodobnosť jednotlivých ochorení* [44, vlastný preklad]“.

Algoritmus má dva typy funkcionality:

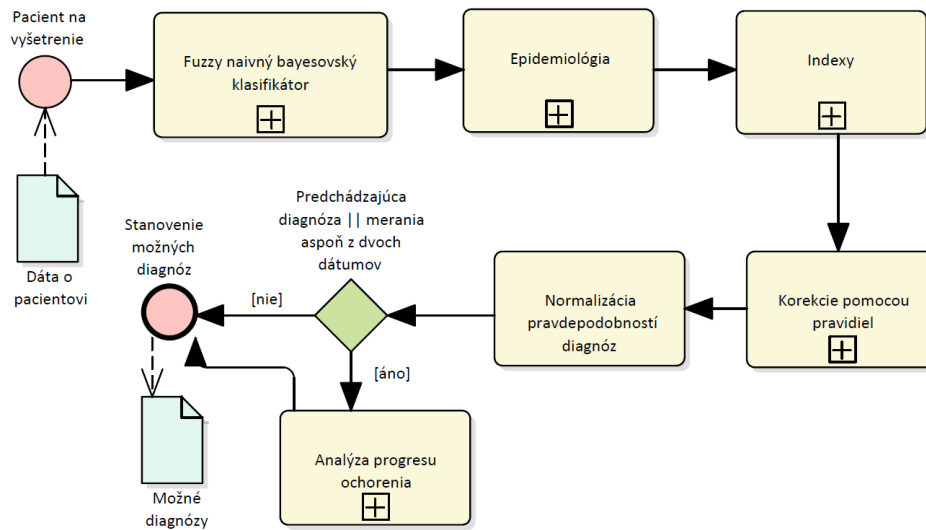
1. Pomoc s primárnou diagnostikou ochorení štítnej žľazy.
2. Zhodnotenie progresu liečby skôr diagnostikovaného ochorenia.

Základom **primárnej diagnostiky ochorení** je fuzzy naivný bayesovský klasifikátor, ktorý podľa laboratórných výsledkov pacienta určí základné pravdepodobnosti diagnóz (viac v časti 2.2.1). Tieto pravdepodobnosti sú následne upravené pomocou epidemiologickej prevalencie jednotlivých ochorení pre daný vek a pohlavie pacienta (viac v časti 2.2.2).

Potom sú vyčíslené indexy, pomery niektorých markerov voči sebe, ktoré sú použité na odlíšenie diagnóz s podobným chovaním markerov (viac v časti 2.2.3).

Na záver sú aplikované expertné znalosti o chovaní markerov pri jednotlivých diagnózach (napríklad, ktoré merania sú pre danú diagnózu relevantnejšie ako iné), ktoré ďalej upravujú pravdepodobnosti diagnóz (viac v časti 2.2.4).

Posledným krokom je normalizácia pravdepodobností, aby ich súčet bol rovný 1.



Obr. 2.5: Diagram, zobrazujúci jednotlivé fázy algoritmu

**Zhodnotenie progresu liečby** sa vykonáva ako doplnok primárnej diagnostiky v prípade, že bola algoritmu poskytnutá predchádzajúca diagnóza alebo merania z minimálne dvoch rozličných dátumov. Algoritmus skúma ako sa v čase menili koncentrácie markerov a aplikovaním expertných znalostí vyhodnotí progres liečby ochorenia (viac v kapitole 2.2.5).

### 2.2.1 Základný model výpočtu pomocou fuzzy naivného bayesovského klasifikátora

Základom výpočtu pravdepodobností jednotlivých diagnóz je fuzzy naivný klasifikátor (obecne popísaný v podkapitole 2.1.5).

Z pozorovaní atribútov týkajúcich sa koncentrácie markerov (hormónov, proteínov a protilátok) z krvného séra pacienta sa snaží určiť, ktorá diagnóza je pre toto chovanie najpravdepodobnejšia (zoznam vstupných markerov a možných výstupných diagnóz sa nachádza v prílohe B).

Algoritmus je schopný pracovať s viacerými meraniami jedného markeru vykonanými v rôznych dátumoch a rovnako je schopný určiť pravdepodobnosti diagnóz aj keď niektoré merania neboli vykonané.

Algoritmus pracuje s rozšírenou definíciou referenčného intervalu (viď podkapitola 1.2.1), kde okrem stanovovania či je hodnota merania normálna, zvýšená alebo znížená pridáva dve ďalšie cut-off hodnoty. Interval pre zvýšené hodnoty rozdeľuje na zvýšené a veľmi zvýšené a interval pre znížené hodnoty rozdeľuje na znížené a veľmi znížené. Toto doplňujúce rozdelenie je vytvorené z dôvodu, aby sa dali lepšie odlíšiť patologické koncentrácie.

Koncentrácia merania markeru teda môže byť veľmi znížená, znížená, normálna, zvýšená alebo veľmi zvýšená. Niekedy sa však môže nachádzať v okolí hranice (cut-off hodnoty) rozdeľujúcej dva disjunktné intervaly.

V takom prípade je nutné vziať do úvahy prípadnú chybu merania a uvažovať oba možné intervaly. Z toho dôvodu hodnoty jednotlivých meraní predstavujú fuzzy atribúty modelu.

Preto, že sa jedná o naivný bayesovský klasifikátor, sú jednotlivé atribúty chápané ako podmienene nezávislé vzhľadom na diagnózu. V realite toto zjednodušenie nemusí platiť a napríklad zníženie hladiny jedného markeru môže spôsobovať zvýšenie iného.

Táto podkapitola tiež pracuje s rozšíreným oborom hodnôt pre náhodné veličiny o reťazce, ako bolo dané v podkapitole 2.1.5.

Nech

- *Diag* – množina všetkých diagnóz, ktoré algoritmus berie do úvahy.
- *D* – diskrétna náhodná veličina reprezentujúca diagnózy (triedy klasifikácie), ktorá každej diagnóze  $d \in \text{Diag}$  priradí číslo.  $M_D = \{1, 2, \dots, n\}$
- *Mark* – množina všetkých typov markerov, ktoré algoritmus berie do úvahy.
- $V_m$  – diskrétna náhodná veličina popisujúca javy: hodnota merania markeru  $m \in \text{Mark}$  je veľmi znížená, znížená, normálna, zvýšená alebo veľmi zvýšená. Obor hodnôt  $M_V$  je definovaný nasledovne:

$$\begin{aligned} V_m = vl & \quad \text{veľmi znížená,} \\ V_m = l & \quad \text{znížená,} \\ V_m = n & \quad \text{normálna,} \\ V_m = h & \quad \text{zvýšená,} \\ V_m = vh & \quad \text{veľmi zvýšená.} \end{aligned}$$

- Obor hodnôt  $M_V = \{vl, l, n, h, vh\}$  je pre všetky náhodné veličiny  $V_m$  rovnaký a zároveň jeho hodnoty tvoria fuzzy množiny.
- $\underline{m}_{(i)f}$  – fuzzy atribút pre  $i$ -té meranie markeru  $m$ . Jeho hodnoty s určitým stupňom príslušnosti patria do fuzzy množín  $vl, l, n, h, vh$ .
- $\mu_{vl}(\underline{m}_{(i)f}), \mu_l(\underline{m}_{(i)f}), \mu_n(\underline{m}_{(i)f}), \mu_h(\underline{m}_{(i)f}), \mu_{vh}(\underline{m}_{(i)f})$  – funkcie príslušnosti určujúce stupeň príslušnosti  $\underline{m}_{(i)f}$  do fuzzy množín  $vl, l, n, h, vh$ .
- $M_f$  – náhodný vektor všetkých podmienene nezávislých fuzzy atribútov  $\underline{m}_{(i)f}$  (podmienených diagnózou  $d$ ), ktoré nastali.



Potom

$$P(\underline{m}_{(i)f}|D = d) = \sum_{\forall x \in M_V} P(V_m = x|D = d)\mu_x(\underline{m}_{(i)f}) \quad (2.26)$$

$$P(D = d|M_f) = P(D = d) \prod_{\forall \underline{m}_{(i)f}} P(\underline{m}_{(i)f}|D = d) \quad (2.27)$$

Základný model tiež predpokladá, že pravdepodobnosť, že nastane nejaká diagnóza bez informácií z laboratórnych vyšetrení  $P(D = d)$  je rovnaká pre všetky diagnózy. Je teda možné ju v rovnici zanedbať. Funkciu  $P(D = d)$  v ďalšej fáze algoritmu nahradzujú hodnoty z epidemiológie.

Používaním algoritmu v praxi sa zistilo, že protilátky nemajú až takú váhu ako ostatné parametre pri určovaní diagnózy, nakoľko môžu reflektovať dávnu minulosť, ktorá pre vyšetrenie už nie je relevantná. Preto sa za účelom vylepšenia modelu ich vplyv oslabuje použitím logaritmu.

Normalizácia hodnôt  $P(D = d|M_f)$  sa vykonáva až po aplikovaní všetkých fáz algoritmu.

Výsledná rovnica zahŕňajúca zanedbanie  $P(D = d)$  a vylepšenie pre protilátky je nasledovná:

$$P(D = d|M_f) = \prod_{\forall \underline{m}_{(i)f} \notin pr} P(\underline{m}_{(i)f}|D = d) \prod_{\forall \underline{m}_{(i)f} \in pr} \ln(1 + 20P(\underline{m}_{(i)f}|D = d)), \quad (2.28)$$

kde  $pr$  značí protilátky.

Logaritmus je posunutý o 1, aby výstupom bola kladná hodnota a pravdepodobnosť sa násobí konštantou 20, aby oslabenie vplyvu nebolo až také výrazné. Táto konštanta bola autorom algoritmu zvolená po skúšaní rôznych variánt a validovaní modelu na reálnych dátach.

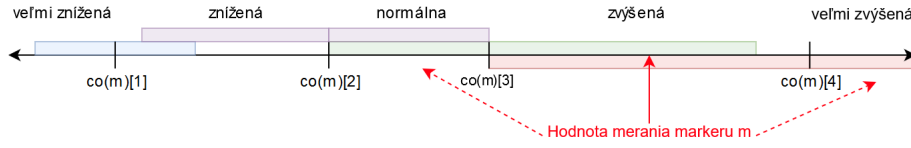
Pravdepodobnosti  $P(V_m = x|D = d)$  pre  $\forall x \in M_V$  a  $\forall d \in M_D$  sú vypočítané z predchádzajúcich pozorovaní chovania markerov pre jednotlivé diagnózy. Pre tieto účely boli analyzované dáta pacientov z 5 endokrinologických pracovísk v Českej Republike.

Stupne príslušnosti atribútov  $\underline{m}_{(i)f}$  do jednotlivých fuzzy množín sú vypočítané dynamicky pomocou funkcií príslušnosti  $\mu_x(\underline{m}_{(i)f})$  pre  $\forall x \in M_V$ .

Nech

- $val(\underline{m}_{(i)f})$  – hodnota  $i$ -teho merania markeru  $m$ ,  $val(\underline{m}_{(i)f}) \in \mathbb{R}^*$ .
- $co(m)[j]$  – cut-off hodnoty oddelujúce jednotlivé intervaly koncentrácií ( $M_V$ ) pre marker  $m$ ,  $j \in \langle 1, 4 \rangle \subset \mathbb{N}$  a  $co(m)[j] \in \mathbb{R}^*$ . Funkcia berie do úvahy aj zadaný vek a pohlavie pacienta.

## 2. ALGORITMUS NA DETEKCIU PORÚCH ŠTÍTNEJ ŽLAZY



Obr. 2.6: Ukážka, do ktorých intervalov môže patriť hodnota merania markeru, pričom farebné obdĺžniky znázorňujú okolie cut-off hodnoty

Hodnota merania markeru  $m$  ( $val(\underline{m}_{(i)f})$ ) môže s určitým stupňom príslušnosti patriť maximálne do troch fuzzy množín:

1. fuzzy množina reprezentovaná intervalom  $I$  cut-off hodnôt, do ktorého hodnota merania patrí,
2. fuzzy množina reprezentovaná intervalom  $I + 1$ , pre ktorý horná cut-off hodnota ( $co_u$ ) intervalu  $I$  je spodnou cut-off hodnotou,
3. fuzzy množina reprezentovaná intervalom  $I - 1$ , pre ktorý spodná cut-off hodnota ( $co_l$ ) intervalu  $I$  je hornou cut-off hodnotou.

Hodnota merania  $val(\underline{m}_{(i)f})$  s určitým stupňom príslušnosti patrí do  $I + 1$ , ak  $I + 1$  existuje a hodnota merania je maximálne o polovicu nižšia ako  $co_u$ .

Naopak, do  $I - 1$  patrí vtedy, ak  $I - 1$  existuje a hodnota merania je maximálne o polovicu vyššia ako  $co_l$ .

Do intervalu  $I$  patrí vždy.

Z príkladu na obrázku 2.6 a vyššie uvedeného popisu je možné si všimnúť istú asymetriu medzi veľkosťou okolia spodnej a hornej cut-off hodnoty (polovica hornej cut-off je vyššie číslo ako polovica spodnej cut-off). Toto je zámerne spôsobené tým, že funkcie príslušnosti boli pôvodne vyvinuté pre nádorové markery, kde znížená hodnota nespôsobuje zdravotné problémy, naopak, čím vyššia je koncentrácia nádorového markeru, tým je ochorenie závažnejšie.

V prípade hormónov však aj znížená koncentrácia môže spôsobovať zdravotné komplikácie, preto by bolo vhodné vyskúšať upravené funkcie príslušnosti, ktoré túto asymetriu eliminujú a dávajú hornej a spodnej cut-off hodnote rovnakú váhu.

Autor algoritmu testoval rôzne typy funkcií pre  $I + 1$  a  $I - 1$ . Najlepšie výsledky dosiahol pomocou:

$$\mu_{I-1}(\underline{m}_{(i)f}) = \max \left( \left( 1,5 - \frac{val(\underline{m}_{(i)f})}{co_l} \right)^3, 0 \right), \quad co_l > 0 \quad (2.29)$$

$$\mu_{I+1}(\underline{m}_{(i)f}) = \max \left( \left( \frac{val(\underline{m}_{(i)f})}{co_u} - 0,5 \right)^3, 0 \right), \quad co_u > 0 \quad (2.30)$$

$$\mu_I = 1 - \mu_{I-1} - \mu_{I+1} \quad (2.31)$$

Kompletný formálny popis pre funkcie príslušnosti je nasledujúci:

$$\mu_{vl}(\underline{m}_{(i)f}) = \begin{cases} \max\left(\left(1,5 - \frac{val(\underline{m}_{(i)f})}{co(m)[1]}\right)^3, 0\right) & \begin{array}{l} val(\underline{m}_{(i)f}) \geq co(m)[1] \\ val(\underline{m}_{(i)f}) < co(m)[2] \\ co(m)[1] > 0 \end{array} \\ 1 - \mu_l(\underline{m}_{(i)f}) & val(\underline{m}_{(i)f}) < co(m)[1] \\ 0 & \text{inak} \end{cases} \quad (2.32)$$

$$\mu_l(\underline{m}_{(i)f}) = \begin{cases} \max\left(\left(\frac{val(\underline{m}_{(i)f})}{co(m)[1]} - 0,5\right)^3, 0\right) & val(\underline{m}_{(i)f}) < co(m)[1] \\ \max\left(\left(1,5 - \frac{val(\underline{m}_{(i)f})}{co(m)[2]}\right)^3, 0\right) & \begin{array}{l} val(\underline{m}_{(i)f}) \geq co(m)[2] \\ val(\underline{m}_{(i)f}) \leq co(m)[3] \\ co(m)[2] > 0 \end{array} \\ 1 - \mu_{vl}(\underline{m}_{(i)f}) - \mu_n(\underline{m}_{(i)f}) & \begin{array}{l} val(\underline{m}_{(i)f}) \geq co(m)[1] \\ val(\underline{m}_{(i)f}) < co(m)[2] \end{array} \\ 0 & \text{inak} \end{cases} \quad (2.33)$$

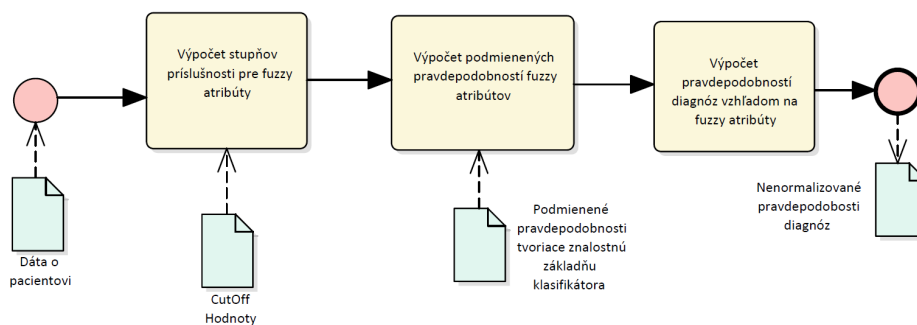
$$\mu_n(\underline{m}_{(i)f}) = \begin{cases} \max\left(\left(\frac{val(\underline{m}_{(i)f})}{co(m)[2]} - 0,5\right)^3, 0\right) & \begin{array}{l} val(\underline{m}_{(i)f}) \geq co(m)[1] \\ val(\underline{m}_{(i)f}) < co(m)[2] \end{array} \\ \max\left(\left(1,5 - \frac{val(\underline{m}_{(i)f})}{co(m)[3]}\right)^3, 0\right) & \begin{array}{l} val(\underline{m}_{(i)f}) > co(m)[3] \\ val(\underline{m}_{(i)f}) \leq co(m)[4] \end{array} \\ 1 - \mu_l(\underline{m}_{(i)f}) - \mu_h(\underline{m}_{(i)f}) & \begin{array}{l} val(\underline{m}_{(i)f}) \geq co(m)[2] \\ val(\underline{m}_{(i)f}) \leq co(m)[3] \end{array} \\ 0 & \text{inak} \end{cases} \quad (2.34)$$

## 2. ALGORITMUS NA DETEKCIU PORÚCH ŠTÍTNEJ ŽLAZY

$$\mu_h(\underline{m}_{(i)f}) = \begin{cases} \max\left(\left(\frac{val(\underline{m}_{(i)f})}{co(m)[3]} - 0,5\right)^3, 0\right) & \begin{array}{l} val(\underline{m}_{(i)f}) \geq co(m)[2] \\ val(\underline{m}_{(i)f}) \leq co(m)[3] \\ co(m)[3] > 0 \end{array} \\ \max\left(\left(1,5 - \frac{val(\underline{m}_{(i)f})}{co(m)[4]}\right)^3, 0\right) & val(\underline{m}_{(i)f}) > co(m)[4] \\ 1 - \mu_n(\underline{m}_{(i)f}) - \mu_{vh}(\underline{m}_{(i)f}) & \begin{array}{l} val(\underline{m}_{(i)f}) > co(m)[3] \\ val(\underline{m}_{(i)f}) \leq co(m)[4] \end{array} \\ 0 & \text{inak} \end{cases} \quad (2.35)$$

$$\mu_{vh}(\underline{m}_{(i)f}) = \begin{cases} \max\left(\left(\frac{val(\underline{m}_{(i)f})}{co(m)[4]} - 0,5\right)^3, 0\right) & \begin{array}{l} val(\underline{m}_{(i)f}) > co(m)[3] \\ val(\underline{m}_{(i)f}) \leq co(m)[4] \\ co(m)[4] > 0 \end{array} \\ 1 - \mu_h(\underline{m}_{(i)f}) & val(\underline{m}_{(i)f}) > co(m)[4] \\ 0 & \text{inak} \end{cases} \quad (2.36)$$

Diagram znázorňujúci jednotlivé kroky tejto fázy algoritmu sa nachádza na obrázku 2.7.



Obr. 2.7: Diagram, zobrazujúci kroky fuzzy naivného bayesovského klasifikátora

### 2.2.2 Aplikácia informácií z epidemiológie

Algoritmus používa prevalenciu ochorení získanú z dát Všeobecnej zdravotní pojišťovny zo 4 okresov, pričom dva reprezentujú mestské okresy a dva vidiek.

Prevalencia je v jednotkách na 100 000 obyvateľov daného pohlavia a veku (vek je braný po desiatkach rokov, tj. 0 – 9, 10 – 19 atď).

Pre diagnózy týkajúce sa užívania liekov neboli k dispozícii liekové prevalencie, preto sú použité odhady odborníkov z oblasti endokrinológie.

Ako bolo povedané v predchádzajúcej kapitole 2.2.1, prevalencie nahradzujú  $P(D = d)$ , avšak nemôžu byť použité v základnom formáte, pretože výrazne znevýhodňujú ochorenia, ktoré sa vyskytujú málo, naopak favorizujú časté ochorenia, aj keď chovanie markerov na ne sedí horšie.

Z tohto dôvodu je prevalencia preškálovaná logaritmicky:

$$preval(d, age, sex) = \ln(1 + preval(d, age, sex)), \quad (2.37)$$

kde  $d$  značí diagnózu,  $age$  vekové rozmedzie a  $sex$  pohlavie.

Pravdepodobnosti, ktoré sú výstupom fuzzy naivného bayesovského klasifikátora sú vynásobené preškálovanou prevalenciou pre vek a pohlavie pacienta.

### 2.2.3 Indexy

Niektoré diagnózy majú podobné chovanie koncentrácií markerov vzhľadom na definovaných 5 kategórií a len aplikáciou základného modelu reprezentovaného fuzzy naivným bayesovským klasifikátorom nie sú od seba odlišiteľné.

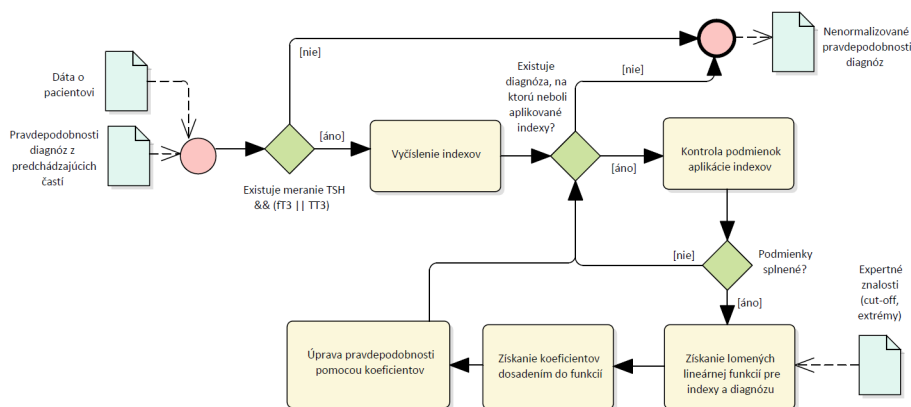
Na odlišenie je možné použiť pomery koncentrácií niektorých markerov voči sebe. V súčasnej verzii algoritmu sú používané dva indexy TSH/TT3 a TSH/fT3. Výpočet indexu sa deje vždy z najnovších meraní daných dvoch markerov.

Pre každý index a diagnózy, ktoré je nutné odlíšiť, existujú expertne dané cut-off hodnoty. Intervaly medzi nimi opäť reprezentujú veľmi zníženú, zníženu, normálnu, zvýšenú a veľmi zvýšenú hodnotu pomeru. Ďalej pre každú kombináciu index-diagnóza existujú lineárne lomené funkcie určené na korekciu pravdepodobností, odvodené z cut-off hodnôt a expertných odhadov o vplyve pomerov na pravdepodobnosť diagnózy.

Nech

- $val(i)$  – hodnota indexu  $i$ ,  $i \in \{\text{TSH/TT3}, \text{TSH/fT3}\}$ ,
- $dg$  – diagnóza  $dg$  patriaca medzi diagnózy, ktoré je nutné odlíšiť,
- $co(i, dg)[j]$  – cut-off hodnoty oddeľujúce jednotlivé intervaly koncentrácií pre daný  $i$  a diagnózu  $dg$ ,  $j \in \langle 1, 4 \rangle \subset \mathbb{N}$  a  $co(i, dg)[j] \in \mathbb{R}^*$ ,
- $f_{i, dg}(x)$  – lineárna lomená funkcia pre index  $i$  a diagnózu  $dg$ .

## 2. ALGORITMUS NA DETEKCIU PORÚCH ŠTÍTNEJ ŽLAZY



Obr. 2.8: Diagram, zobrazujúci proces aplikácie indexov

Tvar funkcie  $f_{i,dg}(x)$  je možné vyjadriť ako:

$$f_{i,dg}(x) = \begin{cases} \alpha & x \in \langle 0, co(i, dg)[1] \rangle \\ \text{smernicový tvar rovnice} \\ \text{priamky danej bodmi} \\ (co(i, dg)[1], \alpha) \text{ a } (co(i, dg)[2], 1) & x \in (co(i, dg)[1], co(i, dg)[2]) \\ 1 & x \in \langle co(i, dg)[2], co(i, dg)[3] \rangle \\ \text{smernicový tvar rovnice} \\ \text{priamky danej bodmi} \\ (co(i, dg)[3], 1) \text{ a } (co(i, dg)[4], \beta) & x \in \langle co(i, dg)[3], co(i, dg)[4] \rangle \\ \beta & x \in \langle co(i, dg)[4], \infty \rangle \end{cases} \quad (2.38)$$

pričom  $\alpha$  a  $\beta$  sú expertné odhady vplyvu pokiaľ je index veľmi znížený alebo veľmi zvýšený. V prípade, že je hodnota indexu normálna, sa pravdepodobnosť neprepočítava. Pre znížený a zvýšený index sa dopočíta koeficient vplyvu podľa predpisu.

Funkcia je aplikovaná na  $val(i)$ .

Celkový proces aplikácie indexov na pravdepodobnosti diagnóz je zobrazený na obrázku 2.8. Lomené lineárne funkcie sú aplikované len vtedy, ak sú splnené určité podmienky (napr. ohľadom koncentrácie markerov), pretože nie je vhodné upravovať pravdepodobnosť diagnózy, pokiaľ nie je relevantná.

### 2.2.4 Korekcie pomocou expertných znalostí

Fuzzy naivný bayesovský klasifikátor dáva všetkým atribútom rovnakú váhu pri určovaní výslednej pravdepodobosti diagnózy.

Avšak v realite sa môže stať, že sú niektoré atribúty dôležitejšie ako iné. Napríklad TG je dôležitým indikátorom možného nádoru štítnej žľazy [14]. Ak sú jeho hodnoty veľmi zvýšené, nádor je veľmi pravdepodobný, aj keď chovanie ostatných parametrov tomu nemusí nasvedčovať. Naopak pokiaľ je TG normálne, pravdepodobne sa nebude jednať o nádor.

Algoritmus teda využíva súbor if-then pravidiel, pričom v if časti sa nachádza podmienka týkajúca sa typu diagnózy a nejakého chovania najnovšieho merania markeru a časť then hovorí o tom, akým koeficientom by sa mala prepočítať pravdepodobnosť pre diagnózu, ak je podmienka splnená.

Hodnoty koeficientov boli stanovené po diskusii s odborníkmi z oblasti endokrinológie.

### 2.2.5 Zhodnotenie progresu liečby

Zhodnotenie progresu liečby sa vykonáva v prípade diagnóz týkajúcich sa hypertyreózy alebo hypotyreózy.

Možné zhodnotenie nastáva v prípade, že v dátach o pacientovi sa nachádza predchádzajúca diagnóza alebo sú zadané meranie z aspoň dvoch rozličných dátumov.

V prípade zadanej predchádzajúcej diagnózy musí byť prítomné aj meranie aspoň jedného z markerov TSH, TT4, fT4, TT3, fT3. V prípade viacerých meraní je použité posledné. Na merania a diagnózu sa aplikujú znalosti, ktoré sú popísané v tabuľke 2.1. V prípade, že nie je možné aplikovať ani jedno pravidlo progres nie je zhodnotený.

Ak dáta o pacientovi neobsahujú predchádzajúcu diagnózu alebo zhodnotenie progresu pomocou tabuľky 2.1 bolo neúspešné a zároveň sú prítomné aspoň dve merania aspoň jedného z markerov TSH, TT4 alebo fT4 sú aplikované znalosti popísané v tabuľke 2.2. V prípade, že sú zamerané dva a viac markerov, musí nastať zhoda v chovaní pre diagnózu aspoň dvoch z nich. Ak nie je možné aplikovať ani jedno pravidlo, progres nie je zhodnotený.

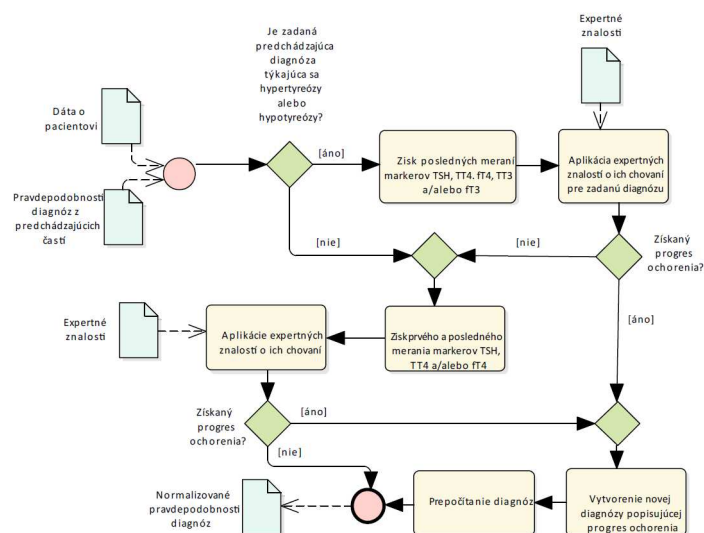
V prípade existujúceho zhodnoteného progresu sa tento progres vloží na prvé miesto zoznamu možných diagnóz s pravdepodobnosťou 0,82 a pravdepodobnosti ostatných diagnóz sú prepočítané. Táto pravdepodobnosť bola určená z analýzy dát endokrinologických pracovísk skúmaním v kolkých percentách prípadov sa jedná o progres ochorenia.

Proces zhodnotenia progresu ochorenia je znázornený na obrázku 2.9.

## 2. ALGORITMUS NA DETEKCIU PORÚCH ŠTÍTNEJ ŽLAZY

Koncentrácia		Predchádzajúca diagnóza	Progres
TSH	Ostatné		
normálna	normálna	Hypotyreóza	Vyliečená hypotyreóza
normálna	normálna	Hypertyreóza	Vyliečená hypertyreóza
znižovaná	zvýšená	Hypotyreóza	Preliečená hypotyreóza
znižovaná	zvýšená	Hypertyreóza	Nestabilizovaná hypertyreóza
zvýšená	znižovaná	Hypotyreóza	Nestabilizovaná hypotyreóza
zvýšená	znižovaná	Hypertyreóza	Preliečená hypertyreóza

Tabuľka 2.1: Znalosti pre zhodnotenie priebehu liečby predchádzajúcej diagnózy



Obr. 2.9: Diagram, zobrazujúci proces zhodnotenia progresu ochorenia



Koncentrácia		Chovanie markeru svedčí o diagnóze		
1. meranie	Posledné meranie	TSH	TT4	fT4
Zvýšená	Normálna	Vyliečená hypo-tyreóza	Vyliečená hyper-tyreóza	Vyliečená hyper-tyreóza
Zvýšená	Znížená	Preliečená hypo-tyreóza	Preliečená hyper-tyreóza	Preliečená hyper-tyreóza
Znížená	Normálna	Vyliečená hyper-tyreóza	Vyliečená hypo-tyreóza	Vyliečená hypo-tyreóza
Znížená	Zvýšená	Preliečená hyper-tyreóza	Preliečená hypo-tyreóza	Preliečená hypo-tyreóza

Tabuľka 2.2: Znalosti pre zhodnotenie priebehu liečby predchádzajúcej diagnózy z meraní markerov



---

# Analýza historickej implementácie programu DITHYRON

Historická implementácia programu DITHYRON nie je v praxi použiteľná, pretože trpí mnohými z problémov CDSS popísaných v podkapitole 1.1.2.3. Zároveň nefunkčné požiadavky na aplikáciu v 90. rokoch minulého storočia limitujú jej nasadenie v súčasnosti. Navyiac zdrojový kód samotnej implementácie nie je členený do funkčných celkov podľa logiky algoritmu popísaného v kapitole 2, jednotlivé funkcionality sú pomiešané a k algoritmu neexistuje žiadna technická dokumentácia.

Táto kapitola analyzuje historickú implementáciu z hľadiska limitujúcich technologických nefunkčných požiadavkov a architektúry, business procesov a nedostatkov historickej implementácie, ktoré je nutné pri návrhu nového modulu zohľadniť a vyvarovať sa im.

## 3.1 Technológie

Aplikácia bola vyvinutá v jazyku Pascal pre počítače s operačným systémom MS-DOS (Microsoft Disk Operating System) vo vývojovom prostredí (IDE) s prekladačom Borland Turbo Pascal 5.0.

Nasledujúce informácie o technológii pochádzajú z užívateľskej príručky [46].

Programovací jazyk Pascal je imperatívny procedurálny programovací jazyk. Obsahuje základné dátové typy ako `integer` a `string` a zároveň pokročilejšie typy ako `record`, `pointer` a `array`. Pričom programátor si môže nadefinovať vlastné dátové typy.

Typ `record` slúži na kompozíciu dátových typov (viď. ukážka kódu 3.1).

Premenné môžu byť deklarované ako globálne na začiatku programu alebo ako lokálne na začiatku procedúry (funkcie). Deklarácia v segmente kódu nie

### 3. ANALÝZA HISTORICKEJ IMPLEMENTÁCIE PROGRAMU DITHYRON

---

je možná. Dĺžka premennej nie je obmedzená, avšak iba prvých 63 bitov sa používa na rozlíšenie premenných.

```
1 type
2   osoba = record
3       vek : integer;
4       pohlavie : char;
5   end;
6 var
7   osoby : array [1..20] of osoba;
```

Ukážka zdrojového kódu 3.1: Definícia nového dátového typu osoba a premennej znázorňujúcej pole osôb

Konštrukty jazyka sú typické pre imperatívne procedurálne jazyky. Patria sem operátory (napr. aritmetické operátory, operátory priradenia, logické operátory ...), funkcie a procedúry, príkazy pre vstup a výstup, podmienky a cykly.

Nevýhodou je, že štandard používaný Borland Turbo Pascalom neobsahuje konštrukt `break` určený na predčasné opustenie cyklu.

Obmedzenia 16-bitových procesorov tej doby (napr. Intel 8086) vyžadovali, že segment kódu nesmie prekročiť veľkosť 64 kB. Programy teda museli byť delené na maximálne 64 kB veľké súbory tzv. *units*, ktoré spojovací program (linker) Turbo Pascalu prepojoval.

Celkové obmedzenie na veľkosť programu (vrátane dát a pamäte alokovanej programom) bolo dané veľkosťou operačnej pamäte počítača a zväčša sa jednalo o 640 kB až 1 MB. V prípade, že program potreboval viac pamäte bolo nutné definovať závislosti medzi časťami programu (units), aby bolo jasné, ktoré units musia byť naraz prítomné v operačnej pamäti.

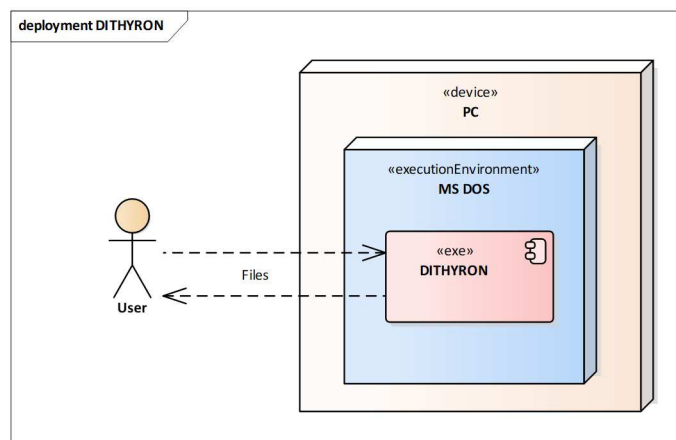
## 3.2 Architektúra

Historická aplikácia DITHYRON [47] je desktopovou aplikáciou pre operačný systém MS-DOS (diagram nasadenia na obrázku 3.1). Aplikácia neobsahuje GUI a komunikuje s užívateľom pomocou súborov.

Ukážka vstupného súboru:

```
----- vstup.txt -----
pacient:Kratky Tomas [7201257789] E05.0      4
6.1.2017 ft4 1.17
2.11.2017 TSH 0.08
-----
```

Aplikácia je zložená z exe súboru a konfiguračných súborov určených na nastavenie referenčných intervalov, formátovania výstupu a prekladu reťazcov.



Obr. 3.1: Diagram nasadenia historickej verzie programu DITHYRON

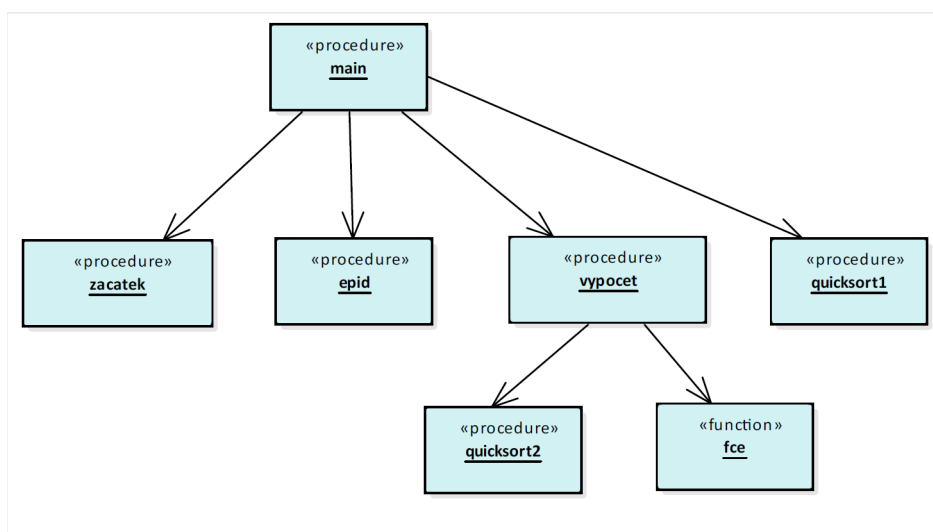
Zdrojový kód pozostáva z dvoch súborov (z dôvodu obmedzenia maximálnej veľkosti časti programu na 64kB) THYROID.PAS (obsahuje 1383 riadkov kódu) a KONEC.PAS (obsahuje 1588 riadkov kódu).

Program je zložený z nasledujúcich procedúr a funkcií (kompozícia sa nachádza na obrázku 3.2):

- **zacatek** – slúži na inicializáciu poľa podmienených pravdepodobností pre fuzzy naivný bayesovský klasifikátor,
- **epid** – slúži na inicializáciu poľa prevalencií,
- **vypocet** – celá implementácia algoritmu popísaného v kapitole 2, vrátane všetkých expertných znalostí a zároveň výpis výsledkov do súboru,
- **quicksort1** – slúži na radenie laboratórnych výsledkov pacienta v procedúre **main**,
- **quicksort2** – slúži na radenie pravdepodobností diagnóz používaných v procedúre **vypocet**,
- **fce** – funkcia, ktorá dáva číslo na tretiu,
- **main** – slúži na načítanie a spracovanie konfigurácie z konfiguračných súborov, inicializáciu potrebných dátových štruktúr, načítanie a spracovanie vstupných súborov, analýzu koncentrácií markerov vzhľadom na referenčné intervaly, zápis tejto analýzy do súboru a volá procedúry **zacatek**, **epid** a **vypocet**.

Procedúry medzi sebou komunikujú využitím globálnych premenných.

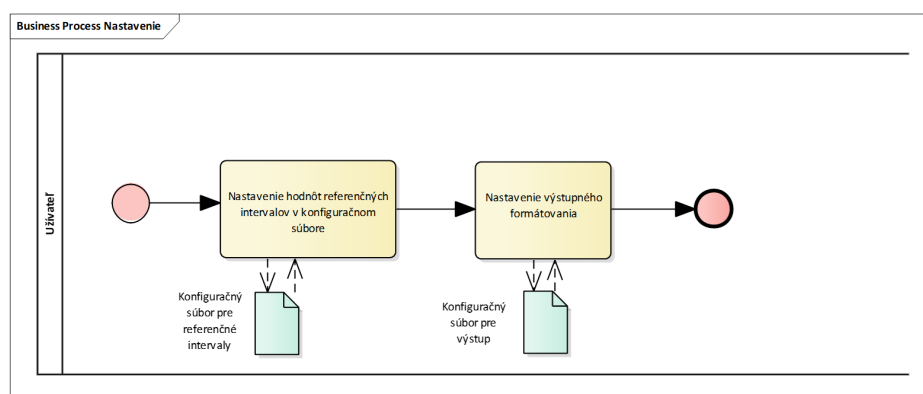
### 3. ANALÝZA HISTORICKEJ IMPLEMENTÁCIE PROGRAMU DITHYRON



Obr. 3.2: Kompozícia funkcií a procedúr historickej verzie programu DITHYRON

### 3.3 Business proces používania aplikácie – AS IS

Historická verzia programu DITHYRON umožňuje nastaviť hodnoty referenčných intervalov pomocou konfiguračného súboru, zároveň umožňuje nastaviť prahovú hodnotu pravdepodobnosti, do ktorej má program vydávať diagnózy a maximálny počet diagnóz na výstupe. Procesný diagram pre proces nastavenia programu sa nachádza na obrázku 3.3.



Obr. 3.3: Proces nastavenia programu DITHYRON

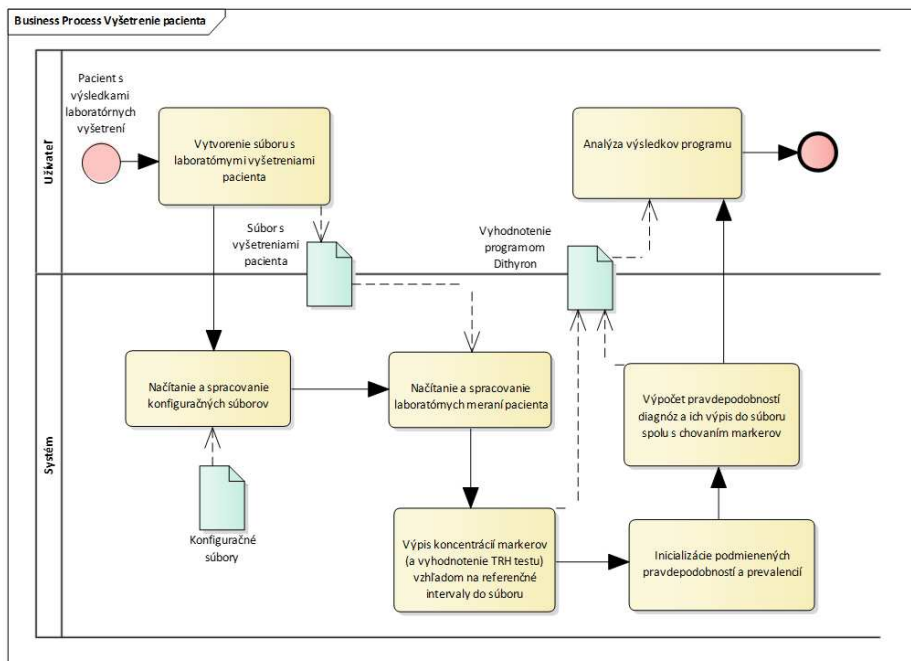
Základnou funkcionalitou je pomoc s vyšetrovaním pacienta. Pričom užívateľovi ponúka ako výstup:

1. zhodnotenie koncentrácií markerov vzhľadom na referenčné intervaly,

### 3.4. Nedostatky historickej implementácie

2. možné diagnózy s informáciou o pravdepodobnosti a o chovaní markerov (umožňuje lekárovi vyhodnotiť, ktoré ďalšie markre by bolo dobré zmerať).

Proces vyšetrovania pacienta sa nachádza na obrázku 3.4.



Obr. 3.4: Proces vyšetrenia pacienta programom DITHYRON

Inú funkcionlitu ako napríklad prepojenie s medicínskymi systémami alebo správu predchádzajúcich vyšetrení program neponúka.

Správa vstupných a výstupných súborov je na uvážení užívateľa.

### 3.4 Nedostatky historickej implementácie

Táto podkapitola zhodnocuje nedostatky historickej implementácie z rôznych pohľadov: nedostatky z pohľadu užívateľa, nedostatky vzhľadom na rozšíriteľnosť a údržbu programu, nedostatky spôsobené cieľmím na a využívaním zastaraných technológií, odhalené chyby vo funkčnosti a neefektívne nakladanie s prostriedkami.

Niektoré z týchto nedostatkov sú typickými zástancami vzorov (anti-patterns), ktoré sa na vývoj softvéru nemajú používať.

#### 3.4.1 Nedostatky z pohľadu užívateľa

Táto podkapitola prináša zhrnutie nedostatkov spojených s používaním programu.

##### 3.4.1.1 Chýbajúca užívateľská dokumentácia

Napriek tomu, že systém neobsahuje užívateľské rozhranie a na komunikáciu využíva súbory, neexistuje dokumentácia pre užívateľa, ktorá by mu ozrejmila akým spôsobom majú vyzeráť vstupné súbory a aké sú možnosti konfigurácie programu.

##### 3.4.1.2 Chýbajúca validácia vstupov

Program neobsahuje validáciu konfiguračných súborov a v prípade porušenia formátu konfiguračného súboru (napríklad chýbajúca hodnota) program skončí s chybou (`runtime error`).

Nezmyselné riadky alebo riadky s neznámymi markerami v súboroch s dátami pacienta sú programom preskočené, chyba v hlavičke týkajúca sa rodného čísla pacienta spôsobuje pád programu.

Neexistujúca validácia vstupov spolu s chýbajúcou užívateľskou dokumentáciou predstavujú riziko neočakávaných pádov programu.

Tento nedostatok je typickým prípadom vzoru *improbability factor*, ktorý hovorí o tom, že sa nepredpokladá, že by nejaká známa chyba mohla nastať, preto nie sú vyhradené prostriedky na jej opravu.

##### 3.4.1.3 Manuálne aktualizácie verzií

V prípade vzniku novej verzie programu je nutné si ju manuálne nainštalovať, pričom vzniká možné riziko zastarania programu v prípade neaktualizovania, prípadne používania rôznych verzií na viacerých počítačoch jedného pracoviska.

##### 3.4.1.4 Aplikácia nespĺňa požiadavky na súčasné technológie lekárov

Aplikácia nie je spustiteľná na súčasných operačných systémoch inak ako v emulátore operačného systému MS-DOS a preto je v tomto stave nenasaditeľná.

Rovnako nezohľadňuje požiadavky na prepojenie so súčasnými nemocničnými informačnými systémami (napr. STAPRO Medea, ktorý v roku 2006 mal 40 % podiel na trhu v Českej Republike a na Slovensku [48]) alebo spustiteľnosť na zariadeniach ako mobily a tablety, ktoré sú najmä v zahraničí už rutinne používané v medicínskej praxi [3].



### 3.4.1.5 Chýbajúca správa vyšetrení pacientov

Aplikácia neobsahuje žiadnu funkcionálnosť, ktorá by umožňovala správu vyšetrení pacientov alebo ich zdieľanie v rámci inštitúcie. Výsledky vyšetrení sú ukladané do súborov, ktoré je možné manuálne zdieľať s iným špecialistom, avšak vyžaduje to ďalšie úsilie užívateľa.

### 3.4.1.6 Nedostatok v prezentácii výsledkov

Chovanie markerov pri diagnóze, ktoré program zobrazuje ako doplnok k samotným pravdepodobnostiam diagnózy, umožňujú lekárovi porovnať chovanie jednotlivých markerov pri rôznych typoch diagnóz a určiť, ktoré ďalšie markery by bolo dobré zmerať.

Ukážka výstupu s chovaním markerov [47]:

---

vystup.txt

---

Nejvice pravdepodobne nasledujici diagnozy a stavy:

- ```
1) idiopaticka hypofunkce stitne zlazy ( idiopaticky myxedem ) ->
  -> pravdepodobnost :    41.7 %
  Ch: TSH Z, TT4 S, FT4 S, TT3 SN, FT3 NS
      TBG ZN, aTG Z, aTPO Z(N), TG N
      TT4/TBG S, je-li TT3=S je TSH/TT3>40

2) subklinicka hypofunkce stitne zlazy (oligosymptomaticka funkce) ->
  -> pravdepodobnost :    26.0 %
  Ch: TSH Z, TT4 N, FT4 SN, TT3 NS, FT3 NS
      TBG N, aTG N, aTPO ZN, TG N
      TT4/TBG N

3) difuzni lymfoidni tyreoiditida Hashimoto ->
  -> pravdepodobnost :    16.8 %
  Ch: TSH Z, TT4 NSZ, FT4 N, TT3 NS(Z), FT3 N
      TBG N, aTG Z, aTPO Z, TG ZN
      TT4/TBG N
```

Ch: = typicke chovani parametru u dane diagnozy  
N = normalni hodnota  
Z = zvysena hodnota  
S = snizena hodnota

---

Prezentácia chovania nie je úplne prehľadná, chýba popis, že hodnoty v zátvorke nastávajú len zriedkavo, a že poradie písmen určuje, ktoré chovanie je najtypickejšie.

### 3.4.2 Nedostatky vzhľadom na rozšíriteľnosť a údržbu programu

Táto podkapitola popisuje nedostatky, ktoré spôsobujú problémy s nasledujúcou údržbou programu a zároveň komplikujú jeho rozšíriteľnosť o novú funkcionalitu, správu znalostí alebo znovupoužitie jeho častí v iných kontextoch.

#### 3.4.2.1 Neprehľadná vnútorná architektúra

Štruktúra zdrojového kódu predstavuje vzor *Spaghetti Code*. Spaghetti Code sa vyskytuje v programoch, ktoré postrádajú logické členenie kódu a to až do takej miery, že častokrát po uplynutí určitej doby nie je jasné ani samotnému autorovi kódu, čo sa v ktorej časti vykonáva [49].

V rámci programu DITHYRON sa tento vzor prejavuje tým, že jednotlivé procedúry majú príliš mnoho zodpovedností. Napríklad procedúra `vypocet` implementuje celý algoritmus popísaný v kapitole 2. Navyše jednotlivé časti algoritmu od seba nie sú oddelené a prelínajú sa. Okrem výpočtu má procedúra na starosti aj výpis výsledkov do súboru, ktoré sa vypisujú priebežne a miešajú sa s kódom algoritmu.

Absencia členenia kódu do menších funkcií a procedúr vynucuje využívanie rozsiahlych vnorených podmienok a cyklov, príklad sa nachádza na ukážke kódu 3.2.

```
1 for i:=1 to 40 do begin
2   nas:=ln(1+pr[i]);
3   if (vek>2) and (i=37) then nas:=0;
4   if ma14<>-1 then begin
5     if (i=2) and (tt3<=hl[4,2]*1.1) then
6       begin
7         if ma14>=60 then nas:=nas*6;
8         if (ma14>=35) and (ma14<60) then nas:=nas*(ma14-30)/5;
9         if ma14<=5 then nas:=nas/3;
10        if (ma14>5) and (ma14<15) then nas:=nas*ma14/15;
11        end;
12        ...
13    end;
14  for j:=1 to pocet do
15    begin
16      ii:=mark[j];
17      if hlad[j]<hl[ii,1] then
18        begin
19          ...
20        end;
21    end;
22    ...
23  end;
```

Ukážka zdrojového kódu 3.2: Vnorené cykly a znalosti zakódované do zdrojového kódu DITHYRON [47]

Naviac sú niektoré znalosti priamo zakódované do zdrojového kódu. Na riadkoch 4–12 ukážky 3.2 sa nachádza aplikácia indexu na pravdepodobnosť. Pričom `ma14` značí hodnotu indexu a `i` diagnózu, ktorej pravdepodobnosť sa upravuje.

Hodnoty 5, 15, 35 a 60 na riadkoch 7–12 reprezentujú cut-off hodnoty pre index `ma14` a diagnózu `i`. V časti then sa nachádzajú časti lineárnej lomenej funkcie (viď podkapitolu 2.2.3), pričom rovnice sú manuálne vypočítané z cut-off hodnôt.

Takto zakódované znalosti komplikujú údržbu a nie je možné ich jednoducho zmeniť.

#### 3.4.2.2 Porušovanie princípu Don't repeat yourself (DRY)

V rámci zdrojového kódu je tiež využívaný vzor *Cut-And-Paste Programming*, kedy sú jednotlivé časti kódu prekopírované na rôznych miestach. Využívanie tohto vzoru porušuje princíp *DRY*, ktorý hovorí o tom, že každá dielčia znalosť musí mať v programe jedinú a jednoznačnú reprezentáciu [49].

#### 3.4.2.3 Neexistujúca programátorská dokumentácia

Zdrojový kód programu neobsahuje žiadne komentáre a zároveň nie je dostupná ani iná forma programátorskej dokumentácie.

Chýbal aj obecný popis algoritmu, ktorého formulácia bola jedným z cieľov tejto diplomovej práce (viď kapitola 2).

Tento nedostatok sťažuje údržbu kódu a komplikuje možné rozširovanie programu inou osobou ako autorom algoritmu.

#### 3.4.2.4 Chýbajúce testy

K aplikácie neboli napísané žiadne testy, ktoré by testovali jednotlivé časti programu. Údržba programu s nejasnou architektúrou a prelínaním funkčností je bez testov ešte náročnejšia, pretože nie je jasné, aké všetky časti systému možná zmena (oprava chyby) ovplyvní.

#### 3.4.2.5 Používanie názvov premenných, procedúr a funkcií, ktoré nedostatočne reflektujú to, čo popisujú

Napríklad názov metódy `zacatek` nijakým spôsobom neindikuje, že sa v metóde vykonáva inicializácia poľa podmienených pravdepodobností. Rovnako názov funkcie `fce` neindikuje, že sa jedná o funkciu, ktorá počíta tretiu mocninu čísla.

Názvy niektorých premenných sú tiež nejasné (viď ukážku kódu 3.3 zobrazujúcu deklaráciu globálnych premenných). Napríklad premenné `rec`, `REC1`, `REC2` obsahujú jednotlivé riadky popisu chovania markerov pre diagnózu, pole

### 3. ANALÝZA HISTORICKEJ IMPLEMENTÁCIE PROGRAMU DITHYRON

---

`prp` obsahuje prevalenciu diagnóz a premenná `st` je používaná na načítavanie častí súborov.

Nezrozumiteľné názvy premenných a metód sťažujú orientáciu v kóde.

```
1 var
2 hloupost : file of word ;
3 chacha : char ;
4 hl : array [1..28 , 1..4] of double ;
5 pr : array [1 ..48] of double ;
6 pravd , prdiag : array [1 ..48] of double ;
7 mark : array [1..250] of integer ;
8 tbghlad , psahlad , hlad : array [1..250] of double ;
9 pocet , iii , ijk , code , ii , jj , i , j , k , vek : integer ;
10 ay , by , cy , dy , py , yyy , mmm , dddd , nnn : word ;
11 pomert , pomer , max , r1 , r2 , r3 , r4 , r5 , proc , sum , nas : double ;
12 f , ff : text ;
13 jetbg , nizkypomer , sex , uzbylo : boolean ;
14 rec , REC1 , REC2 : array [1..46] of string [68] ;
15 ppopis , st , popis , fuj : string ;
16 tbgdd , tbgmm , psadd , psamm , dd , mm : array [1..250] of byte ;
17 yy , psayy , tbgyy : array [1..250] of integer ;
18 material : array [1..250] of string [5] ;
19 maxdd , maxmm , maxyy , mindd , minmm , minyy : integer ;
20 p : array [1..13 , -2..2 , 1..40] of single ;
21 prp : array [0..7 , 0..1 , 1 ..40] of single ;
```

Ukážka zdrojového kódu 3.3: Definícia časti globálnych premenných v programe DITHYRON [47]

#### 3.4.2.6 Pevne daná veľkosť dynamických premenných

Veľkosti polí alebo reťazcov sú pevne dané. Maximálny limit na počet laboratórnych meraní pacienta je 250 (viď premennú `mark` v ukážke kódu 3.3). V prípade menšieho počtu nastáva zbytočné alokovanie pamäte, väčší počet meraní je veľmi nepravdepodobný, ale spôsobil by chybu programu.

V niektorých prípadoch je chybne zadaná veľkosť premenných, kde je dopredu známe koľko ich bude. Napríklad pole cut-off hodnôt `hl` sa alokuje pre 28 markerov, pričom v poslednej verzii programu je ich používaných len 13, pole výsledných pravdepodobností diagnóz `prdiag` sa alokuje pre 48 diagnóz, pričom maximálne sa ich v poli môže nachádzať 41 (viď ukážku kódu 3.3).

Tento nedostatok spôsobuje zbytočnú alokáciu pamäte a zároveň zhoršuje čitateľnosť zdrojového kódu.

#### 3.4.2.7 Nevyužívanie pokročilých konštruktov jazyka

Implementácia programu nevyužíva pokročilejších konštruktov jazyka ako sú ukazatele (`pointer`), ktoré by umožnili dynamickú alokáciu pamäte a pomohli by vyriešiť problém popísaný v predchádzajúcej podkapitole 3.4.2.6. Je však

možné, že použitie ukazateľov v tejto architektúre by spôsobilo zhoršenie čitateľnosti kódu.

Čitateľnosť zhoršuje aj nepoužitie konštruktu `record` (viď ukážku kódu 3.1), ktorý by mohol byť použitý napríklad na uchovávanie informácií o pacientovi v jednej štruktúre namiesto individuálnych premenných `vek`, `pohlavi` a `dg`.

Rovnako by tento konštrukt mohol byť použitý na uchovávanie laboratórnych meraní pacienta, ktoré sú v súčasnosti ukladané vo viacerých poliach `dd` (deň), `mm` (mesiac), `yy` (rok), `hlad` (koncentrácia merania) a `mark` (meraný marker), ktoré sú spolu prepojené rovnakou hodnotou indexu.

#### 3.4.2.8 Pozostatky kódu, ktoré sa nikde nevyužívajú

Zdrojový kód rovnako obsahuje mnoho pozostatkov kódu, ktoré sa nikde nevyužívajú. Týka sa to napríklad definície premenných ako `hloupost` a `chacha` (viď ukážku kódu 3.3), ktoré boli pravdepodobne použité na ladenie programu.

Rovnako sa v kóde nachádzajú aj časti algoritmu, kde sa niečo počíta, ale výsledok sa nikde nevyužije. Takýmto príkladom je výpočet indexov prtilátok zobrazeným na ukážke kódu 3.4.

```

1 if (atpo>=0) and (atshr>=0) then
2   if atshr>0 then ma16:=ln(1+ln(1+hl[9,3]*atpo/hl[8,3]/atshr))
3   else ma16:=150;
4 if (atg>=0) and (atshr>=0) then
5   if atshr>0 then ma17:=ln(1+hl[9,3]*atg/hl[7,3]/atshr)
6   else ma17:=150;

```

Ukážka zdrojového kódu 3.4: Výpočet indexov prtilátok [47]

Nevyužitie pozostatky kódu reprezentujú vzor *lava flow*. Lava flow je vzor, ktorý sa často vyskytuje v systémoch, ktoré pochádzajú z výskumnej oblasti a skúšajú sa v nich rôzne varianty funkcionality. Pozostatky tejto funkcionality ostávajú zabudnuté v zdrojovom kóde [49].

### 3.4.3 Technologické nedostatky

Technologické nedostatky sú nedostatky, ktoré vznikli orientovaním sa na technológie, ktoré už v dnešnej dobe nie sú využívané.

#### 3.4.3.1 Obmedzenia hardvéru 90. rokov

Obmedzenia hardvéru popísané v podkapitole 3.1 spojené s veľkosťou pamäte vynucovali jej šetrenie. Správa veľkého množstva dát (laboratórnych výsledkov) v rámci aplikácie nebola možná.

#### 3.4.3.2 Nedostatky zvoleného jazyka

Turbo Pascal 5.0 neumožňuje navrhnuť systémy inak ako v podobe desktopovej aplikácie a to je pri požiadavkách na dostupnosť, jednoduchú aktualizáciu

a zdieľanie výsledky nedostačujúce. Naviac výsledná aplikácia nie je spustiteľná na súčasných operačných systémoch.

Zároveň jeho syntax neobsahuje `break`, čo vynucuje využívanie iných metód ako `goto`, čo zneprehľadňuje kód, pretože programátor musí hľadať na aké miesto programu sa skáče.

#### 3.4.4 Chyby vo funkčnosti

Analýza zdrojového kódu odhalila aj niekoľko chýb, ktoré aplikácia obsahuje vďaka neexistencii testov.

##### 3.4.4.1 Chyba pri vyhodnocovaní progresu ochorenia

Progres ochorenia sa zhodnocuje vzhľadom na výsledky meraní markerov TSH, TT4, fT4, TT3 a fT3. Ak má premenná reprezentujúca meranie hodnotu -1, značí to, že meranie nebolo vykonané.

Podmienky pre zhodnocovanie progresu obsahujú chybu (viď ukážka kódu 3.5), ktorá pri neprítomnosti akéhokoľvek jedného merania uvedených markerov vyhodnotí podmienku ako pravdivú. Podmienka je pravdivá aj vtedy, pokiaľ žiadne meranie týchto markerov nebolo vykonané.

```
1 if ((tsh=-1) or (tsh>=hl[1,3])) or
2 ((tt4=-1) or (tt4<=hl[2,2])) or
3 ((ft4kon=-1) or (ft4kon<=hl[3,2])) or
4 ((tt3=-1) or (tt3<=hl[4,2])) or
5 ((ft3=-1) or (ft3<=hl[5,2]))
6 then krit:=45;
```

Ukážka zdrojového kódu 3.5: Chyba v podmienke pri vyhodnocovaní progresu ochorenia [47]

##### 3.4.4.2 Problém s hranicami intervalov

V niektorých prípadoch nie sú dobre implementované hranice disjunktných intervalov (viď ukážka kódu 3.6).

```
1 if tyreog<0.7 then if sum/80<prdiag[20] then prdiag[20]:=sum/80;
2 if tyreog<1 then prdiag[20]:=prdiag[20]/4; {interval <0.7,1}
3 if (tyreog>2) then prdiag[20]:=prdiag[20]*3;
4 ...
5 if ma14>=70 then nas:=nas/3; {70 patrí do oboch intervalov}
6 if (ma14>58) and (ma14<=70) then nas:=nas*(76-ma14)/18;
```

Ukážka zdrojového kódu 3.6: Chybná hranica disjunktného intervalu [47]

##### 3.4.4.3 Problém s prechodom milénia

Vzhľadom na dobu, v ktorej program vznikol sa nerieši prechod milénia, pričom sa vek extrahuje z rodného čísla pacienta. K veku pacientov narodených po roku 1999 je pripočítaných 100 rokov.

### 3.4.5 Neefektivita

Aplikácia obsahuje aj niekoľko neefektívnych riešení. Týmto prípadom je napríklad plynutie pamäťou pri definícii premenných (viď podkapitola 3.4.2.6), vykonávanie zbytočných častí kódu (viď podkapitola 3.4.2.8) alebo vykonávanie celého výpočtu procedúry `vypocet` aj keď sú všetky hodnoty meraní normálne a na záver sa vzhľadom na ne vypíše, že pacient je zdravý.

### 3.4.6 Záver analýzy nedostatkov

Z analýzy nedostatkov historickej aplikácie vyplynulo niekoľko úkonov, ktoré v rámci návrhu a implementácie modulu DITHYRON musia byť vykonané. Prehľad úkonov sa nachádza v tabuľke 3.1.

| #  | Úkon                                                                                                    | Kategória                 |
|----|---------------------------------------------------------------------------------------------------------|---------------------------|
| 1  | Navrhnuť GUI, ktoré bude citlivé na možné chyby užívateľa                                               | Použitelnosť              |
| 2  | Navrhnuť GUI, ktoré bude jednoducho použiteľné                                                          | Použitelnosť              |
| 3  | Navrhnuť spôsob ako čitateľne prezentovať výsledky vyšetrení                                            | Použitelnosť              |
| 4  | Umožniť podporu pre širokú škálu zariadení bez ohľadu na veľkosť obrazovky a operačný systém            | Použitelnosť              |
| 5  | Umožniť správu a zdieľanie pacientov v rámci aplikácie                                                  | Funkcionalita             |
| 6  | Umožniť import informácií o pacientoch z používaných medicínskych systémov                              | Funkcionalita             |
| 7  | Zvoliť vhodné technológie, ktoré majú v súčasnosti podporu a sú dobre zdokumentované                    | Technológie               |
| 8  | Odstrániť identifikované chyby vo funkčnosti                                                            | Údržba                    |
| 9  | Vytvoriť sériu testov komponentov                                                                       | Údržba,<br>Ladenie        |
| 10 | Odstrániť identifikované nepoužívané a neefektívne časti                                                | Údržba,<br>Efektivita     |
| 11 | Vytvoriť programátorskú dokumentáciu                                                                    | Údržba,<br>Rozšíriteľnosť |
| 12 | Navrhnuť rozšíriteľnú architektúru s vymeniteľnými a znovupoužiteľnými komponentami                     | Údržba,<br>Rozšíriteľnosť |
| 13 | Pri implementácii vychádzať z vytvoreného popisu algoritmu a nie neprehľadnej historickej implementácie | Údržba,<br>Rozšíriteľnosť |
| 14 | Využiť vhodných návrhových vzorov namiesto vzorov, ktoré sa na vývoj softvéru nemajú používať           | Údržba,<br>Rozšíriteľnosť |

Tabuľka 3.1: Úkony nutné na odstránenie nedostatkov identifikovaných v historickej implementácii Dithyron





---

# Analýza súčasnej webovej aplikácie Alma Diagnostics

Alma Diagnostics je webová aplikácia, ktorá vznikla, aby zastrešila a zmodernizovala CDSS systémy:

- BIANTA – diagnostika nádorov,
- CRACTES – diagnostika recidívy rakoviny,
- OSTEON – diagnostika porúch kostného metabolizmu,
- CYCLON – diagnostika endokrinných porúch v gynekológii,
- DITHYRON – diagnostika porúch štítnej žľazy.

V súčasnosti aplikácia ponúka funkcionality modulov BIANTA a CRACTES. Vytvorenie modulu pre DITHYRON je cieľom tejto diplomovej práce.

Táto kapitola analyzuje architektúru systému, skúma využité technológie, popisuje funkcionality systému, prezentuje proces vyšetrenia pacienta týmto systémom a popisuje jeho jednotlivé komponenty.

Na záver z týchto poznatkov a poznatkov získaných analýzou domény a historickej verzie programu DITHYRON formuluje požiadavky na nový modul.

## 4.1 Architektúra systému

Architektúra systému je klient-server, pričom Alma Diagnostics aplikácia je napísaná v jazyku *Python* vo frameworku *Flask* a nasadená na *Apache HTTP serveri*. Diagram nasadenia sa nachádza na obrázku 4.1.

Aplikácia na strane servera komunikuje s prehliadačom klienta pomocou protokolu HTTP (Hypertext Transfer Protocol) alebo HTTPS (HTTP secure) pre zabezpečenie šifrovaného spojenia na transportnej vrstve.

#### 4. ANALÝZA SÚČASNEJ WEBOVEJ APLIKÁCIE ALMA DIAGNOSTICS

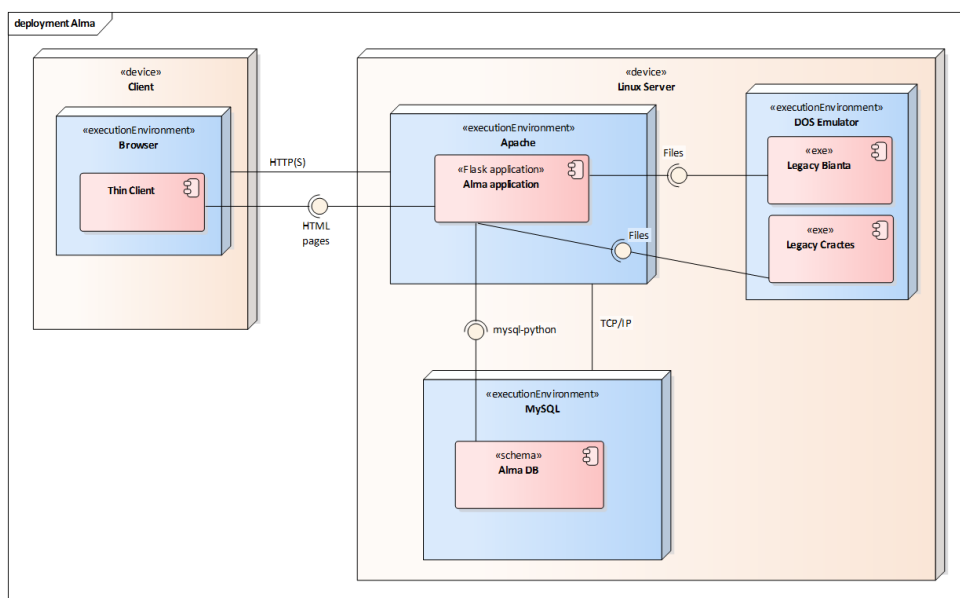
Odpovede pomocou dynamických webových stránok nevyžadujú implementáciu klientskej aplikácie. Je využitý tzv. tenký klient reprezentovaný internetovým prehliadačom, ktorý užívateľovi zobrazuje dynamické webové stránky obsahujúce formuláre, ktoré slúžia na odoslanie dát na server.

Na odosielanie dát na server je využitá metóda HTTP POST, na získavanie dát metóda HTTP GET. Iné HTTP metódy nie sú využité.

Jedná sa o stavový server (stateful), ktorý ukladá informácie o aktuálnom pripojení klienta medzi jednotlivými požiadavkami. Týka sa to informácii o autentifikácii a autorizácii užívateľa, zvolenom jazyku či objektoch, s ktorými užívateľ práve pracuje.

Na ukladanie užívateľov a ich pacientov sú využité tabuľky *MySQL* relačnej databázy.

Alma Diagnostics aplikácia slúži ako moderné rozhranie medzi užívateľom a aplikáciami *BIANTA* a *CRACTES*, ktoré sa spúšťajú v emulátore operačného systému *DOS*. Spracováva požiadavky od užívateľa, z ktorých vytvára vstupné súbory pre *BIANTU* a *CRACTES*. Z výstupných súborov vytvára odpovede v podobe HTML (Hypertext modeling language) stránok.



Obr. 4.1: Diagram nasadenia aktuálnej webovej aplikácie Alma Diagnostics

## 4.2 Technológie

Nasledujúce podkapitoly sú venované popisu použitých technológií.

### 4.2.1 Python

Python [50] je interpretovaný jazyk, ktorý umožňuje písať objektovo-orientovaný kód. Zároveň podporuje niektoré rysy typické pre funkcionálne programovanie ako sú anonymné lambda funkcie alebo funkcie prvého rádu (tj. funkcie môžu byť argumentom inej funkcie, môžu tvoriť návratovú hodnotu funkcie alebo môžu byť uložené do premennej).

Niektoré rysy jazyka Python sú:

- dynamické určovanie typov premenných za behu programu,
- automatické uvoľňovanie pamäte,
- využívanie dátových typov `list` (zoznam), `dict` (slovník) a `set` (množina) a podpora mnohých operácií na týchto typoch.

Aplikácia využíva verziu jazyka 2.7.

### 4.2.2 Flask

*Flask* je framework určený na tvorbu webových klient-server aplikácií v jazyku Python, pričom je založený na dvoch komponentoch *Werkzeug* a *Jinja2*, ktorých funkcionálnosť prepojuje [51],[52].

Jinja2 je šablónovací nástroj, ktorý je používaný na generovanie dynamických webových stránok. Časť prezentačnej logiky sú vďaka použitiu šablón odtienené od spracovania požiadavky. Šablóny okrem klasických HTML značiek, obsahujú konštrukty ako cykly a podmienky, ktoré umožňujú vytvoriť stránku dynamicky podľa typu a obsahu odpovede.

Werkzeug zabezpečuje aplikačnú časť rozhrania WSGI (Web Server Gateway Interface), ktoré slúži na nasadenie aplikácie na webový server (napríklad Apache). Werkzeug tiež obsahuje funkcie určené na smerovanie a správu požiadaviek, ladenie aplikácie a správu sieťového spojenia medzi klientom a serverom (session).

Informácie o session sú v rámci Flasku ukladané u klienta v podobe bezpečnej cookie, ktorú prehliadač užívateľa posiela serveru v hlavičke každej požiadavky.

Flask natívne neobsahuje ďalšie komponenty napríklad na prístup do databázy alebo prihlasovanie užívateľov, avšak tieto komponenty je možné pridať pomocou existujúcich rozšírení.

### 4.2.3 Apache HTTP Server

Apache HTTP Server je voľne dostupný webový server pre operačné systémy ako Linux a Windows [53]. Webový server spracováva prichádzajúce požiadavky cez HTTP protokol a odpovedá na ne.

Na nasadenie Python webovej aplikácie na Apache HTTP Server slúži *mod\_wsgi* balíček, ktorý poskytuje modul implementujúci serverovú časť rozhrania WSGI [54].

Apache HTTP server smeruje prichádzajúce požiadavky na jednotlivé URI (jednotný identifikátor prostriedku určený na pomenovanie zdroja – v tomto prípade webová adresa) do Flask aplikácie a stará sa o odosielanie odpovedí. Flask aplikácia rieši len samotné spracovanie požiadavky.

### 4.2.4 Databáza

Webová aplikácia používa na ukladanie dát relačnú SQL databázu, ktorá beží na MySQL databázovom serveri.

Na komunikáciu s databázou je použitá knižnica *SQLAlchemy* [55], ktorá komunikuje s databázovým serverom pomocou ovládača *MySQL-Python*.

*SQLAlchemy* umožňuje mapovanie databázových entít na objekty pomocou *object-relational mapper* (ORM). Súčasťou knižnice sú aj nástroje, ktoré zjednodušujú písanie dotazov a nevynucujú využívanie SQL jazyka (Structured Query Language).

### 4.2.5 Bootstrap

Na tvorbu vzhľadu generovaných webových stránok je použitý framework *Bootstrap 3* [56], ktorý ponúka nástroje na tvorbu responzívnych webových stránok optimalizovaných pre bežne používané prehliadače (napríklad Chrome, Firefox, Internet Explorer, Safari, ...) a menšie zariadenia ako mobily či tablety.

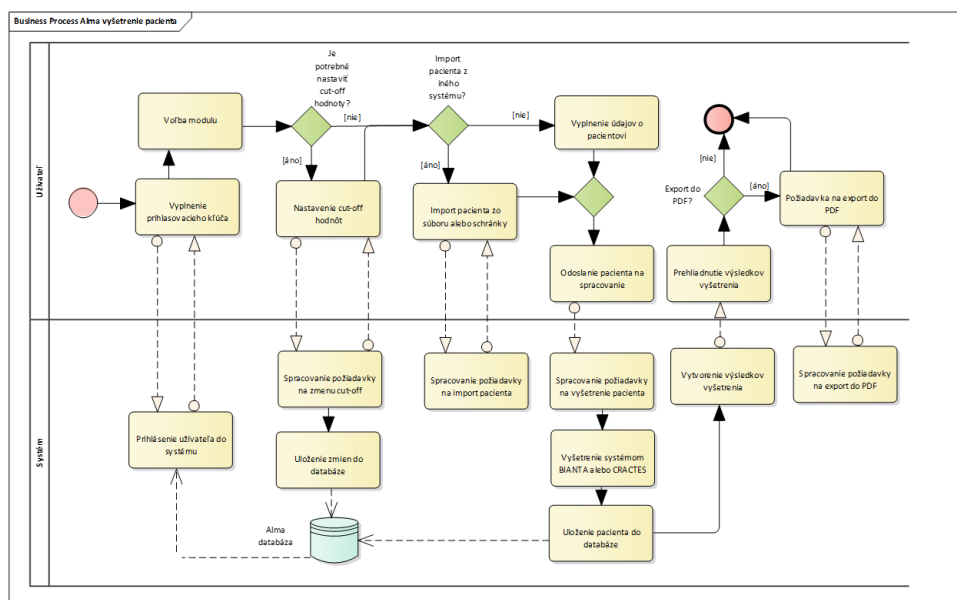
### 4.2.6 Rozšírenia

Aplikácia využíva aj ďalšie knižnice na rozšírenie funkcionality Flasku.

Na autentifikáciu užívateľov, prihlásenie a správu prihlásených užívateľov je použitá knižnica *Flask-Login* [57]. *Flask-Login* ukladá identifikátor prihláseného užívateľa do jeho session. Ďalej obsahuje `@login-required` dekorátor, ktorým je možné označiť URI, na ktoré nemá prístup neprihlásený užívateľ. Prístup neprihláseného užívateľa na takéto URI spôsobí presmerovanie na URI určené pre prihlásenie.

Na lokalizáciu aplikácie je použitá knižnica *Flask-Babel* [58]. *Flask-Babel* umožňuje definovanie *selector* funkcie, ktorá slúži na extrakciu jazyka z nastavení prihláseného užívateľa alebo hlavičky HTTP dotazu. Na preklad reťazcov je možné použitie funkcie `gettext(string)` alebo skrátene `_(string)` pre reťazce definované v rámci spracovania dotazu, zároveň je možné použiť funkciu `lazy_gettext(string)` pre reťazce definované na iných miestach v programe, ktorých preklad sa vykoná až v rámci spracovania dotazu. Takto označené reťazce sa ukladajú do .pot súboru, ktorý je šablónou pre .po súbory obsahujúce preklad do želaných jazykov.

### 4.3 Funkcionalita Alma Diagnostics aplikácie



Obr. 4.2: Procesný diagram vyšetrenia pacienta pomocou webovej aplikácie Alma Diagnostics

Hlavná funkcionality Alma Diagnostics aplikácie je:

- prihlásenie do systému unikátnym kľúčom,
- diagnostika pacientov pomocou modulov BIANTA a CRACATES,
- import údajov o pacientoch zo súboru alebo schránky (podpora pre niektoré medicínske systémy),
- export výsledkov vyšetrení do PDF,
- doporučovanie markerov podľa veku a pohlavia alebo predchádzajúcej diagnózy,
- voliteľné zadávanie osobných údajov pacienta a ich šifrovanie,
- správa predchádzajúcich vyšetrení pacientov,
- zdieľanie vyšetrení v rámci skupiny,
- nastavenie cut-off hodnôt podľa laboratórneho pracoviska,
- lokalizácia do češtiny, slovenčiny a angličtiny s možnou podporou ďalších jazykov,

- podpora pre zariadenia s variabilnou veľkosťou obrazovky.

Procesný diagram vyšetrovania pacienta vrátane prihlásenia a možnosti nastaviť cut-off hodnoty, importovať údaje o pacientovi a exportovať výsledky do PDF sa nachádza na obrázku 4.2.

## 4.4 Komponenty Alma Diagnostics aplikácie

Vnútrotná architektúra Alma Diagnostics aplikácie je vytvorené podľa vzoru Model-View-Controller (MVC).

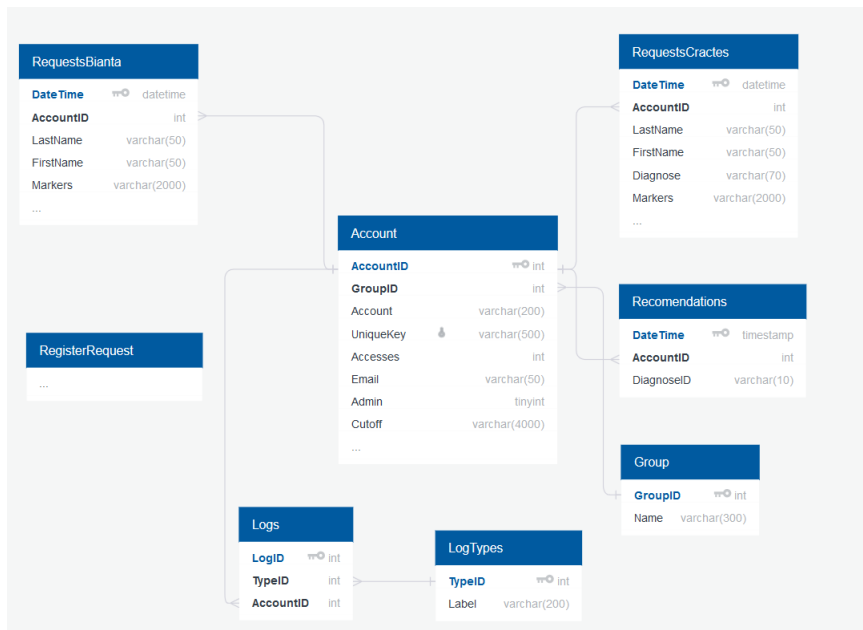
- **Model** – obsahuje dátový model aplikácie a obsluhuje jej logiku. Obsahuje napríklad moduly slúžiace na komunikáciu s DOS aplikáciami a databázou, šifrovanie a tvorbu PDF súborov.
- **View** – je reprezentovaný Jinja2 šablónami, ktoré tvoria dynamické webové stránky zobrazované prehliadačom užívateľa, .css súbormi na definovanie vzhľadu stránok a skriptami v jazyku JavaScript na dynamické chovanie vzhľadu a validáciu vstupov u klienta (ukážka užívateľského rozhrania sa nachádza na obrázku 4.4).
- **Controller** – je reprezentovaný funkciami, pričom každá funkcia spracováva požiadavky od užívateľov smerujúce na jej pridelenú adresu URI. Informácie získané z požiadavky predáva modelu. Zároveň slúži na generovanie dynamickej webovej stránky, ktoré je užívateľovi vrátená ako odpoveď alebo presmerovanie na inú adresu URI (viď ukážku kódu 4.1)

```
1 @app.route('/nejake_uri', methods=['GET', 'POST']) # URI
2 @login_required # len pre prihlásených
3 def nejake_uri():
4     uri = 'nejake uri'
5
6     # Spracovanie požiadavky
7     if request.method == 'POST':
8         # volanie metód modelu, databáza atď...
9         ...
10
11     # Generovanie View
12     return render_template(uri + '.html', title='Nejake Uri', uri=uri,
13                             menu=menu, objects=objects, session=session, locale=get_locale())
```

Ukážka zdrojového kódu 4.1: Funkcia starajúca sa o spracovanie požiadavky

Databázový model aplikácie (viď obrázok 4.3) je vystavaný okolo entity užívateľského účtu uloženej v tabulke **Account**. Účet má priradený unikátny kľúč určený na prihlasovanie. Ku každému účtu sú priradené ďalšie informácie ako názov, práva, počet zakúpených dotazov a laboratórne cut-off hodnoty.

Každý užívateľ má priradenú skupinu **Group**, do ktorej patrí.



Obr. 4.3: Databázový model webovej aplikácie Alma Diagnostics

Ďalej sú pre každého užívateľa ukladané logy o jeho činnosti v tabuľke **Logs**.

Dôležitými entitami sú dotazy na aplikácie BIANITA a CRACTES, ktoré sa ukladajú do tabuliek **RequestsBianta** a **RequestsCractes**. Dotaz tvoria informácie o vyšetrovanom pacientovi ako sú jeho meno, priezvisko, rok narodenia, pohlavie a výsledky laboratórnych meraní.

Poslednou entitou previazanou s užívateľským účtom sú vykonané odporúčenia na zmeranie markerov na základe diagnózy, ktoré sú ukladané do tabuľky **Recommendation**.

Databáza navyše obsahuje tabuľku na požiadavky o registráciu.

## 4.5 Analýza požiadaviek na modul DITHYRON

Táto podkapitola najskôr rozoberá technologické možnosti zakomponovania modulu do existujúcej webovej aplikácie.

Potom pomocou znalostí získaných z analýzy domény (kapitola 1), nedostatkov historickej implementácie (kapitola 3) a analýzy architektúry a funkcionality súčasnej webovej aplikácie formuluje funkčné a nefunkčné požiadavky na modul.

Na záver vymenováva prípady použitia modulu.

Vzhľadom na to, že procesy používania nového modulu vychádzajú z procesov používania existujúcich modulov Alma Diagnostics aplikácie, je možné

## 4. ANALÝZA SÚČASNEJ WEBOVEJ APLIKÁCIE ALMA DIAGNOSTICS

The screenshot shows the 'Vytvoriť nového pacienta' (Create new patient) form in the Alma Diagnostics web application. The interface includes a navigation bar with 'Domov', 'Blanta', 'Odporučíť', 'Vytvoriť pacienta', and 'Zobraziť výsledky'. The user is identified as 'Petra Krnáčová'. The form fields are:

- Priezvisko:** Priezvisko (voliteľné)
- Krstné meno:** Krstné meno (voliteľné)
- ID:** ID (voliteľné, maximálne 11 znakov)
- Pohlavie:** Muž / Žena
- Rok narodenia:** 1954
- Poznámky:** Poznámka (voliteľné)

Below the form, a message states: 'Pre vek a pohlavie pacienta je vhodné vziať do úvahy tieto ukazovatele:'. It lists 'Typ' (Všetko) and 'Hlavné markery' (PSA (PSA, proPSA), CEA, CYFRA 21-1, CA72-4) and 'Ďalšie markery' (B2M, CA19-9, NSE, AFP, SCCA).

| Marker | Dátum      | Hodnota |
|--------|------------|---------|
| CA195  | 27.04.2018 | 3,03    |
| FPSA   | 27.04.2018 | 2,03    |
| CEA    | 27.04.2018 | 4,03    |

Buttons: Pridať +3, Odobrať, Potvrdiť pacienta

Obr. 4.4: Ukážka užívateľského rozhrania webovej aplikácie Alma Diagnostics [59]

procesný diagram 4.2 aplikovať aj na nový modul a neuvádzať duplicitné procesné diagramy.

### 4.5.1 Technologické riešenie

Alma Diagnostics aplikácia rieši nedostatky z pohľadu užívateľa, ktoré obsahovali historické programy BIANITA a CRACTES a boli zistené aj v prípade historického programu DITHYRON. Komunikácia so zastaranými programami pomocou súborov však nerieši nedostatky týkajúce sa rozširiteľnosti a údržby programu. Preto pre algoritmus DITHYRON musí byť navrhnutý nový modul, ktorý týmito nedostatkami netrpí. V budúcnosti budú navrhnuté a implementované nové moduly aj pre BIANITU, CRACTES a ďalšie algoritmy, ktoré je nutné revitalizovať.

Algoritmus na detekciu porúch štítnej žľazy popísaný v kapitole 2, je možné zakomponovať priamo ako modul do Alma Diagnostics aplikácie alebo oddeliť do samostatnej aplikácie komunikujúcej s Alma Diagnostics aplikáciou pomocou zvoleného rozhrania. V takomto prípade by Alma Diagnostics aplikácia zastrešovala komunikáciu s klientom a nová samostatná aplikácia funkcionality algoritmu.



Výhodou tohto riešenia je možnosť voľby akéhokoľvek programovacieho jazyka a technológií na implementáciu algoritmu. Zároveň by bolo možné navrhnuť rozhranie, ktoré by nebolo previazané s webovým prehliadačom a algoritmus by mohol byť poskytovaný ako služba iným systémom.

Voľba ďalších technológií a iného programovacieho jazyka však prináša zbytočnú komplexnosť a zvyšuje nároky na údržbu systému. Zároveň paralelne s touto diplomovou prácou vzniká REST (Representational state transfer) rozhranie pre Alma Diagnostics moduly, ktoré využíva na komunikáciu JSON (JavaScript Object Notation) objektov. Toto rozhranie bude využité na komunikáciu s inými typmi klientov a môže byť rozšírené aj na modul DITHYRON.

Z týchto dôvodov je vhodnejšie vytvoriť modul pre algoritmus ako súčasť aplikácie Alma Diagnostics.

### 4.5.2 Funkčné a nefunkčné požiadavky

#### Funkčné požiadavky:

- FP1 – určovanie diagnóz pacientov pomocou ich laboratórnych výsledkov, veku a pohlavia.
- FP2 – zhodnotenie laboratórnych výsledkov.
- FP3 – zobrazovanie chovania markerov pre diagnózy.
- FP4 – správa vyšetrení pacientov.
- FP5 – zdieľanie vyšetrení pacientov v rámci skupiny.
- FP6 – možnosť nastavenia referenčných intervalov.
- FP7 – podpora pre import údajov o pacientovi zo súboru a schránky.
- FP8 – podpora pre export výsledkov vyšetrení do PDF.
- FP9 – možnosť vynechania alebo šifrovania osobných údajov pacienta.
- FP10 – lokalizácia modulu do slovenčiny a angličtiny (a v budúcnosti do češtiny a ďalších jazykov).
- FP11 – možnosť konfigurácie modulu.

#### Nefunkčné požiadavky:

- NP1 – využitie technológií daných Alma Diagnostics aplikáciou (Python, Flask, Jinja2, Bootstrap, MySQL, Flask-Babel).
- NP2 – vzhľad a používanie užívateľského rozhrania v súlade s existujúcim GUI Almy.

#### 4. ANALÝZA SÚČASNEJ WEBOVEJ APLIKÁCIE ALMA DIAGNOSTICS

- NP3 – využitie existujúcej funkcionality Almy (napríklad šifrovanie, niektoré komponenty rozhrania ...).
- NP4 – architektúra modulu algoritmu, ktorá umožňuje nahraditeľnosť a rozširiteľnosť jednotlivých komponentov a ich znovupoužitie v iných algoritmoch.
- NP5 – čitateľný a komentovaný zdrojový kód.
- NP6 – implementácia neobsahujúca nedostatky zistené u historickej verzii algoritmu.

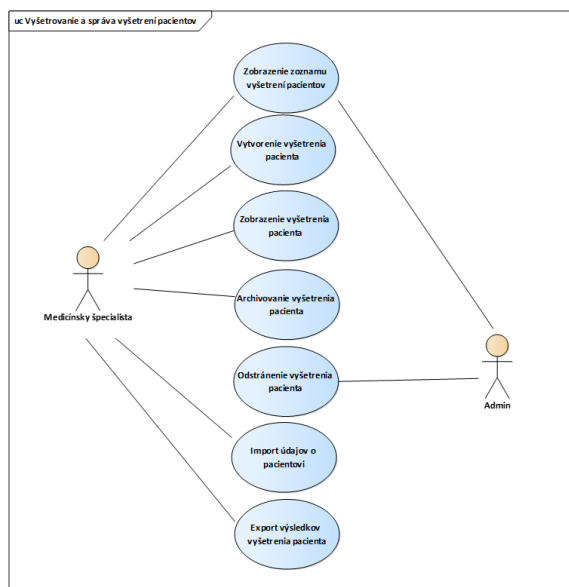
##### 4.5.3 Prípady použitia

Modul môžu používať dva typy užívateľov **Admin** (skratka pre administrátor) a **Medicínsky špecialista**.

Prípady použitia je možné rozdeliť do troch kategórií: vyšetovania a správa pacientov, konfigurácia algoritmu a užívateľské nastavenia.

###### 4.5.3.1 Vyšetovanie a správa pacientov

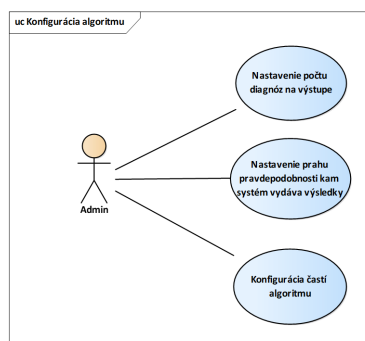
Prípady použitia týkajúce sa vyšetovania a správy pacientov sa nachádzajú na obrázku 4.5. Špecialista môže archivovať vyšetrenia pacientov, čo znamená, že sa mu už nebudú zobrazovať v zozname vyšetrení. Nevratne mazať vyšetrenia z databázy však môže len administrátor. V prípade, že špecialista je v skupine s inými špecialistami, môže si prezeráť výsledky vyšetrení ich pacientov.



Obr. 4.5: Prípady použitia: Vyšetovanie a správa pacientov

### 4.5.3.2 Konfigurácia algoritmu

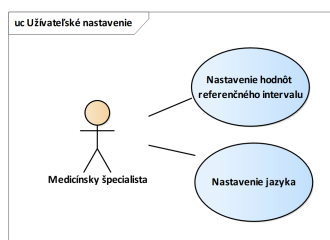
Prípady použitia týkajúce sa konfigurácie systému sa nachádzajú na obrázku 4.6. Administrátor môže konfigurovať výstup zo systému do akého prahu pravdepodobnosti bude systém vydávať diagnózy a aký je ich maximálny možný počet na výstupe. Zároveň môže nastaviť chovanie a typ algoritmu.



Obr. 4.6: Prípady použitia: Konfigurácia algoritmu

### 4.5.3.3 Uživatelské nastavenie

Prípady použitia týkajúce sa užívateľského nastavenia sa nachádzajú na obrázku 4.7. Špecialista môže nastaviť hodnoty referenčných intervalov na hodnoty použitého laboratória. Tieto hodnoty systém použije v rámci vyhodnocovania koncentracii markerov. Tiež si môže nastaviť jazyk užívateľského rozhrania a výstupu.



Obr. 4.7: Prípady použitia: Užívateľské nastavenie



---

## Návrh modulu

Táto kapitola sa venuje návrhu modulu na diagnostiku porúch štítnej žľazy. Na úvod sa venuje zapojeniu algoritmu do celkovej architektúry aplikácie a potom rozoberá návrh jeho jednotlivých častí.

### 5.1 Zapojenie algoritmu do celkovej architektúry aplikácie

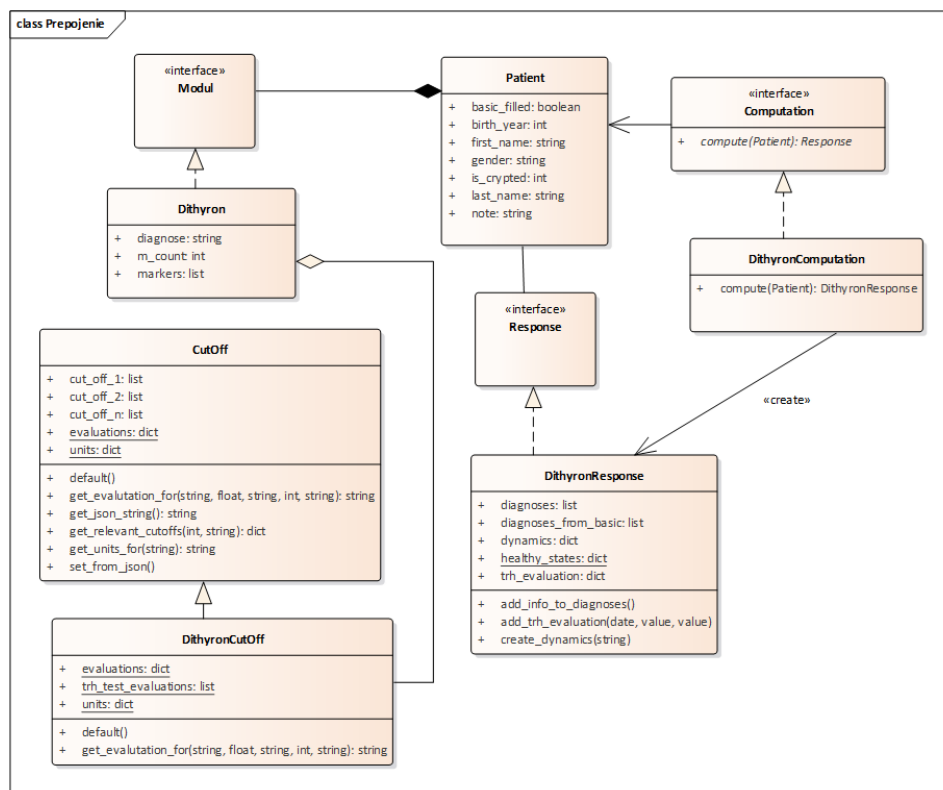
Modul Dithyron bude rešpektovať MVC architektúru danú aplikáciou. Algoritmus výpočtu patrí do časti `Model`. Je však nutné definovať akým spôsobom bude komunikovať zvyšok aplikácie s algoritmom. Zároveň je nutné dodržať požiadavku na rozširiteľnosť a ľahkú zámenu algoritmu za iný.

Na tieto účely bude slúžiť interface `Computation`, ktorý bude vyžadovať jedinú metódu `compute`, ktorej vstupným parametrom je inštancia triedy `Patient` a výstupom je inštancia odpovede algoritmu `Response`, kde objekt `Patient` udržiava užívateľom zadané informácie o vyšetrovanom pacientovi a objekt `Response` obsahuje výsledky vyšetrenia, ktoré sú prezentované užívateľovi.

Tento interface bude možné použiť pre akýkoľvek výpočtový algoritmus, ktorý dodrží obmedzenia dané vstupným a výstupným objektom.

Pre algoritmus popísaný v kapitole 2 bude vytvorená trieda `DithyronComputation` realizujúca interface `Computation`. Trieda bude využívať návrhový vzor *Facade*. Cieľom vzoru *Facade* je zjednodušiť rozhranie systému a obaliť jeho jednotlivé komponenty [60]. Návrhu jednotlivých komponentov sa venuje podkapitola 5.2.1. Zodpovednosťou triedy `DithyronComputation` bude poskytnúť komponentom potrebné údaje z objektu `Patient`, zabezpečiť ich prepojenie podľa procesného diagramu definovaného v kapitole 2 (obrázok 2.5) a vytvorí objekt odpovede `DithyronResponse`, ktorý budú môcť funkcie `Controllera` ďalej spracovávať. Diagram tried zobrazujúci toto prepojenie sa nachádza na obrázku 5.1.

## 5. NÁVRH MODULU



Obr. 5.1: Diagram tried: Prepojenie triedy obalujúcej algoritmus s ostatnými časťami systému, diagram pre zjednodušenie neobsahuje všetky atribúty a metódy tried

**Patient** je objekt, ktorý aplikácia uchováva v session v rámci jednotlivých dotazov užívateľa. Objekt **Patient** zostáva uložený v session objekte na serveri dokým ho užívateľ neodošle na vyšetrenie (alebo vyšetrenie zruší, prípadne mu vyprší čas). Objekt **Patient** obsahuje údaje o pacientovi a v priradenom objekte **Module** obsahuje informácie týkajúce sa vyšetrenia daným modulom.

V prípade modulu **Dithyron** sú nutné laboratórne merania **markers**, ich počet **m\_count**, predchádzajúca diagnóza pacienta **diagnose** a cut-off hodnoty **DithyronCutOff** priradené k užívateľovi.

**DithyronCutOff** bude rozširovať existujúcu triedu **CutOff**, ktorá obsahuje pre každý marker priradené cut-off hodnoty. Cut-off hodnoty pre marker sú uložené v zozname **list**, pričom obsahom zoznamu môžu byť rovno samotné cut-off pokiaľ sú rovnaké pre rozličné pohlavia a vekové kategórie. V prípade, že je nutné rozlišovať vek a pohlavie je položkou zoznamu slovník **dict**, s kľúčmi pre dané pohlavie a vekové rozmedzie.

Trieda **CutOff** obsahuje metódy na získanie cut-off pre daný vek a pohlavie

vie `get_relevant_cutoffs`, serializáciu do a deserializáciu z objektu JSON, slovné vyhodnotenie hodnoty markeru vzhľadom na cut-off hodnoty `get_evaluation_for` a ďalšie metódy na operácie s cut-off hodnotami.

Trieda `DithyronCutOff` bude preberať funkcionality rodičovskej triedy pričom metódu `get_evaluation_for` rozšíri o vyhodnotenie TRH testu, ktorý má iný popis. Zároveň redefinuje metódu `default`, ktorá nastaví cut-off hodnoty na východzie definované autorom algoritmu.

Konstruktore `DithyronResponse` bude nutné poskytnúť zoznam diagnóz obsahujúci slovníkové entity s položkami `code`, kde bude uložený unikátny kód diagnózy a `probability`, kde bude uložená pravdepodobnosť diagnózy. Trieda `DithyronResponse` bude obsahovať metódu `add_info_to_diagnoses`, ktorá k diagnózam priradí ich názov a poznámku. Ďalej bude obsahovať metódu `create_dynamics`, slúžiacu na vytvorenie slovníka obsahujúceho vývoj jednotlivých meraní v čase a zhodnotenie ich koncentrácií a metódu `add_TRH_evaluation`, ktorá vytvorí `list` s vyhodnotením TRH testu, obsahujúci dátum, hodnotu bazálneho TSH, hodnotu stimulovaného TSH a slovné zhodnotenie. Zároveň bude mať triedny atribút `healthy_states`, ktorý bude použitý na výpis výsledkov ak pacient netrpí žiadnou diagnózou. Tieto údaje budú využité na prezentáciu výsledkov užívateľovi.

## 5.2 Návrh častí

Nasledujúce podkapitoly obsahujú návrh častí modulu, pričom je kladený dôraz na návrh komponentov algoritmu.

### 5.2.1 Komponenty algoritmu

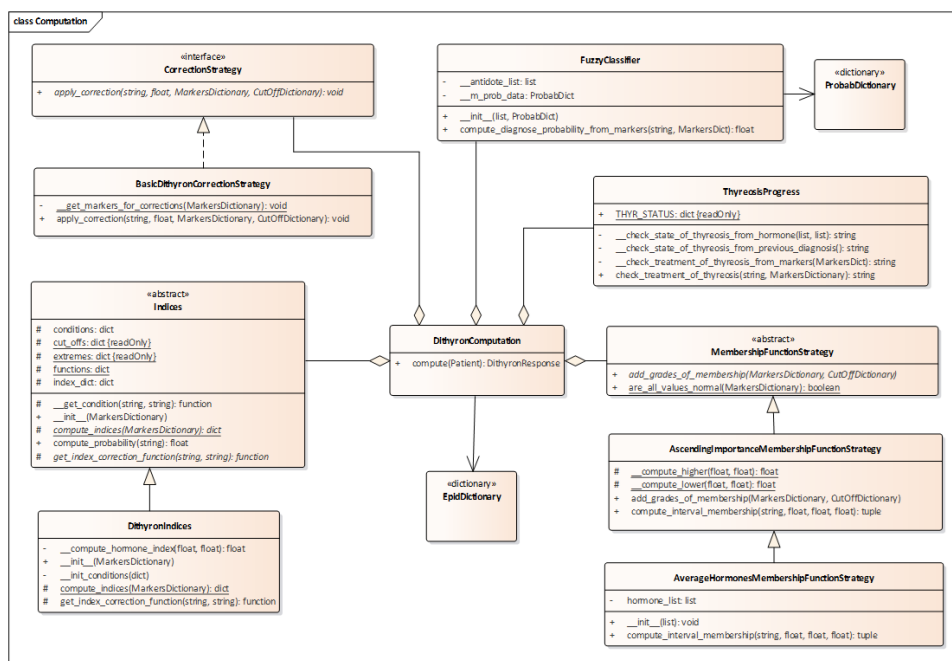
Komponenty algoritmu budú vychádzať z rozdelení algoritmu na časti v kapitole 2, pričom bude použitý objektovo-orientovaný prístup. Na modelovanie znalostí a dátových objektov, ktoré nebudú mať žiadne metódy budú použité dátové typy zoznam `list` a slovník `dict`.

Tieto dátové typy sú vhodné aj na modelovanie komplexnejších entít, pretože v jazyku Python nie je obmedzenie na ich obsah. Obsahom `list` a `dict` môžu byť rôzne dátové typy a dokonca aj funkcie.

Funkcionalita algoritmu bude implementovaná presne podľa vzniknutého popisu, preto sa táto kapitola bude venovať len objektovému návrhu a nebude zachádzať do detailov fungovania, ktoré už boli raz popísané.

Diagram tried jednotlivých častí, ktoré zastrešuje trieda `DithyronComputation` sa nachádza na obrázku 5.2.

## 5. NÁVRH MODULU



Obr. 5.2: Diagram tried: Komponenty algoritmu

### 5.2.1.1 Znalosti

Najdôležitejšími znalosťami, ktoré algoritmus používa sú podmienené pravdepodobnosti pre fuzzy naivný bayesovský klasifikátor a prevalencie diagnóz.

Tieto znalosti budú uložené v súbore, ktorý by mal byť čitateľný medicínskym špecialistom, ktorý sa bude starať o správu znalostí. Nebude sa teda jednať o súbor vo formáte JSON alebo XML, ale obyčajný textový súbor.

Formát súboru pre podmienené pravdepodobnosti bude nasledovný:

Podmienené pravdepodobnosti

```
diag1 TSH 0.0000 0.0050 0.9900 0.0050 0.0000
diag1 TT4 0.0002 0.0198 0.9600 0.0198 0.0002
...
diag40 rT3 0.0050 0.0400 0.9100 0.0400 0.0050
diag40 rT4 0.0050 0.0400 0.9100 0.0400 0.0050
```

V riadku sa bude nachádzať kód diagnózy, kód markeru, podmienené pravdepodobnosti, že pri danej diagnóze je koncentrácia markeru veľmi znížená, znížená, normálna, zvýšená a veľmi zvýšená.



Formát súboru pre prevalencie diagnóz bude nasledovný:

Prevalencie

```
diag1 Female 10.0000 10.0000 20.0000 20.0000 25.0000 25.0000 50.0000 100.0000
diag1 Male 10.0000 10.0000 10.0000 10.0000 10.0000 10.0000 10.0000 10.0000
...
diag40 Female 0.100 0.200 0.0500 0.0500 0.0500 0.0500 0.0500 0.0500
diag40 Male 0.100 0.200 0.0500 0.0500 0.0500 0.0500 0.0500 0.0500
```

V riadku sa bude nachádzať kód diagnózy, pohlavie a prevalencie (na 100 000 obyvateľov) pre vekové intervaly po 10 rokoch (0 – 10, 10 – 19 atď).

Znalosti budú modulom načítané do slovníkov a tieto budú využívané jednotlivými časťami výpočtu. Ukážka kódu 5.1 zobrazuje ako bude vyzeráť slovník pre podmienené pravdepodobnosti. Obdobne bude prevedený aj súbor s prevalenciami.

```
1 DITHYRON_PROB_DATA = {
2   'diag1': {
3     'TSH': [0, 0.005, 0.99, 0.005, 0],
4     'TT4': [0.0002, 0.0198, 0.96, 0.0198, 0.0002]
5     ...
6   },
7   ...
8   'diag40': {
9     ...
10    'rT3': [0.005, 0.04, 0.91, 0.04, 0.005],
11    'rT4': [0.005, 0.04, 0.91, 0.04, 0.005]
12  }
13 }
```

Ukážka zdrojového kódu 5.1: Návrh formátu podmienených pravdepodobností

Ostatné menšie znalosti vzťahujúce sa k jednotlivým komponentom budú tvoriť triedne atribúty tried realizujúcich tieto komponenty.

Zároveň bude implementácia obsahovať enumerácie všetkých používaných kódov diagnóz a markerov.

### 5.2.1.2 Výmenné formáty

Výsledky laboratórnych vyšetrení využívajú všetky časti algoritmu, je teda nutné definovať ako budú vyzeráť. Vzhľadom na to, že k samotným výsledkom sa priamo neviažu žiadne metódy a je nutné ich postupne obohacovať o vyhodnotenia bude na ich reprezentáciu použitý slovník, ktorého formát je zobrazený na ukážke kódu 5.2.

Kľúčmi slovníka budú jednotlivé názvy markerov. Každý marker bude mať priradený zoznam meraní, pričom meranie bude musieť povinne obsahovať dátum a hodnotu. Výhodou slovníkov je, že majú dynamickú štruktúru. Každé

meranie bude teda možné obohatiť o ďalšie atribúty, napríklad stupne príslušnosti do fuzzy množín, pričom jednotlivé časti algoritmu budú môcť pracovať len s atribútami, ktoré práve potrebujú.

```

1 {
2   'TT4': [{ 'date': 1.2.2016, 'value': 13.5},
3           { 'date': 15.6.2017, 'value': 11.7}]
4   'TT3': [{ 'date': 1.2.2016, 'value': 0.78}]
5 }
```

Ukážka zdrojového kódu 5.2: Návrh výmenného formátu meraní markerov

Niektoré časti budú pracovať aj s cut-off hodnotami. Tie budú získané z objektu `DithyronCutOff` pomocou zdedenej metódy `get_relevant_cutoffs` zadáním veku a pohlavia pacienta. Formát výstupu z metódy sa nachádza na ukážke kódu 5.3.

```

1 {
2   'TT4': [0.5, 2, 10, 15]
3   'TT3': [2, 5, 15.8, 20.7]
4   ...
5 }
```

Ukážka zdrojového kódu 5.3: Výstup metódy `get_relevant_cutoffs`

### 5.2.1.3 Výpočet stupňov príslušnosti

Vzhľadom na to, že funkcie, ktoré budú použité na výpočet stupňov príslušnosti meraní markerov do fuzzy množín sa môžu rôzniť, je vhodné tento výpočet oddeliť od fuzzy naivného bayesovského klasifikátora a navrhnúť podľa návrhového vzoru *strategy*. Strategy je návrhový vzor, ktorý umožňuje definovať rodinu algoritmov, ktoré budú mať rovnaký výstup a budú implementovať rovnaké rozhranie [60]. Jednotlivé algoritmy je možné ľubovoľne zamieňať bez toho, aby to vynucovalo nejaké zmeny v klientskej triede využívajúcej algoritmus.

Rozhranie pre výpočet stupňov príslušnosti bude realizované abstraktnou triedou `MembershipFunctionStrategy`, ktorá bude obsahovať dve metódy:

- `add_grades_of_membership` – abstraktá metóda, ktorú jednotlivé stratégie budú musieť implementovať. Vstup do metódy bude slovník meraní markerov a cut-off hodnoty pre dané pohlavie a vek pacienta. Zodpovednosť metódy bude doplniť ku jednotlivým meraniam ich stupne príslušnosti do fuzzy množín.
- `are_all_values_normal` – statická metóda, ktorá z meraní, na ktoré bola aplikovaná metóda `add_grades_of_membership` určí, či všetky merania majú normálnu koncentráciu. Toto určenie sa bude využívať na zjednodušenie výpočtu. V prípade normálnych hodnôt bude možné preskočiť ďalšie časti výpočtu.

`AscendingImportanceMembershipFunctionStrategy` bude obsahovať výpočet stupňov príslušnosti, tak ako boli popísané v časti 2.2.1. Na lepšie členenie výpočtu a obmedzenie DRY bude mať aj nasledujúce metódy:

- `_compute_lower` – privátna statická metóda, ktorá na základe hodnoty merania a príslušnej cut-off hodnoty vypočíta stupeň príslušnosti do intervalu  $I - 1$ .
- `_compute_higher` – to isté pre interval  $I + 1$ .
- `compute_interval_membership` – pre jedno meranie určí jeho stupne príslušnosti za použitia príslušných cut-off hodnôt.

V časti 2.2.1 bolo uvedené, že tento výpočet stupňov príslušnosti vznikol pre nádorové markery, kde iba zvýšená hodnota spôsobuje zdravotné problémy.

Vzhľadom na to, že pre hormóny je nebezpečná aj znížená hodnota bude implementovaná aj druhá stratégia `AverageHormonesMembershipFunctionStrategy`, ktorá rozšíri predchádzajúcu. Túto stratégiu bude nutné inicializovať zoznamom názvov hormónov `hormone_list`. V prípade, že sa bude počítať marker, ktorý nie je hormón, sa použijú metódy rodiča. V prípade hormónu budú využité upravené funkcie príslušnosti, ktoré nebudú brať spodnú a vrchnú cut-off zvlášť, ale budú brať ich aritmetický priemer.

#### 5.2.1.4 Naivný fuzzy bayesovský klasifikátor

Funkcionalita bude implementovaná v triede `FuzzyClassifier`. Inštanciu klasifikátora bude nutné inicializovať slovníkom podmienených pravdepodobností a zoznamom názvov markerov, na ktoré sa má aplikovať logaritmická korekcia (viď popis v časti 2.2.1).

Metóda `compute_diagnose_probability_from_markers` bude určená na výpočet pravdepodobnosti diagnózy a bude vyžadovať jej názov a slovník meraní markerov s priradenými stupňami príslušnosti niektorou zo stratégií.

#### 5.2.1.5 Epidemiológia

Nakoľko aplikácia prevalencií je priamočiara, nebude na tieto účely využitá žiadna ďalšia trieda a na pravdepodobnosť diagnózy sa priamo aplikuje hodnota zo slovníka prevalencií.

#### 5.2.1.6 Indexy

Funkcionalita indexov bude implementovaná pomocou návrhového vzoru *Template Method*. `Template Method` [60] je návrhový vzor, pri ktorom abstraktná trieda definuje kostru algoritmu. Časti algoritmu, ktoré sa môžu meniť sú reprezentované abstraktnými metódami a sú implementované potomkami triedy.

Abstraktná trieda `Indices` bude obsahovať nasledujúce atribúty a metódy:

- `index_dict` – slovník vypočítaných hodnôt indexov.
- `conditions` – slovník podmienok vo forme anonymných funkcií, ktoré musia byť splnené pred aplikáciu niektorého indexu.
- `cut_offs` – slovník cut-off hodnôt pre indexy a diagnózy.
- `extremes` – slovník expertných odhadov vplyvu hodnoty indexu na diagnózu.
- `functions` – lineárne lomené funkcie určené na výpočet pravdepodobností diagnóz dynamicky vytvorené z `cut_offs` a `extremes`.
- `compute_indices` – abstraktná metóda, ktorá inicializuje `index_dict` z meraní markerov.
- `get_index_correction_function` – abstraktná metóda, ktorej vstupom je názov indexu a diagnózy a výstupom časť lomenej funkcie, ktorá sa má aplikovať na pravdepodobnosť.
- `compute_probability` – metóda na výpočet pravdepodobnosti diagnóz pomocou indexov. Vstupom bude kód diagnózy. Metóda skontroluje podmienky aplikácie, na hodnoty indexov aplikuje lineárne lomené funkcie a podľa výstupov vyčíslí pravdepodobnosť diagnózy z indexov.

`DithyronIndices` bude realizáciou tejto abstraktnej triedy. Do atribútov `cut_offs`, `extremes` a `conditions` budú vložené znalosti popísané v časti 2.2.3. Metóda `compute_indices` bude slúžiť na výpočet indexov tiež popísaných v časti 2.2.3.

### 5.2.1.7 Korekcie

Korekcie budú podobne ako funkcie na výpočet stupňov príslušnosti implementované pomocou návrhového vzoru Strategy, pretože sa v budúcnosti môžu často meniť podľa znalostí získaných používaním algoritmu.

`CorrectionStrategy` bude rozhranie obsahujúce deklaráciu jedinej metódy `apply_corrections`, ktorá na vstupnú pravdepodobnosť diagnózy aplikuje korekcie. Ako budú korekcie vyzeráť a akým spôsobom budú aplikované závisí na triede realizujúcej rozhranie.

Realizácia `BasicDithyronCorrectionStrategy` bude obsahovať korekcie vo forme `if-then` pravidiel ako boli popísané v časti 2.2.4.

### 5.2.1.8 Zhodnotenie progresu ochorenia

`ThyreosisProgres` bude trieda starajúca sa o zhodnotenie progresu ochorenia (viď časť 2.2.5). Bude mať triedny atribút `THYR_STATUS` v podobe slovníka, v ktorom budú uložené znalosti potrebné na zhodnotenie progresu.

Metóda `check_treatment_of_thyreosis` bude slúžiť na zhodnotenie progresu. Jej vstupom bude voliteľná predchádzajúca diagnóza a slovník meraní markerov. Táto metóda bude volať dve privátne metódy:

- `_check_state_of_thyreosis_from_previous_diagnosis` – zhodnotenie progresu predchádzajúcej diagnózy z najnovších meraní.
- `_check_treatment_of_thyreosis_from_markers` – zhodnotenie progresu ochorenia z najstarších a najnovších meraní markerov, pričom pôvodné ochorenie bude vydedukované z chovania markerov.

### 5.2.1.9 DithyronComputation

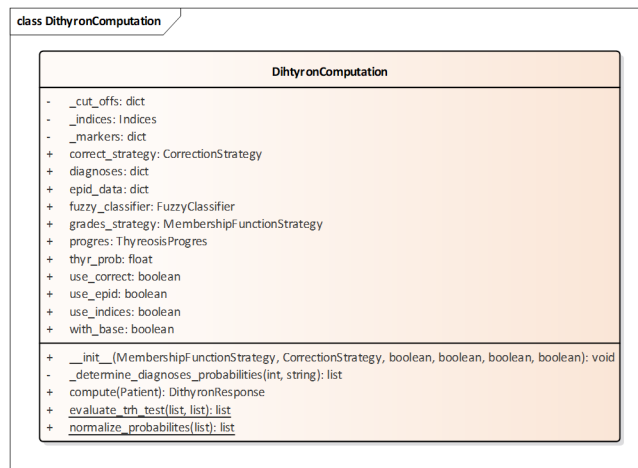
V predchádzajúcich častiach bolo prezentované, že trieda `DithyronComputation` bude slúžiť na prepojenie jednotlivých komponentov výpočtu, nebolo však popísané akým spôsobom bude možné inštancie tejto triedy konfigurovať, nakoľko možnosť konfigurácie je jednou z funkčných požiadaviek.

Inštancie triedy budú nastaviteľné pomocou nasledujúcich parametrov:

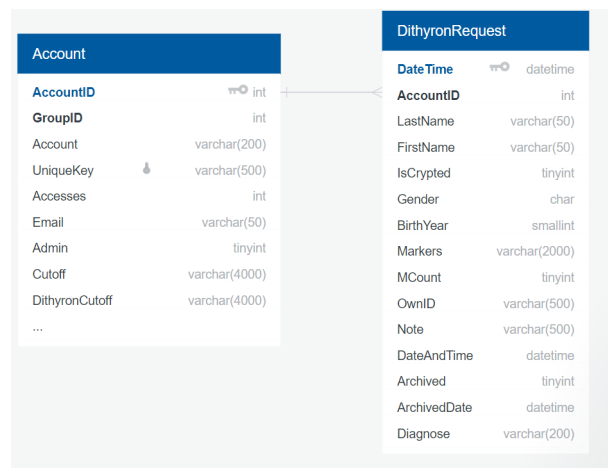
- `use_epid=true` – pri výpočte pravdepodobností sa použijú prevalencie diagnóz.
- `use_indices=true` – pri výpočte pravdepodobností sa použijú indexy.
- `use_correction=true` – pri výpočte pravdepodobností sa použijú korekcie.
- `correct_strategy` – aká `CorrectionStrategy` sa použije, prípadne `None` ak sa korekcie nemajú použiť.
- `grades_strategy` – aká `MembershipFunctionStrategy` sa použije.
- `with_base=true` – do `DithyronResponse` sa uložia dva zoznamy pravdepodobností, jeden s pravdepodobnosťami vyčíslenými aplikovaním všetkých zvolených častí a druhý, kde budú pravdepodobnosti len z `FuzzyClassifier`.

Administrátor bude schopný vypnúť všetky časti algoritmu okrem fuzzy naivného klasifikátora, ktorý je považovaný za základ algoritmu, prípadne vynútiť dva zoznamy pravdepodobností, kde jeden bude obsahovať len pravdepodobnosti z fuzzy naivného bayesovského klasifikátora. Zároveň bude schopný zvoliť akým spôsobom sa majú počítat korekcie a aké funkcie príslušnosti budú použité.

## 5. NÁVRH MODULU



Obr. 5.3: Diagram tried: Podrobný popis triedy DithyronComputation



Obr. 5.4: Databázový model modulu DITHYRON

### 5.2.2 Databázový model

Do databázového modelu Alma aplikácie (viď obrázok 4.3) je nutné doplniť entitu reprezentujúcu požiadavky na modul Dithyron (viď obrázok 5.4). Entita rešpektuje štruktúru požiadaviek na moduly BIANTA a CRACTES.

Jednotlivé atribúty entity `DithyronRequest` sú:

- `DateTime` – primárny kľúč obsahujúci dátum a čas požiadavku.
- `AccountID` – cudzí kľúč do tabuľky `Account` reprezentujúci unikátny identifikátor užívateľa.
- `LastName` – voliteľné priezvisko pacienta.
- `FirstName` – voliteľné krstné meno pacienta.
- `Gender` – pohlavie pacienta.
- `BirthYear` – rok narodenia pacienta.
- `Markers` – všetky laboratórne merania pacienta obsahujúce názov markeru, dátum a hodnotu merania.
- `MCount` – počet meraní.
- `OwnID` – voliteľné rodné číslo alebo iný identifikátor pacienta.
- `Note` – voliteľná poznámka o vyšetrení pacienta.
- `Archived` – príznak, či sa jedná o archivované vyšetrenie.
- `ArchivedDate` – dátum archivácie.
- `Diagnose` – voliteľná predchádzajúca diagnóza pacienta.

K užívateľským účtom je nutné pridať atribút `DithyronCutoff` reprezentujúci cut-off hodnoty referenčných intervalov nastavených podľa užívateľovho laboratória. Tento atribút bude uložený vo formáte JSON. Ak bude atribút prázdny, budú použité predvolené hodnoty modulu.

Databázová entita `DithyronRequest` bude pomocou ORM namapovaná na triedu s rovnakým názvom a rovnakými atribútami v rámci Alma Diagnostics aplikácie. Inštancie triedy `DithyronRequest` sa budú vytvárať z atribútov `Patient`.

Ako bolo povedané v časti 5.1, trieda `DithyronCutoff` v rámci aplikácie bude obsahovať metódy na serializáciu a deserializáciu JSON objektu uloženého v databáze.

### 5.2.3 Controller

Funkcie Controllera a adresy, ktoré obsluhuje budú vychádzať z existujúcich adries a funkcií pre moduly BIANTA a CRACTES, aby bolo zaručené podobné chovanie pre všetky moduly a zároveň, aby mohol byť znovupoužitý existujúci kód.

Vzhľadom na to, že sa nejedná o žiadnu inovatívnu činnosť návrh tejto časti nie je nutný.

### 5.2.4 Ostatné funkcie modelu

Funkcie na komunikáciu s databázou, import údajov o pacientovi a export vyšetrenia do PDF tiež nie je nutné navrhovať, pretože budú použité už existujúce funkcie aplikácie Alma Diagnostics.

Tieto funkcie budú rozšírené tak, aby zvládali pracovať aj s dátami nového modulu.

### 5.2.5 View – užívateľské rozhranie

Návrh užívateľského rozhrania počíta s využitím existujúcich šablón.

Rozhranie pre zadávanie informácií o pacientovi bude vychádzať z rozhrania modulu BIANTA (viď obrázok 4.4).

Na tomto rozhraní však bolo identifikovaných niekoľko nedostatkov vzhľadom na použiteľnosť:

1. Pri zadávaní koncentrácie meraní markerov sa nezobrazujú ich jednotky (jednotky používané aplikáciou a laboratóriom sa môžu líšiť).
2. Užívateľovi je umožnené zmazať vždy len posledné meranie (prípadne posledné tri) a nie ktorékoľvek.
3. Formulár na zadávanie meraní sa nevhodne zobrazuje na zariadeniach s menšou obrazovkou (viď obrázok 5.6).

Na obrázku 5.5 sa nachádza návrh riešenia prvých dvoch problémov a na obrázku 5.7 tretieho.

Komponenty na správu vyšetrení pacienta, import zo súbor a import zo schránky môžu byť prevzaté z existujúcich modulov bez zmien.

Návrh komponentu na nastavenie hodnôt referenčných intervalov sa nachádza na obrázku 5.8.

Najdôležitejšou časťou návrhu je prezentácia výsledkov vyšetrení, ktoré sa pre všetky moduly líšia. Na vyhodnotenie koncentrácií markerov vzhľadom na cut-off hodnoty a ich dynamiku vývoja v čase môžu byť použité existujúce komponenty modulu BIANTA.



Obr. 5.5: Návrh řešení problémů při zadávání měření

Obr. 5.6: Nevhodně zobrazený formulář na zadávání měření na malých obrazovkách v aplikaci Alma Diagnostics [59]

Obr. 5.7: Návrh řešení problému s nevhodně zobrazeným formulářem

Obr. 5.8: Návrh komponentu na nastavenie hodnôt referenčných intervalov

Časť rozhrania zobrazujúcu jednotlivé diagnózy a chovanie markerov pre tieto diagnózy však bolo nutné navrhnuť.

Dôraz pri návrhu bol kladený na prehľadné zobrazenie chovania markerov, pretože to je jeden z identifikovaných nedostatkov historickej implementácie (viď časť 3.4.1.6).

Na tieto účely boli vytvorené tri počiatočné návrhy:

1. **tabuľka** (viď obrázok 5.9),
2. **stĺpcový graf** (viď obrázok 5.10),
3. **kruhový graf** (viď obrázok 5.11).

Nevýhodou tabuľky je to, že návrh už počíta s využitím tabuliek na vyhodnocovanie koncentrácií markerov a dynamiku ich vývoja v čase. Rozhranie obsahujúce mnoho tabuliek by bolo neprehľadné a užívateľ by sa v ňom mohol ťažko orientovať.

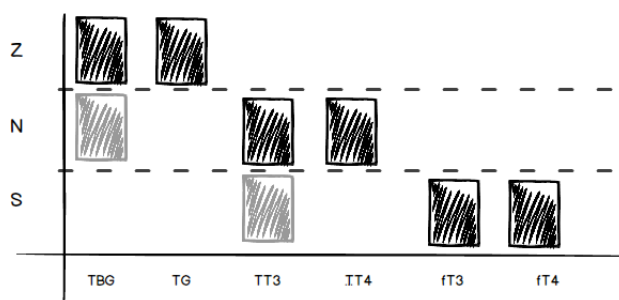
Nevýhodou stĺpcového grafu je to, že pri zobrazení mnohých markerov by sa stal horizontálne veľmi dlhý, čo by zhoršovalo jeho čitateľnosť.

Pri implementácii rozhrania sa teda využije kruhový graf, ktorý je kompaktný aj pre väčší počet markerov. Prvotný návrh z obrázka 5.11 bol rozpracovaný do finálneho návrhu, ktorý sa nachádza na obrázku 5.12.

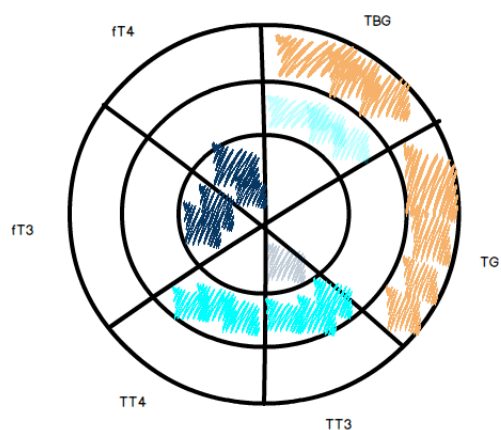
Návrh celkového rozhrania na zobrazovanie výsledkov sa nachádza na obrázku 5.13.

|     | Zvýšený | Normální | Znížený |
|-----|---------|----------|---------|
| TBG | ✓       | občas    | ✗       |
| TG  | ✓       | ✗        | ✗       |
| TT3 | ✗       | ✓        | občas   |
| TT4 | ✗       | ✓        | ✗       |
| fT3 | ✓       | ✗        | ✗       |
| fT4 | ✓       | ✗        | ✗       |

Obr. 5.9: Návrh zobrazenia chovania markerov pomocou tabuľky

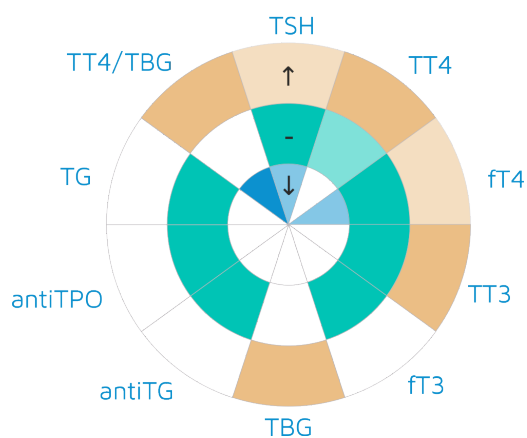


Obr. 5.10: Návrh zobrazenia chovania markerov pomocou stĺpcového grafu, kde Z znamená zvýšená hladina, N normálna hladina, S znížená hladina a slabší odtieň farby znamená, že toto chovanie nastáva len občas

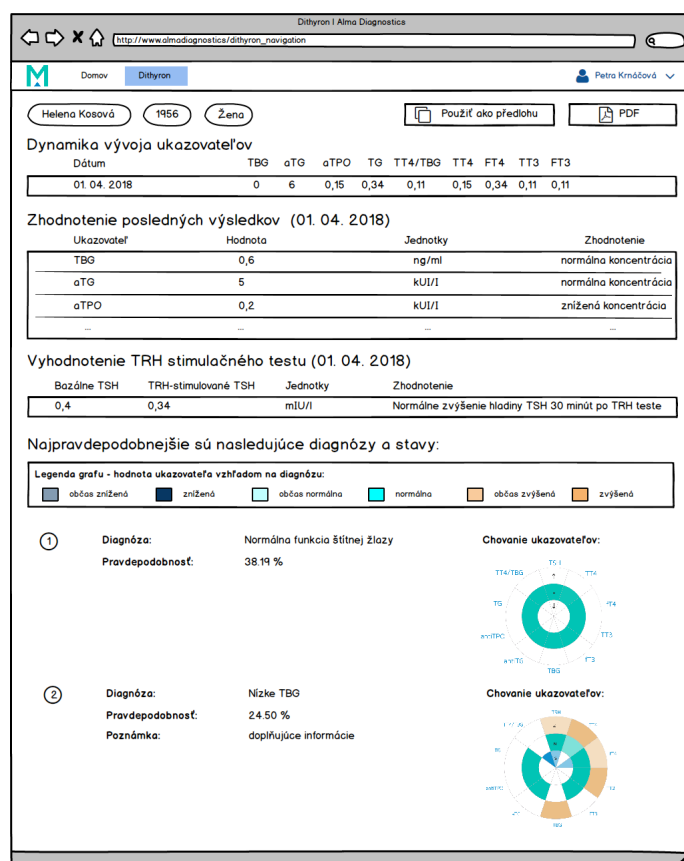


Obr. 5.11: Návrh zobrazenia chovania markerov pomocou kruhového grafu, kde slabší odtieň farby znamená, že toto chovanie nastáva len občas

## 5. NÁVRH MODULU



Obr. 5.12: Finálny návrh zobrazenia chovania markerov pomocou kruhového grafu, kde slabší odtieň farby znamená, že toto chovanie nastáva len občas



Obr. 5.13: Návrh rozhrania na zobrazovanie výsledkov vyšetrení

---

## Realizácia

Táto kapitola popisuje akým spôsobom bolo postupované pri implementácii navrhnutého riešenia.

### 6.1 Verzovanie

Na správu verzií Alma Diagnostics aplikácie je použitý verzovací nástroj Mercurial [61]. Implementácia bola započatá naklonovaním repozitára obsahujúceho zdrojové kódy k Alma Diagnostics a vytvorením vetvy „Dithyron“ určenej na vývoj nového modulu.

Vytvorenie novej vetvy bolo vykonané z dôvodu, že nie je vhodné do produkčného kódu pridávať neotestované zmeny.

### 6.2 Fázy implementácie

Implementácia bola rozdelená do dvoch fáz:

1. Implementácia výkonného algoritmu.
2. Zapojenie algoritmu do Alma Diagnostics aplikácie.

#### 6.2.1 Prvá fáza

V prvej fáze implementácie bolo nutné vytvoriť súbory so znalosťami, implementovať algoritmus a otestovať jeho základnú funkčnosť.

##### 6.2.1.1 Vytvorenie súborov so znalosťami

Na vytvorenie súborov so znalosťami bolo nutné tieto znalosti vyextrahovať zo zdrojového kódu historickej implementácie algoritmu, kde sú uložené v poliach.

Na tieto účely bola do zdrojového kódu historickej implementácie pridaná metóda na výpis polí do súboru v navrhnutom formáte (viď časť 5.2.1.1).

### 6.2.1.2 Implementácia

Implementácia bola vytvorená podľa objektového návrhu popísaného v časti 5.2.1, pričom funkcionálnosť jednotlivých komponentov zodpovedá popisu algoritmu z kapitoly 2.

Implementácia tried a definícia rozhraní, ktoré môžu byť znovupoužité inými algoritmami je umiestnená v Python module (súbore) `computation_models.py`. Pomocné funkcie, ktoré sú využívané viacerými komponentami algoritmu sú umiestnené v module `computation_functions.py`.

Implementácia triedy `DithyronComputation`, ktorá prepojuje jednotlivé komponenty, sa nachádza v module `dithyron_computation.py`.

Slovníky znalostí a funkcie na ich naplnenie zo súboru sú umiestnené v module `computation_data.py`. Súbor so znalosťami sú umiestnené v zložke `static` v podzložke `data`.

Počas tejto fázy boli priebežne s implementáciou tvorené aj jednotkové testy (viac v časti 7.1) na odladenie chýb komponentov výpočtu. Zároveň boli do zdrojového kódu ku každému komponentu a jeho metódam pridané komentáre vo formáte, z ktorého je možné pomocou vstavaného modulu Pythonu `pydoc` vygenerovať dokumentáciu v HTML formáte.

### 6.2.2 Druhá fáza

Po otestovaní základnej funkcionality algoritmu bola započatá druhá fáza implementácie a to vytvorenie nového modulu realizujúceho ostatné funkčné požiadavky a jeho zakomponovanie do existujúcej Alma Diagnostics aplikácie. Na tieto účely bolo nutné vytvoriť užívateľské rozhranie, funkcie na obsluhu užívateľských požiadaviek, prepojenie s databázou a preklad.

Väčšina funkčných požiadaviek na modul DITHYRON vychádza z funkčných požiadavkov modulov BIANITA a CRACTES. Preto výraznou črtou druhej fázy implementácie bolo využitie už existujúcich častí Alma Diagnostics aplikácie a refactoring tých častí, ktoré nebolo možné hneď použiť.

Druhá fáza implementácie bola ukončená vykonaním integračných testov (viac v časti 7.3).

#### 6.2.2.1 Refactoring

Užívateľské rozhranie je z veľkej časti realizované šablónami, ktoré používajú moduly BIANITA a CRACTES, v niektorých prípadoch však nebolo možné použiť celé existujúce šablóny, pretože sa návrh užívateľského rozhrania DITHYRONU líšil v malých detailoch, prípadne do šablóny pridával novú funkcionálnosť. Z tohto dôvodu boli z existujúcich šablón vyextrahované vybrané ucelené celky (napríklad komponent na zadávanie základných údajov o pacientovi) do nových šablón, ktoré môžu byť vložené do iných šablón pomocou príkazu `{% include "názov šablóny"%}`.

Ako už bolo povedané v časti 5.2.3, rovnako funkcie Controlleru slúžiace na obsluhu požiadaviek prichádzajúcich na pridelené URI zdieľajú väčšinu funkcionality, pretože požiadavky sú podobné.

Spracovanie konkrétnych typov požiadaviek však bolo často priamo súčasťou funkcií obsluhujúcich URI pre moduly BIANTA a CRACTES a nebolo teda priamo použiteľné modulom DITHYRON.

Pre umožnenie zdieľania funkcionality modulom DITHYRON bolo toto spracovanie vyextrahované do metód. Jedná sa napríklad o tieto metódy:

- `handle_cutoff_request` – spracováva požiadavky na aktualizáciu cut-off hodnôt v databáze,
- `handle_import_request` – spracováva požiadavky na vyšetrenie pacienta zo súboru,
- `handle_clipboard_request` – spracováva požiadavky na vyšetrenie pacienta zo schránky,
- `get_results` – spracováva požiadavky na získanie všetkých vyšetrení z databázy.

Metódy boli upravené tak, aby zvládali pracovať so všetkými modulami a na rozlíšenie, o ktorý modul sa práve jedná, bola vytvorená enumerácia `Modules`. Jedna z hodnôt tejto enumerácie je posledným parameterom metód.

### 6.2.2.2 Implementácia užívateľského rozhrania a metód obsluhujúcich požiadavky

Šablóna na vytváranie nového pacienta bola poskladaná z existujúcich častí, do ktorých bolo doplnené tlačidlo na mazanie ktoréhokoľvek merania. Funkcionality pre mazanie ľubovoľného merania bola doplnená aj do funkcie obsluhujúcej požiadavky.

Na účely zobrazovania jednotiek podľa aktuálne zvoleného typu markeru bol napísaný jednoduchý skript v jazyku JavaScript, ktorý sleduje zmeny na komponente `select` a podľa zvolenej možnosti zobrazí priradené jednotky.

Šablóna, obsahujúca formulár na zmenu hodnôt referenčných intervalov, bola realizovaná podľa návrhu. Na zabránenie zadania nesprávnych hodnôt (spodná cut-off hodnota vyššia ako horná) bol vytvorený skript v jazyku JavaScript, ktorý kontroluje túto podmienku a nedovolí užívateľovi zadať nevalidné hodnoty.

Metóda na spracovanie formuláru a uloženie zmenených hodnôt do databázy vychádza z existujúcej metódy pre moduly BIANTA a CRACTES, avšak v ich prípade sa zadáva vždy len jedna cut-off hodnota a zvyšné 3 sa z nej pomerovo dopočítajú. V prípade modulu DITHYRON sa zo spodnej hodnoty referenčného intervalu dopočíta hodnota oddeľujúca intervaly pre veľmi

znižené a znížené hodnoty. Analogicky sa z hornej hodnoty dopočíta cut-off hodnota oddeľujúca intervaly pre zvýšené a veľmi zvýšené hodnoty.

Na zobrazovanie výsledkov bola vytvorená šablóna realizujúca návrh. Jednotlivé grafy chovania markerov boli pre každú diagnózu nakreslené pomocou programu *CorelDRAW* [62].

Zvyšné časti rozhrania a obsluhujúce metódy boli vytvorené s využitím existujúceho kódu. Ukážka výsledného užívateľského rozhrania sa nachádza v prílohe C.

### 6.2.2.3 Konfigurácia algoritmu

Na konfiguráciu algoritmu administrátorom slúži súbor `config.py`, ktorý obsahuje konfiguráciu celej Alma Diagnostics aplikácie.

V súbore sú uvedené konštanty, ktoré je možné zmeniť bez toho, aby bolo nutné reštartovať server. Súbor obsahuje konštanty na prispôsobenie algoritmu podľa možností jeho nastavenia navrhnutých v časti 5.2.1.9.

Ďalej obsahuje konštanty na úpravu výstupu – koľko diagnóz a do akej pravdepodobnosti má byť zobrazené užívateľovi.

### 6.2.2.4 Preklad

Ako bolo povedané v časti 4.2.6 na lokalizáciu aplikácie je použitá knižnica Flask-Babel, pričom všetky reťazce sú v rámci aplikácie písané v angličtine.

Do vygenerovaných `.po` súborov bolo nutné doplniť preklad modulu do slovenského jazyka. Na jednoduchšiu prácu s `.po` súbormi bol použitý program *Poedit* [63].

Do slovenskej lokalizácie je ešte nutné doplniť preklady názvov diagnóz, ktoré budú vykonané odborníkom z oblasti medicíny.

Preklady do češtiny a iných jazykov budú vykonané osobami, ktoré tieto jazyky dobre ovládajú.



---

# Testovanie, nasadenie a porovnanie

Táto kapitola popisuje vykonané testovania, nasadenie verzie Alma Diagnostics s modulom DITHYRON a prináša záverečné porovnanie súčasnej a historickej implementácie algoritmu na detekciu porúch štítnej žľazy.

## 7.1 Jednotkové testy

Na testovanie jednotlivých komponentov boli vytvorené jednotkové testy pomocou frameworku *PyUnit* [64], ktorý je štandardným testovacím frameworkom programov napísaných v jazyku Python.

Na testy bol vytvorený balíček s názvom `tests`. Pre každú testovanú triedu bol vytvorený Python modul s názvom `test_nazov_triedy.py`.

Obsahom modulov sú jednotlivé testovacie prípady vzťahujúce sa k danej triede. Testy komponentov boli vykonávané priebežne počas implementácie, pričom dôraz bol kladený na otestovanie funkčnosti komponentu podľa popisu algoritmu, otestovanie možných variánt prechodov zdrojového kódu v komponente a otestovanie chovania pri hraničných hodnotách.

Jednotkové testy odhalili niekoľko chýb týkajúcich sa napríklad možného neošetreného delenia nulou, chybného výpočtu lineárnych lomených funkcií či nesprávneho prechodu slovníkom meraní markerov.

Všetky tieto chyby boli počas implementácie priebežne opravované a jednotlivé komponenty boli znovu testované.

## 7.2 Testovanie na reálnych dátach

Na otestovanie správnosti implementácie algoritmu bolo autorom algoritmu poskytnutých 140 súborov s dátami vyšetrení reálnych pacientov s potvrdenou

diagnózou. Rozmedzie dátumov vyšetrení sa pohybuje od roku 1997 do roku 2001.

Tieto dáta boli použité na validovanie historickej implementácie, pričom správna diagnóza sa vždy nachádzala na jednom z prvých troch miest, prípadne bolo správne určené, že sa jedná o zdravého pacienta. Bohužiaľ sa nezachovala informácia, o ktorú z výstupných diagnóz sa jednalo.

V rámci testovania novej implementácie algoritmu boli tieto vyšetrenia použité na porovnanie výstupov novej implementácie s historickou.

Pri testovaní bolo narazené na problém, že historická implementácia dáva na prvé miesto, kvôli chybe pri vyhodnocovaní progresu ochorenia (chyba popísaná v časti 3.4.4.1), progres ochorenia, aj keď chovanie markerov tomu v skutočnosti nenasvedčuje.

Výsledky aktuálnej a historickej implementácie sa teda líšili. Z dôvodu objektívneho otestovania súčasnej verzie voči historickej implementácii bolo v rámci aktuálnej implementácie vypnuté vyhodnocovanie progresu a v rámci historickej implementácie bola zakomentovaná časť kódu, kedy sa progres pridať do poľa pravdepodobností diagnóz.

Po tejto úprave sa výsledky oboch implementácií líšili len v zaokrúhľovaní desiatinných čísiel.

Správnosť implementácie progresu bohužiaľ nebolo možné na reálnych dátach otestovať.

### 7.3 Integračné testovanie

Integračné testovanie všetkých častí modulu bolo vykonané manuálne pomocou scenárov, ktoré reflektujú prípady použitia. Jednotlivé scenáre aj s ich vyhodnotením a návrhom opravy zistených chýb sa nachádzajú v prílohe D.

Celkom bolo pomocou testovacích scenárov zistených 5 chýb, z ktorých boli opravené 4. Chyba zistená tretím scenárom (viď tabuľka D.3), týkajúca sa resetovania zadanej predchádzajúcej diagnózy pri zmene jazyka, nebola opravená, pretože nie je vážna a na jej opravu je nutné zasiahnuť do viacerých častí systému. Chyba bude opravená v rámci údržby modulu.

### 7.4 Testovanie užívateľského rozhrania

Testovanie užívateľského rozhrania bolo vykonané 4 študentami medicíny. Študenti medicíny boli zvolení z dôvodu, že rozumejú pojmom použitým v aplikácii a prípadné komplikácie s používaním aplikácie by nevznikli z nepochopenia domény.

### 7.4.1 Priebeh testovania

Pre začatím testovania bol študentom poskytnutý leták so stručným popisom na čo slúži modul Dithyron, z akých zdrojov boli čerpané jeho znalosti a aká je funkcionálna Alma Diagnostics aplikácia (import, export, šifrovanie, správa vyšetrení, nastavenie referenčných intervalov). V letáku bol zároveň uvedený unikátny kľúč slúžiaci na prihlásenie do aplikácie a webová adresa aplikácie.

Študenti boli potom usadení za počítač, kde im bola ukázaná zložka, v ktorej boli tri súbory `pacient1.txt`, `pacient2.txt` a `ref-intervaly.txt` a bolo im vysvetlené, že prvé dva súbory sú laboratórne výsledky vyšetřovaného pacienta a druhý súbor referenčné intervaly použitého laboratória.

Ďalej bol študentom na papieri poskytnutý scenár v nasledujúcom znení:

---

`scenar.txt`

---

Na vyšetrenie do vašej ordinácie prišiel pacient so strumou neznámeho pôvodu. Pôvodnú diagnózu ste stanovili na E04. Avšak, zdravotné komplikácie pacienta pretrvávajú, preto ste sa rozhodli využiť novozakúpený systém Dithyron. Výsledky pacienta z vášho systému sú v súbore `pacient.txt`.

1. Vyšetrite pacienta pomocou systému Dithyron.
2. Z vyšetrenia vzhľadom na najpravdepodobnejšiu diagnózu určite, ktoré ďalšie markre by bolo vhodné zmerať.

Po získaní nových laboratórnych výsledkov pacienta, s domeranými markerami, chcete zistiť či laboratórne výsledky potvrdzujú hypotetickú diagnózu. Výsledky pacienta z vášho systému sú v súbore `pacient2.txt`.

3. Zadaťte do systému nové výsledky.
4. Prezrite si najpravdepodobnejšiu diagnózu.

Po prezretí najpravdepodobnejšej diagnózy usúdite, že tento typ diagnózy by potreboval vyšetrenie špecialistom z onkológie.

5. Exportujte výsledky do PDF pre odoslanie špecialistovi.

---

Študenti boli ubezpečení, že sa pri testovaní nemusia ponáhľať, a že je testovaný systém a nie ich vedomosti z oblasti medicíny. Po ubezpečení bolo započaté samotné testovanie.

Po vykonaní hlavného scenára boli študenti vyzvaní vykonať doplňujúce scenáre, pokiaľ ich už nevykonali počas hlavného scenára:

1. Nastavte hodnoty referenčných intervalov na hodnoty vášho pracoviska.
2. Importuje laboratórne výsledky pacienta zo súboru.
3. Zadaťte informácie o nejakom pacientovi ručne pomocou pripraveného rozhrania. Vymažte meranie niektorého z markerov.

4. Odstráňte výsledky vyšetrenia vášho prvého pacienta.
5. Vstúpte do šifrovacieho módu a vytvorte pacienta so zašifrovanými osobnými údajmi.

### 7.4.2 Postup vyhodnocovania

Pri vykonávaní hlavného scenára bolo skúmané:

- ktoré komponenty systému pri vyšetrení pacienta študenti medicíny využijú,
- jednoduchosť orientácie na stránkach,
- použiteľnosť jednotlivých komponentov,
- čitateľnosť prezentovaných výsledkov.

Pri doplnkových scenároch bola skúmaná použiteľnosť jednotlivých komponentov.

### 7.4.3 Výsledky testovania

**Priebeh hlavného scenára:**

- Všetci 4 študenti využili na zadanie informácií o pacientovi import zo súboru.
- Jeden študent pri zadávaní informácií aj nastavil hodnoty referenčných intervalov.
- Všetci študenti vedeli manipulovať s hlavným aj postranným menu.
- Všetci 4 študenti sa vedeli orientovať vo výsledkoch, určili najpravdepodobnejšiu diagnózu a určili, ktoré markre by bolo dobré domerať na jej potvrdenie alebo vyvrátenie.
- Nikto zo študentov nevyužil možnosť šifrovania.

**Čitateľnosť grafov:**

- Všetci študenti boli schopní vyčítať z grafov chovanie markerov a nikto s tým nemal väčší problém.
- Dvaja študenti uviedli, že ikony vo vnútri grafu sú mätúce a duplicitné k prehľadnej a jasnej legende a uvítali by ich odstránenie.
- Jeden študent uviedol, že použitie slabšieho odtieňa farby na odlíšenie, kedy chovanie nastáva len občas, nie je úplne vhodné pre užívateľov, ktorí majú problémy so zrakom, prípadne v zlých svetelných podmienkach.

- Jeden študent uviedol, že kruhové grafy v rámci štúdia medicíny nepoužívajú, ale nemal problém z nich vyčítať potrebné informácie.
- Jeden študent by uvítal, keby grafy boli interaktívne a chovanie sa zobrazovalo prejdением myšou nad jednotlivými časťami grafu.

#### **Doplnenie výsledkov vyšetrenia pacienta:**

- Dvaja študenti na doplnenie výsledkov hneď použili tlačidlo použiť ako predlohu.
- Dvaja študenti chceli doplniť merania pomocou stránky pre zobrazovanie predchádzajúcich výsledkov, kde si zobrazili predchádzajúce výsledky pacienta a potom použili tlačidlo použiť ako predlohu.
- Jeden študent uviedol, že mu prišlo zvláštne, že sa dajú upravovať osobné údaje pacienta.
- Jeden študent bol trochu zmätený z faktu, že sa vyšetrenia neukladajú ku pacientovi a pre rôzne vyšetrenia toho istého pacienta je v tabuľke viacero riadkov, keď sa komponent nazýva vytvoriť pacienta. Uvítal by zmenu názvu na vytvoriť vyšetrenie alebo nové vyšetrenie.
- Všetci študenti nové laboratórne výsledky importovali zo súboru a skontrolovali ich správnosť.

#### **Export do PDF**

- Ani jeden študent nemal s týmto krokom žiadny problém.
- Jeden študent uviedol, že by bolo vhodnejšie, keby sa PDF otvára v novej karte, pretože na návrat späť z PDF do aplikácie je nutné použiť tlačidlo späť internetového prehliadača.

#### **Vytvorenie pacienta pomocou formulára a zmazanie merania**

- Ani jeden študent nemal s týmto krokom žiadny problém.
- Jeden študent uviedol, že mu vadilo, že pri pridaní nového merania sa zobrazenie stránky posunie na začiatok.

#### **Nastavenie hodnôt referenčných intervalov**

- Všetci študenti boli schopní nastaviť hodnoty referenčných intervalov.
- Jeden študent s tým mal menšie problémy, pretože mu nebolo jasné, čo znamená „Muž: 0 – 0“, „Muž: 1 – 5“ a musel sa na to opýtať. Po zistení, že sa jedná o vek, referenčné intervaly nastavil.

- Dvaja študenti nemali problém s pochopením o čo sa jedná, ale chýbalo im tam slovo „rokov“. V prípade „0 – 0“ by im čitateľnejšie pripadalo „menej ako 1 rok“.

### Zmazanie posledného vyšetrenia pacienta

- Všetci študenti boli schopní zmazať vyšetrenie.
- Jeden študent uviedol, že mu chýba potvrdzovací formulár na mazanie, pretože je veľmi pravdepodobné, že by sa niekedy pri používaní systému uklikol.
- Jeden študent uviedol, že by uvítal možnosť filtrovania výsledkov.

### Šifrovanie

- Všetci študenti boli schopní lokalizovať, kde sa šifrovanie zapína.
- Všetci však s jeho nastavením mali problémy.
- Dvaja študenti chceli do políčka na zadanie nimi vymysleného šifrovacieho kľúča, zadať prihlasovací kľúč.
- Traja si neboli istí ako šifrovací kľúč potvrdiť a či po potvrdení sú v šifrovacom móde.
- Dvaja študenti boli zmätení zobrazovaním tooltipu v znení „Zapamätajte si váš šifrovací kľúč!“ pri ikone určenej na zrušenie šifrovania.

#### 7.4.4 Analýza výsledkov

Všetci študenti boli schopní vykonať zadané scenáre, pričom vážnejšie problémy mali jedine s komponentom šifrovania, ktorého vývoj nebol cieľom tejto diplomovej práce.

Testovanie však odhalilo niekoľko nedostatkov, ktoré je nutné odstrániť a prinieslo niekoľko návrhov, ktoré je vhodné zvážiť.

Ich zoznam je nasledovný:

1. Zmeniť spôsob zapisovania vekových rozhraní pri nastavovaní cut-off hodnôt.
2. Zvážiť odstránenie duplicitných šípok v grafe.
3. Otestovať čitateľnosť grafov osobami s problémami so zrakom.
4. Zvážiť premenovanie tlačidla „Vytvoriť nového pacienta“ na „Vytvoriť nové vyšetrenie“.

5. Zvážiť otváranie PDF súboru v novej karte.
6. Zvážiť pridanie potvrdzovacieho formulára pri mazaní pacienta, prípadne sprístupnenie archívu bežným užívateľom.
7. Zvážiť možnosť doplnenia funkcionality filtrovania a vyhľadávania.
8. Ku komponentu šifrovania doplniť tlačítko na odosielanie, ktoré chýba.
9. Ku komponentu šifrovania doplniť podrobnejšie tooltipy, ktoré by opísali jeho použitie. Pričom by sa tooltipy menili v prípade aktívneho a neaktívneho šifrovania.

## 7.5 Nasadenie

Pred nasadením novej verzie aplikácie Alma Diagnostics, obsahujúcej aj modul DITHYRON, do produkčného prostredia Apache HTTP servera, bolo v rámci repozitára nutné zlúčiť vetvu určenú na vývoj modulu s hlavnou vetvou.

Po tomto zlúčení bol na serveri spustený skript, ktorý stiahol novú verziu z repozitára, vytvoril pre ňu nové virtuálne prostredie so všetkými potrebnými závislosťami a nasadil ju na Apache HTTP server.

Modul DITHYRON je nasadený v testovacej prevádzke, to znamená, že je dostupný len užívateľom s administrátorskými právami.

## 7.6 Záverečné porovnanie s historickou verziou

Pri tvorbe modulu do webovej aplikácie Alma Diagnostic bolo postupované podľa popisu algoritmu a objektového návrhu, a zároveň boli vykonané úkony (viď tabuľka 3.1) na vyriešenie problémov zistených u historickej implementácie algoritmu.

Na overenie správnosti implementácie algoritmu boli vykonané jednotkové testy a testy na reálnych dátach, ktoré ukázali, že obe implementácie vydávajú rovnaké výsledky (okrem spomenutého problému s vyhodnotením progresu ochorenia).

Výsledné porovnanie dvoch implementácií sa nachádza v tabuľke 7.1.

| <b>Historická implementácia</b>                 | <b>Súčasná implementácia</b>                                                                  |
|-------------------------------------------------|-----------------------------------------------------------------------------------------------|
| Desktopová aplikácia pre operačný systém MS-DOS | Modul webovej aplikácie fungujúcej na všetkých bežne používaných prehliadačoch                |
| Chýba podpora pre iné zariadenia ako počítače   | Podpora pre mobily a tablety                                                                  |
| Pascal                                          | Python                                                                                        |
| Komunikácia cez súbory                          | GUI, možnosť importu zo súborov nemocničných informačných systémov a exportu výsledkov do PDF |
| Konfigurácia cez súbory                         | Užívateľská konfigurácia cez GUI, administrátorská cez súbory                                 |
| Manuálne aktualizácie                           | Automatické aktualizácie                                                                      |
| Chýbajúca programátorská dokumentácia           | Dokumentácia vo forme pydoc komentárov                                                        |
| Chýbajúca užívateľská dokumentácia              | Dokumentácia v podobe tooltipov GUI                                                           |
| Spaghetti architektúra                          | Rozšíriteľná architektúra s jasne oddelenými časťami algoritmu                                |
| Znalosti súčasťou zdrojového kódu               | Znalosti oddelené do súborov a triednych atribútov                                            |
| Chýbajúca správa vyšetrení pacientov            | Vyšetrenia pacientov uložené v databáze, možnosť zdieľania v rámci skupiny užívateľov         |
| Chýbajúce testy                                 | Sada jednotkových testov                                                                      |

Tabuľka 7.1: Porovnanie historickej a súčasnej implementácie algoritmu na pomoc s diagnostikou porúch štítnej žľazy



---

## Diskusia

V rámci diplomovej práce bol revitalizovaný historický program DITHYRON, ktorý vznikol v 90. rokoch, pre operačný systém MS-DOS. Bol vytvorený modul webovej aplikácie Alma Diagnostics s rozšíriteľnou architektúrou a oddelenými znalosťami do súborov. V rámci implementácie boli odstránené identifikované nedostatky historickej implementácie a funkčnosť modulu bola otestovaná.

Testovanie užívateľského rozhrania však ukázalo, že použiteľnosť programu trpí menšími nedostatkami, ktoré je nutné odstrániť. Zároveň je nutné vykonať robustnejšie testovanie, ktoré by zahŕňalo priamo odborníkov z oblasti endokrinológie.

Aplikácia podporuje import dát z niektorých medicínskych informačných systémov ako je napríklad STAPRO Medea, ale na umožnenie širokého rozšírenia a plnohodnotného prepojenia s medicínskymi informačnými systémami, by bolo vhodné, keby aplikácia podporovala import zo a export do súčasnej verzie Datového standardu Ministerstva zdravotníctví ČR (DASTA DS4).

Základňa znalostí, ktorou modul disponuje pochádza z 90. rokov, preto je nutné ju v rámci ďalšieho rozvoja modulu aktualizovať. Na tieto účely by bolo vhodné získať aktuálne prevalencie ochorení a aktuálne výsledky vyšetrení pacientov s validovanou diagnózou.

Vylepšiť by sa dalo aj oddelenie znalostí, ktoré sa v súčasnosti nachádzajú v textových súboroch. Znalosti je možné modelovať v podobe ontológií ako u analyzovaného systému OBESTDD [17], za uváženie by stálo aj ich ukladanie do nejakej NoSQL databázy.

Algoritmus pracuje s mnohými matematickými výrazmi, ktoré sú v rámci realizácie modulu naprogramované pomocou štandardných prostriedkov jazyka Python. Pre zlepšenie čitateľnosti týchto výrazov by bolo možné využiť knižnicu *NumPy* [65], ktorá poskytuje funkcionality pre vedecké výpočty.



---

## Záver

Hlavným cieľom práce bola revitalizácia zastaraného programu DITHYRON do formy modulu webovej aplikácie Alma Diagnostics, tak aby spĺňal požiadavky na použitie v súčasnosti a možné rozšírenie v budúcnosti.

Na tieto účely bola zanalyzovaná doména medicínskych systémov, existujúce systémy na detekciu porúch štítnej žľazy aj komerčne využívané CDSS a boli identifikované problémy, ktorými mnohé CDSS trpia, a pri návrhu a implementácii nového modulu, ich bolo nutné mať na zreteli.

Pomocou analýzy zdrojového kódu historickej implementácie a následných konzultácií s autorom algoritmu doc. RNDr. Ladislavom Pecnom, CSc., bol vytvorený podrobný popis algoritmu, ktorý bol využitý na vytvorenie modulu, nakoľko časti zdrojového kódu historickej aplikácie, z dôvodu neprehľadnej architektúry a miešania zodpovedností, neboli priamo použiteľné.

Ďalej bola vykonaná analýza nedostatkov historickej implementácie, ktoré boli zohľadnené a odstránené pri návrhu a implementácii modulu.

Modul bol vytvorený v objektovo-orientovanom štýle za pomoci návrhových vzorov, tak aby jeho jednotlivé časti boli vymeniteľné a rozšíriteľné o novú funkcionality. Znalosti boli oddelené do súborov, aby bola zjednodušená ich správa. Modul bol zakomponovaný do webovej aplikácie a bolo k nemu vytvorené užívateľské rozhranie.

Na overenie funkčnosti jednotlivých komponentov boli vytvorené jednotkové testy, na overenie funkčnosti algoritmu ako celku bolo vykonané testovanie na reálnych dátach s následným porovnaním s výstupom historickej implementácie. Na otestovanie celého modulu v rámci aplikácie Alma Diagnostics boli vykonané integračné testy. Chyby odhalené týmito testovaniami boli opravené. Tiež boli vykonané základné testy užívateľského rozhrania, ktoré odhalili niekoľko nedostatkov v použiteľnosti, ktoré je nutné do budúcnosti vyriešiť.

Na záver bolo vykonané nasadenie Alma Diagnostics aplikácie obohatenej o nový modul a bolo vykonané porovnanie súčasnej a historickej implementácie.



---

## Literatúra

- [1] Encyclopedia.com. Medical Systems. *Computer Sciences* [online]. [cit. 2018-04-01]. Dostupné z: <https://www.encyclopedia.com/computing/news-wires-white-papers-and-books/medical-systems>
- [2] Hachaj, T.; Ogiela, M.: Nowadays and future computer application in medicine. *IT CoNvergence PRActice*, ročník 1, marec 2013. Dostupné z: [https://www.researchgate.net/publication/286447606\\_Nowadays\\_and\\_future\\_computer\\_application\\_in\\_medicine](https://www.researchgate.net/publication/286447606_Nowadays_and_future_computer_application_in_medicine)
- [3] Greenes, R. A. (editor): *Clinical Decision Support: The Road to Broad Adoption*. Oxford: Academic Press, druhé vydání, 2014, ISBN 978-0-12-398476-0, 1–2, 20–50 s.
- [4] Finlay, P.: *Introducing Decision Support Systems*. New Jersey: Blackwell Pub, druhé vydání, 1994, ISBN 978-1855543140.
- [5] Puppe, F.: *Systematic Introduction to Expert Systems: Knowledge Representations and Problem-Solving Methods*. Berlin: Springer-Verlag Berlin Heidelberg, první vydání, 1993, ISBN 978-3-642-77973-2, 3–4 s.
- [6] Alder, H.; aj.: Computer-Based Diagnostic Expert Systems in Rheumatology: Where Do We Stand in 2014? *International Journal of Rheumatology* [online], júl 2014, [cit. 2018-04-17]. Dostupné z: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4119620/>
- [7] Tsihrintzis, G. A.; Virvou, M.; Jain, L. C. (editoři): *Intelligent Computing Systems: Emerging Application Areas*. Studies in Computational Intelligence 627, Berlin: Springer-Verlag Berlin Heidelberg, první vydání, 2016, ISBN 978-3-662-49177-5, 1–2 s.
- [8] Graham, T. A. D.; Kushniruk, A. W.; Bullard, M. J.; aj.: How usability of a web-based clinical decision support system has the potential to contribute to adverse medical events. *AMIA Annual Symposium Proceedings*

- [online], november 2008: s. 257–261, ISSN 1942-597X, [cit. 2018-04-17]. Dostupné z: <http://europepmc.org/articles/PMC2655970>
- [9] Bates, D. W.; Kuperman, G. J.; Wang, S.; aj.: Ten Commandments for Effective Clinical Decision Support: Making the Practice of Evidence-based Medicine a Reality. *Journal of the American Medical Informatics Association* [online], ročník 10, č. 6, 2003: s. 523–530, [cit. 2018-04-17]. Dostupné z: <https://academic.oup.com/jamia/article/10/6/523/760582>
- [10] Porta, M.: *A dictionary of Epidemiology*. New York: Oxford University Press, šesté vydání, 2014, ISBN 978-0-19-997672-0.
- [11] MedicineNet. Medical Definition of Marker, blood. *MedicineNet.com* [online]. [cit. 2018-04-17]. Dostupné z: <https://www.medicinenet.com/script/main/art.asp?articlekey=11621>
- [12] Pilný, R.; Greplová, K.; Mrkvicová, M.: Referenční hodnoty laboratorních vyšetření OLM MOÚ [online]. Technická zpráva, 2018, [cit. 2018-04-15]. Dostupné z: <https://www.mou.cz/olm-referencni-hodnoty-laboratornich-vysetreni/f707>
- [13] Jiskra, J.: *Poruchy štítné žlázy: Praktický přehled nejen pro laickou veřejnost*. Praha: Mladá fronta a.s., první vydání, 2011, ISBN 978-80-204-2456-3.
- [14] Jiří Lukeš, P. K.: Laboratorní diagnostika onemocnění štítně žlázy [online]. *Interní medicína pro praxi*, ročník 3, č. 3, 2001: s. 120–123, ISSN 12127299, [cit. 2018-04-15]. Dostupné z: <https://www.internimedicina.cz/artkey/int-200103-0005.php>
- [15] Savelonas, M.; Maroulis, D.; Iakovidis, D.; aj.: A variable background active contour model for automatic detection of thyroid nodules in ultrasound images. In *IEEE International Conference on Image Processing 2005*, ročník 1, Nevada: IEEE Computer Society, 2005, ISBN 0-7803-9134-9, ISSN 1522-4880, s. 17–20.
- [16] Papageorgiou, E. I.; Papandrianos, N. I.; Apostolopoulos, D. J.; aj.: Fuzzy Cognitive Map based decision support system for thyroid diagnosis management. In *2008 IEEE International Conference on Fuzzy Systems (IEEE World Congress on Computational Intelligence)*, Nevada: IEEE Computer Society, 2008, ISBN 978-1-4244-1818-3, ISSN 1098-7584, s. 1204–1211.
- [17] Rawte, V.; Roy, B.: OBESTDD: Ontology Based Expert System for Thyroid Disease Diagnosis. In *2015 International Conference on Nascent Technologies in the Engineering Field (ICNTE)*, Nevada: IEEE Computer Society, 2015, ISBN 978-1-4799-7264-7, s. 1–6.

- [18] Fitzgerald, L. T.; Overall, J. E.; Williams, C. M.: A computer program for diagnosis of thyroid disease. *American Journal of Roentgenology*, ročník 97, č. 4, 1966: s. 901–905, ISSN 0361-803X.
- [19] Khanale, P.; Ambilwade, R. P.: A Fuzzy Inference System for Diagnosis of Hypothyroidism. *Journal of Artificial Intelligence* [online], ročník 4, 2011: s. 45–54, ISSN 1994-5450, [cit. 2018-04-21]. Dostupné z: <https://scialert.net/fulltext/?doi=jai.2011.45.54>
- [20] Ionita, I.; Ionita, L.: Prediction of Thyroid Disease Using Data Mining Techniques. *BRAIN. Broad Research in Artificial Intelligence and Neuroscience* [online], ročník 7, č. 3, 2016: s. 115–124, ISSN 2067-3957, [cit. 2018-04-21]. Dostupné z: <http://www.edusoft.ro/brain/index.php/brain/article/view/628>
- [21] Hoshi, K.; Kawakami, J.; Kumagai, M.; aj.: An Analysis of Thyroid Function Diagnosis Using Bayesian-Type and SOM-Type Neural Networks. *Chemical and Pharmaceutical Bulletin* [online], ročník 53, č. 12, 2005: s. 1570–1574, ISSN 1347-5223, [cit. 2018-04-21]. Dostupné z: [https://www.jstage.jst.go.jp/article/cpb/53/12/53\\_12\\_1570/\\_article](https://www.jstage.jst.go.jp/article/cpb/53/12/53_12_1570/_article)
- [22] Kousarrizi, M. R. N.; Seiti, F.; Teshnehlab, M.: An Experimental Comparative Study on Thyroid Disease Diagnosis Based on Feature Subset Selection and classification. *International Journal of Electrical & Computer Sciences IJECS-IJENS* [online], ročník 12, č. 1, 2012: s. 13–19, ISSN 2077-1231, [cit. 2018-04-21]. Dostupné z: [http://ijens.org/Vol\\_12\\_I\\_01/126001-8989-IJECS-IJENS.pdf](http://ijens.org/Vol_12_I_01/126001-8989-IJECS-IJENS.pdf)
- [23] Senol, C.; Yildirim, T.: Thyroid and breast cancer disease diagnosis using fuzzy-neural networks. In *2009 International Conference on Electrical and Electronics Engineering - ELECO 2009*, Nevada: IEEE Computer Society, 2009, ISBN 978-1-4244-5106-7, s. 390–393.
- [24] Ahmad, W.; Huang, L.; Ahmad, A.; aj.: Thyroid Diseases Forecasting Using a Hybrid Decision Support System Based on ANFIS, k-NN and Information Gain Method. *Journal of Applied Environmental and Biological Sciences*, ročník 7, 2017: s. 78–85, ISSN 2090-4274.
- [25] Mohammad Reza Obeidavi, O. M., Ali Rafiee: Diagnosing Thyroid Disease by Neural Networks. *Biomedical and Pharmacology Journal* [online], ročník 10, č. 2, 2017: s. 509–524, ISSN 0974-6242, [cit. 2018-04-21]. Dostupné z: <http://biomedpharmajournal.org/vol10no2/diagnosing-thyroid-disease-by-neural-networks/>
- [26] Keles, A.; Keles, A.: ESTDD: Expert system for thyroid diseases diagnosis. *Expert Systems with Applications*, ročník 34, č. 1, 2008: s. 242–246, ISSN 0957-4174.

- [27] Vardell, E.; Moore, M.: Isabel, a Clinical Decision Support System. *Medical reference services quarterly* [online], ročník 30, č. 2, 2011: s. 158–166, ISSN 1540-9597, [cit. 2018-04-21]. Dostupné z: [https://www.researchgate.net/publication/51090284\\_Isabel\\_a\\_Clinical\\_Decision\\_Support\\_System](https://www.researchgate.net/publication/51090284_Isabel_a_Clinical_Decision_Support_System)
- [28] Graber, M. L.; Mathew, A.: Performance of a Web-Based Clinical Diagnosis Support System for Internists. *Journal of General Internal Medicine*, ročník 23, č. 1, 2008: s. 37–40, ISSN 1525-1497.
- [29] Isabel Healthcare. Validation – Peer Reviews. *Isabel* [online]. [cit. 2018-04-21]. Dostupné z: <https://www.isabelhealthcare.com/validation/peer-reviews>
- [30] Isabel Healthcare. *Isabel* [software]. [přístup 2018-04-21]. Dostupné z: <https://www.isabelhealthcare.com/>
- [31] Massachusetts General Hospital: *DXplain Demo* [online]. [cit. 2018-04-21]. Dostupné z: <http://dxplain.org/demo2/dxpdemo.pdf>
- [32] Massachusetts General Hospital. DXplain. *DXplain* [online]. [cit. 2018-04-21]. Dostupné z: <http://www.mghlcs.org/projects/dxplain/>
- [33] Bond, W. F.; Schwartz, L. M.; Weaver, K. R.: Performance of a Web-Based Clinical Diagnosis Support System for Internists. *Journal of General Internal Medicine*, ročník 27, č. 2, 2012: s. 213–219, ISSN 0884-8734.
- [34] MDCalc. About MDCalc. *MDCalc.com* [online]. MDCalc. [cit. 2018-04-21]. Dostupné z: <https://www.mdcalc.com/about-us>
- [35] MDCalc. *MDCalc* [software]. [přístup 2018-04-21]. Dostupné z: <https://www.mdcalc.com/>
- [36] Marek, L.: *Pravděpodobnost*. Praha: Professional Publishing, první vydání, 2012, ISBN 978-80-7431-087-4.
- [37] Neubauer, J.; Sedláček, M.; Kříž, O.: *Základy statistiky: Aplikace v technických a ekonomických oborech*. Praha: Grada, první vydání, 2012, ISBN 978-80-247-4273-1.
- [38] Darwiche, A.: *Modeling and Reasoning with Bayesian Networks*. New York: Cambridge University Press, první vydání, 2009, ISBN 978-0521884389, 53–63 s.
- [39] Škrabánek, P.: *Teorie fuzzy množin a její aplikace* [online]. Pardubice: Univerzita Pardubice, 2014, ISBN 978-80-7395-875-6, [cit. 2018-04-08]. Dostupné z: [https://www.researchgate.net/publication/309010508\\_Teorie\\_fuzzy\\_mnoziny\\_a\\_jejich\\_aplikace](https://www.researchgate.net/publication/309010508_Teorie_fuzzy_mnoziny_a_jejich_aplikace)



- 
- [40] Murphy, K. P.: An Introduction to Graphical Models [online]. Technická zpráva, 2001, [cit. 2018-04-09]. Dostupné z: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.29.1410>
- [41] Koski, T.; Noble, J.: *Bayesian Networks: An Introduction*. New Jersey: Wiley Publishing, první vydání, 2009, ISBN 978-0470743041, 58–63 s.
- [42] Tang, Y.; Pan, W.; Li, H.; aj.: Fuzzy Naive Bayes classifier based on fuzzy clustering. In *IEEE International Conference on Systems, Man and Cybernetics*, ročník 5, Nevada: IEEE, október 2002, ISSN 1062-922X.
- [43] Wagholikar, K. B.; Vijayraghavan, S.; Deshpande, A. W.: Fuzzy naive bayesian model for medical diagnostic decision support. In *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Nevada: IEEE, september 2009, ISSN 1094-687X, s. 3409–3412. Dostupné z: <http://ieeexplore.ieee.org/document/5332468>
- [44] Program Dithyron. *Imunoanalytické metody v diagnostice* [online], ročník 2, č. 5, 1998, [cit. 2018-04-21]. Dostupné z: <http://oldweb.lfp.cuni.cz/journals/imj/1998/5/>
- [45] Jřina, M.; Pecen, L.; Novák, J.: Diagnostic software for decision support of detection and interpretation of tumor markers. *IJBH*, 2015: s. 1–4.
- [46] Borland International: *Turbo Pascal User's Guide Version 5.0*. 1989.
- [47] Pecen, L.: *DITHYRON* [software]. [prístup 2018-03-20].
- [48] STAPRO. O společnosti. *Stapro.cz* [online]. [cit. 2018-04-21]. Dostupné z: <http://www.stapro.cz/spolecnost/>
- [49] SourceMaking. AntiPatterns. *Sourcemaking.com* [online]. [cit. 2018-04-21]. Dostupné z: <https://sourcemaking.com/antipatterns>
- [50] Lessa, A.: *Python Developer's Handbook*. Indianapolis: Sams Publishing, první vydání, 2000, ISBN 978-0672319945.
- [51] Maio, I.: *Building Web Applications with Flask*. Birmingham: Packt Publishing Ltd., první vydání, 2015, ISBN 978-1-78439-615-2.
- [52] Grinberg, M.: *Flask Web Development*. Sebastopol: O'Reilly Media, první vydání, 2014, ISBN 978-1-449-37262-0.
- [53] The Apache Software Foundation. Apache HTTP server project. *httpd.apache.org* [online]. [cit. 2018-04-25]. Dostupné z: <https://httpd.apache.org/>
- [54] Python Software Foundation. mod\_wsgi 4.6.4. *Pypi.org* [online]. [cit. 2018-04-25]. Dostupné z: [https://pypi.org/project/mod\\_wsgi/](https://pypi.org/project/mod_wsgi/)

- [55] SQLAlchemy. The Python SQL Toolkit and Object Relational Mapper. *Sqlalchemy.org* [online]. [cit. 2018-04-25]. Dostupné z: <https://www.sqlalchemy.org/>
- [56] Open Source: *Bootstrap 3.3 documentation*. [online]. [cit. 2018-04-25]. Dostupné z: <http://getbootstrap.com/docs/3.3/>
- [57] Open Source: *Flask-Login* [online]. [cit. 2018-04-25]. Dostupné z: <https://flask-login.readthedocs.io/en/latest/>
- [58] Open Source: *Flask-Babel* [online]. [cit. 2018-04-25]. Dostupné z: <https://pythonhosted.org/Flask-Babel/>
- [59] ALMA DIAGNOSTICS. *Alma Diagnostics* [software]. [prístup 2018-04-20]. Dostupné z: <http://almadiagnostics.com/>
- [60] Gamma, E.; Helm, R.; Johnson, R.; aj.: *Design Patterns: Elements of Reusable Object-Oriented Software*. Boston: Addison-Wesley, prvné vydání, 1994, ISBN 978-0201633610.
- [61] Mercurial. *Mercurial* [software]. [prístup 2018-05-03]. Dostupné z: <https://www.mercurial-scm.org>
- [62] Corel Corporation. *CorelDRAW Graphics Suite 2018* [software]. [prístup 2018-04-10]. Dostupné z: <https://www.coreldraw.com>
- [63] Slavík, V.: *Poedit* [software]. [prístup 2018-04-25]. Dostupné z: <https://poedit.net/>
- [64] Purcell, S.: *PyUnit* [software]. [prístup 2018-04-20]. Dostupné z: <http://pyunit.sourceforge.net/>
- [65] NumPy developers. NumPy. *NumPy.org* [online]. [cit. 2018-05-06]. Dostupné z: <http://www.numpy.org/>

---

## Zoznam použitých skratiek

- Admin** Administrátor
- antiTG** Protilátky proti tyreoglobulínu
- antiTPO** Protilátky proti tyreoperoxidáze
- antiTSHR** Protilátky proti TSH receptoru
- CDSS** Clinical decision support system
- CT** Computed tomography
- DRY** Don't repeat yourself
- EDIS** Emergency Department Information Systems
- EHR** Electronic health records
- fT3** Volný tyroxín
- fT4** Volný trijódtyronín
- GUI** Graphical user interface
- JSON** JavaScript Object Notation
- ORM** Object-relational mapper
- HTML** Hypertext Modeling Language
- HTTP** Hypertext Transfer Protocol
- HTTPS** HTTP Secure
- IDE** Integrated Development Environment
- k*-nn** Metóda *k* najbližších susedov

## A. ZOZNAM POUŽITÝCH SKRATIEK

---

- MR** Magnetic resonance
- MS-DOS** Microsoft Disk Operating System
- MVC** Model-View-Controller
- PDF** Portable Document Format
- rT3** Reverzný tyroxín
- rT4** Reverzný trijódtyronín
- SVM** Support vector machines
- SQL** Structured Query Language
- TBG** Tyroxín viažúci globulín
- TG** Tyreoglobulín
- TRH** Tyreoliberín
- TSH** Tyreoidálny stimulačný hormón
- TT3** Tyroxín
- TT4** Trijódtyronín
- URI** Uniform Resource Identifier
- WSGI** Web Server Gateway Interface
- XML** Extensible markup language

---

## Zoznam vstupných markerov a výstupných diagnóz programu DITHYRON

**Vstupné markery:** TSH, TT3, fT3, rT3, TT4, fT4, rT4, TG, TBG, antiTPO, antiTG, antiTSHR a TSH po TRH teste.

**Výstupné diagnózy v anglickom jazyku:**

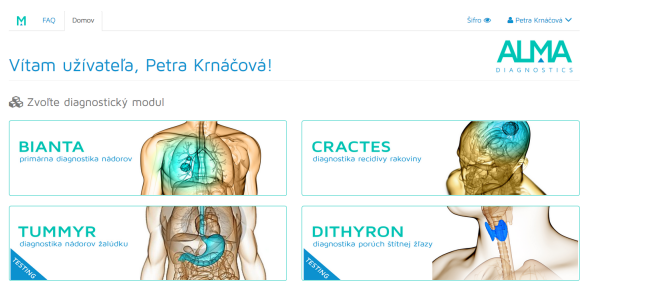
- Normal thyroidal function (euthyroidism)
- Idiopathic hypothyroidism (Idiopathic myxedema)
- Hypothyroidism caused by therapy or injure of thyroid
- Subclinical hypothyroidism (oligosymptomatic)
- Hypothyroidism caused by low iodine intake
- Hypophysal (secondary) hypothyroidism
- Hypothalamic hypothyroidism (tercial hypofunction)
- Graves-Basedow hyperthyroidism (toxic diffuse goiter)
- Toxic goiter (uninodular or multinodul goiter )
- T3 thyrotoxicosis (pholicular hyperfunction)
- Hyperfunction caused by therapy or injure
- Hyperfunction during pregnancy
- Normal function during pregnancy, treatment by estrogenes, contraception

B. ZOZNAM VSTUPNÝCH MARKEROV A VÝSTUPNÝCH DIAGNÓZ PROGRAMU DITHYRON

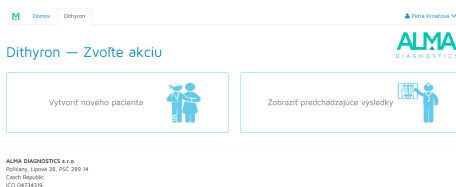
---

- Other tumors (not thyroidal and hypophysal)
- Hypophysal tumor
- Thyroiditis Hashimoto (diffuse lymphoid thyroiditis)
- Subacute tyreoiditis
- Low proteins
- Low T3 syndrome
- Thyroid carcinoma
- Treated thyroid carcinoma or thyrotoxicoses therapy by radioiodine
- Subclinical hyperthyroidism (thyrotoxicosis)
- Serious or critical non-thyroidal (and non-psychiatric) illness
- Psychiatric illness
- Low TBG
- Familial dysalbuminemic hyperthyroxinemia
- Treatment by 1-thyroxin Rx
- Treatment by glucocorticoids
- Treatment by propranolol
- Treatment by amiodarone
- Resistance to thyroid hormone
- Treatment by androgens
- Liver failure or hepatitis
- Nefrotic syndrome
- Treatment by T3
- Treatment by phenytoin
- Neonatal and phetal thyrotoxicosis
- Tyreotoxicosis factia
- Secondary hypothyrodism caused by TRH or TSH insufficiency
- Increased serum TBG concentration

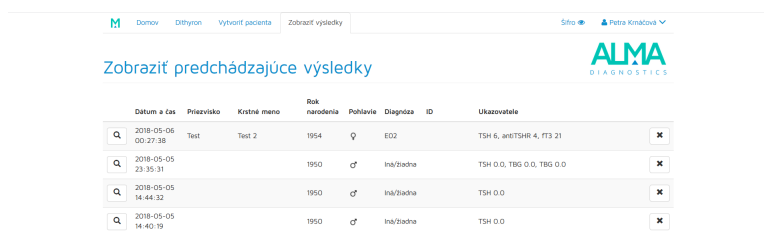
## Užívateľské rozhranie modulu



Obr. C.1: Ukážka rozhrania modulu Dithyron: Voľba modulu



Obr. C.2: Ukážka rozhrania modulu Dithyron: Voľba akcie



Obr. C.3: Ukážka rozhrania modulu Dithyron: Správa vyšetrení

## C. UŽÍVATELSKÉ ROZHRAŇIE MODULU

**Vytvoriť nového pacienta**

Príezvisko:

Krídlo meno:

ID:

Pohlavie:  Muž  Žena

Rok narodenia:

Diagnóza:

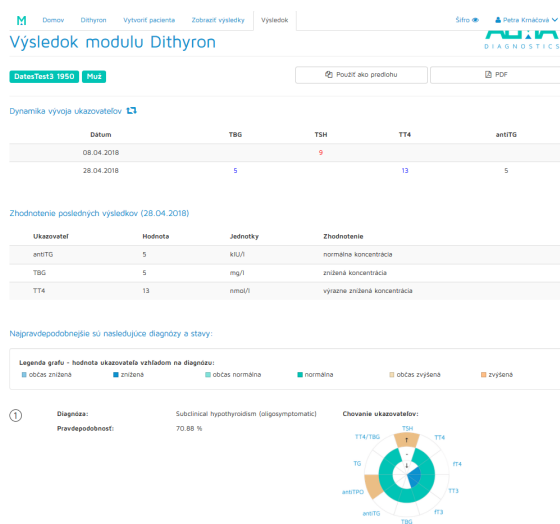
| Marker        | Dátum      | Hodnota [mIU/l] |
|---------------|------------|-----------------|
| TSH (bazálne) | 06.05.2018 | 3               |
| TBG           | 06.05.2018 | 0               |
| TBG           | 06.05.2018 | 0               |

Obr. C.4: Ukážka rozhrania modulu Dithyron: Vytvorenie nového pacienta

**Nastavenie hodnôt referenčných intervalov**

| Ukazovateľ                | Referenčný interval (Min - Max) |
|---------------------------|---------------------------------|
| TSH (mg/l)                | 5.6 - 18.5                      |
| TG (mg/dl)                | 0.0 - 85                        |
| TT4 (mg/dl)               | 7 - 30                          |
| TSH (mIU/l) - Muži - 0-6  | 0.52 - 6                        |
| TSH (mIU/l) - Muži - 6-11 | 0.47 - 5.01                     |
| TSH (mIU/l) - Ženy - 0-6  | 0.52 - 5.5                      |
| antiTPO (mIU/l)           | 0.0 - 51                        |
| antiTPO (mIU/l)           | 0.0 - 123                       |
| antiTSH (mIU/l)           | 0.0 - 140                       |
| FT3 (pmol/l) - Muži - 0-6 | 3.1 - 10.6                      |
| FT3 (pmol/l) - Muži - 6-9 | 3.7 - 10.3                      |
| FT3 (pmol/l) - Muži - 9-9 | 4.4 - 9.2                       |

Obr. C.5: Ukážka rozhrania modulu Dithyron: Nastavenie hodnôt referenčných intervalov



Obr. C.6: Ukážka rozhrania modulu Dithyron: Výstup



## Scenáre integračného testovania

| <b>Scenár 1 – Správa vyšetrení</b> |                                                  |                                                                                       |
|------------------------------------|--------------------------------------------------|---------------------------------------------------------------------------------------|
| <b>Krok</b>                        | <b>Užívateľská akcia</b>                         | <b>Očakávaný výsledok</b>                                                             |
| 1                                  | Kliknutie na zoznam vyšetrení.                   | Zobrazenie výsledkov vyšetrení z databázy (vrátane vyšetrení užívateľov zo skupiny).  |
| 2                                  | Kliknutie na konkrétne vyšetrenie.               | Zobrazenie výsledkov vyšetrenia.                                                      |
| 3                                  | Návrat na zoznam vyšetrení.                      | Zobrazenie rovnakých výsledkov ako v kroku 1.                                         |
| 4a                                 | Kliknutie na zmazanie vyšetrenia.                | Archivácia vyšetrenia v databáze, vrátenie zoznamu neobsahujúceho zmazané vyšetrenie. |
| 4b                                 | Kliknutie administrátora na zmazanie vyšetrenia. | Archivácia vyšetrenia v databáze, zobrazenie vyšetrenia v časti archív.               |
| 5b                                 | Kliknutie na zmazanie archivovaného vyšetrenia.  | Vymazanie vyšetrenia z databázy, odstránenie vyšetrenia z časti archív.               |
| <b>Priebeh testovania</b>          |                                                  |                                                                                       |
| Scenár prebehol bez problémov.     |                                                  |                                                                                       |

Tabuľka D.1: Kroky a vyhodnotenie scenára správy vyšetrení

D. SCENÁRE INTEGRAČNÉHO TESTOVANIA

| <b>Scenár 2 – Vyšetrenie pacienta</b> |                                                                                                              |                                                                                       |
|---------------------------------------|--------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| <b>Krok</b>                           | <b>Užívateľská akcia</b>                                                                                     | <b>Očakávaný výsledok</b>                                                             |
| 1                                     | Kliknutie na tlačidlo vyšetrenie pacienta.                                                                   | Zobrazenie formuláru na zadanie informácií o pacientovi.                              |
| 2                                     | Vyplnenie informácií o pacientovi a kliknutie na tlačidlo pokračovať.                                        | Uloženie informácií do session objektu, zobrazenie formulára na zadávanie meraní.     |
| 3                                     | Pridanie merania tlačidlom pridať.                                                                           | Zobrazenie komponentu pre ďalšie meranie, uloženie existujúcich meraní do session.    |
| 4                                     | Pridanie 3 meraní pomocou tlačidla +3.                                                                       | Zobrazenie 3 komponentov pre ďalšie meranie, uloženie existujúcich meraní do session. |
| 5                                     | Vyplnenie meraní.                                                                                            |                                                                                       |
| 6                                     | Odstránenie ľubovoľného merania tlačidlom x.                                                                 | Odstránenie merania z UI aj zo session.                                               |
| 7                                     | Kliknutie na tlačidlo potvrdiť.                                                                              | Uloženie vyšetrenia do databázy, zobrazenie výsledkov vyšetrenia užívateľovi.         |
| 8                                     | Kliknutie na tlačidlo PDF.                                                                                   | Vytvorenie PDF s výsledkami vyšetrenia.                                               |
| 9                                     | Kliknutie na tlačidlo použiť ako predlohu.                                                                   | Zobrazenie predvyplneného formulára na zadávanie vyšetrení.                           |
| 10                                    | Kliknutie na tlačidlo potvrdiť.                                                                              | Uloženie vyšetrenia do databázy, zobrazenie rovnakých výsledkov ako v kroku 7.        |
| <b>Priebeh testovania</b>             |                                                                                                              |                                                                                       |
| <b>Krok</b>                           | <b>Chyba</b>                                                                                                 | <b>Oprava</b>                                                                         |
| 2                                     | Pri zadávaní informácií a stlačení klávesy enter sa aktualizoval formulár na prázdny.                        | Znemožnenie odosielania formulárov pomocou klávesy enter.                             |
| 7                                     | Do databázy sa neuložil celý reťazec zadanej diagnózy, pretože použitý <code>varchar</code> bol príliš malý. | Zväčšenie <code>varcharu</code> .                                                     |
| 8                                     | V slovenskej lokalizácii PDF sa reťazec „Predchádzajúca diagnóza“ nezmestil do bunky.                        | Zväčšenie bunky.                                                                      |

Tabuľka D.2: Kroky a vyhodnotenie scenára vyšetrenia pacienta

| <b>Scenár 3 – Import údajov a zmena jazyka</b> |                                                                                            |                                                                                                                                                                    |
|------------------------------------------------|--------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Krok</b>                                    | <b>Užívateľská akcia</b>                                                                   | <b>Očakávaný výsledok</b>                                                                                                                                          |
| 1                                              | Kliknutie na tlačidlo vyšetrenie pacienta.                                                 | Zobrazenie formuláru na zadanie informácií o pacientovi.                                                                                                           |
| 2                                              | Nahratie súboru s výsledkami pacienta.                                                     | Vyplnenie formulára hodnotami zo súboru.                                                                                                                           |
| 3                                              | Zmena jazyka na anglický.                                                                  | Preloženie rozhrania do angličtiny.                                                                                                                                |
| 4                                              | Kliknutie na tlačidlo potvrdiť.                                                            | Uloženie vyšetrenia do databázy, zobrazenie výsledkov vyšetrenia užívateľovi v angličtine.                                                                         |
| <b>Priebeh testovania</b>                      |                                                                                            |                                                                                                                                                                    |
| <b>Krok</b>                                    | <b>Chyba</b>                                                                               | <b>Oprava</b>                                                                                                                                                      |
| 3                                              | Zadaná diagnóza pacienta sa nepreložila, ale resetovala na východziu hodnotu v angličtine. | Do session objektu a databázy neukladať lokalizovaný reťazec diagnózy. Ukladať len kód diagnózy, ktorý je rovnaký vo všetkých jazykoch. Upraviť príslušné funkcie. |

Tabuľka D.3: Kroky a vyhodnotenie scenára importu údajov zo súboru a zmeny jazyka

| <b>Scenár 4 – Nastavenie hodnôt referenčných intervalov</b> |                                                                                   |                                                                                                                                                    |
|-------------------------------------------------------------|-----------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Krok</b>                                                 | <b>Užívateľská akcia</b>                                                          | <b>Očakávaný výsledok</b>                                                                                                                          |
| 1                                                           | Kliknutie na tlačidlo pre nastavenie hodnôt referenčných intervalov               | Zobrazenie formuláru na nastavenie hodnôt referenčných intervalov.                                                                                 |
| 2                                                           | Vyplnenie nejakých hodnôt a kliknutie na tlačidlo zrušiť.                         | Neuloženie zmien do databázy.                                                                                                                      |
| 3                                                           | Vyplnenie nejakých hodnôt a kliknutie na tlačidlo uložiť.                         | Uloženie zmien do databázy.                                                                                                                        |
| 4                                                           | Vyšetrenie pacienta.                                                              | Zobrazenie výsledkov, ktoré berú do úvahy novo nastavené referenčné intervaly.                                                                     |
| <b>Priebeh testovania</b>                                   |                                                                                   |                                                                                                                                                    |
| <b>Krok</b>                                                 | <b>Chyba</b>                                                                      | <b>Oprava</b>                                                                                                                                      |
| 4                                                           | Vyšetrenie pacienta bolo vykonané s východzími hodnotami referenčných intervalov. | Opraviť chybu v rámci triedy <code>DithyronComputation</code> , kde sa vytvárali nové cut-off hodnoty namiesto toho, aby sa brali tie z užívateľa. |

Tabuľka D.4: Kroky a vyhodnotenie scenára nastavenia hodnôt referenčných intervalov

| <b>Scenár 5 – Nastavenie výstupu algoritmu administrátorom</b> |                                                                                            |                                                                       |
|----------------------------------------------------------------|--------------------------------------------------------------------------------------------|-----------------------------------------------------------------------|
| <b>Krok</b>                                                    | <b>Užívateľská akcia</b>                                                                   | <b>Očakávaný výsledok</b>                                             |
| 1                                                              | Zmena počtu diagnóz a prahu, do ktorého majú byť diagnózy vydávané v konfiguračnom súbore. |                                                                       |
| 2                                                              | Vyšetrenie pacienta.                                                                       | Zobrazenie výsledkov vyšetrenia, ktoré rešpektujú zmenené nastavenia. |
| 3                                                              | Zmena konfigurácie algoritmu.                                                              |                                                                       |
| 4                                                              | Vyšetrenie toho istého pacienta.                                                           | Zobrazenie výsledkov vyšetrenia, ktoré rešpektujú zmenené nastavenia. |
| <b>Priebeh testovania</b>                                      |                                                                                            |                                                                       |
| Scenár prebehol bez problémov.                                 |                                                                                            |                                                                       |

Tabuľka D.5: Kroky a vyhodnotenie scenára nastavenia výstupu administrátorom

---

## Obsah priloženého CD

|                     |                                                                 |
|---------------------|-----------------------------------------------------------------|
| readme.txt.....     | stručný popis obsahu CD                                         |
| src                 |                                                                 |
| ├ impl .....        | zdrojové kódy implementácie                                     |
| ├ thesis.....       | zdrojová forma práce vo formáte L <sup>A</sup> T <sub>E</sub> X |
| text .....          | text práce                                                      |
| ├ thesis.pdf .....  | text práce vo formáte PDF                                       |
| ├ prirucka.pdf..... | užívateľská príručka vo formáte PDF                             |