

ŽILINSKÁ UNIVERZITA V ŽILINE

FAKULTA RIADENIA A INFORMATIKY

BAKALÁRSKA PRÁCA

PETER RENDEK

**Podpora rozhodovania pre prideľovanie prostriedkov
obsluhy zákazníkov**

Vedúci práce: Mgr. Lýdia Gábrišová, PhD., KMMOA, FRI, ŽU

Registračné číslo: 470/2014

Žilina, 2017

ŽILINSKÁ UNIVERZITA V ŽILINE

FAKULTA RIADENIA A INFORMATIKY

BAKALÁRSKA PRÁCA

INFORMATIKA:

PETER RENDEK

**Podpora rozhodovania pre pridelovanie prostriedkov
obsluhy zákazníkov**

Žilinská univerzita v Žiline

Fakulta riadenia a informatiky

Školiace pracovisko : Žilina

Žilina, 2017

Čestné vyhlásenie

Prehlasujem, že som bakalársku prácu napísal samostatne pod odborným vedením vedúcej bakalárskej práce a že som uviedol všetky použité zdroje a literatúru, z ktorých som čerpal.

V Žiline 10 mája 2017

Peter Rendek

Podpis: _____

Pod'akovanie

Veľmi pekne ďakujem vedúcej mojej bakalárskej práce Mgr. Lýdii Gábrišovej, PhD. za jej trpezlivosť, odbornú pomoc a užitočné pripomienky a cenné rady, ktoré mi poskytovala v priebehu vypracovávaní bakalárskej práce.

ABSTRAKT V ŠTÁTNOM JAZYKU

RENDEK, Peter: *Podpora rozhodovania pre pridelovanie prostriedkov obsluhy zákazníkov*. [Bakalárska práca]. – Žilinská univerzita Žilina. Fakulta riadenia a informatiky; Katedra matematických metód a operačnej analýzy. – Mgr. Lýdia Gábrišová, PhD., KMMOA, ŽU – Žilina: FRI ŽU, 2017. 58 s.

Cieľom bakalárskej práce je vytvoriť užívateľské prostredie, ktoré je určené na podporu rozhodovania pri pridelovaní prostriedkov zákazníkom v prevádzke verejného obslužného systému. Konkrétne pre Operačné stredisko záchranej zdravotnej služby (OS ZZS) Trenčianskeho samosprávneho kraja. S využitím prístupných dát o rozmiestnení stredísk záchranej zdravotnej služby (ZZS) a nemocníc, ktoré poskytujú zdravotnú službu, vytvorím užívateľské prostredie, ktoré by využívali operátori, jednotlivé strediská ZZS a osoby, ktoré majú na starosti príjem zo zásahov v jednotlivých zdravotníckych zariadeniach Trenčianskeho kraja. Táto aplikácia bude poskytovať informačnú podporu pre operátorov OS ZZS a veliteľov staníc ZZS, ktorí telefonicky alebo použitím vysielacky manažujú záchranu pacientov prostredníctvom aktuálnych dostupných posádok staníc ZZS. Dané prostredie zároveň umožní ukladanie záznamov o každom pacientovi a o priebehu príslušného zásahu. Daná téma ma oslovila, lebo jej realizácia je aplikovateľná pre podporu rozhodovania pri pridelovaní prostriedkov zákazníkov v prevádzke iných logistických systémov.

Kľúčové slová: podpora rozhodovania, informačný systém, Záchranná zdravotná služba

ABSTRAKT V CUDZOM JAZYKU

RENDEK, Peter: *Decision support for allocating resources serving customers*. [Bachelor's Thesis]. – The University of Žilina. Faculty of Management Science & Informatics; Department of Mathematical Methods and Operations Research – Mgr. Lýdia Gábrišová, PhD., KMMOA, ŽU – Žilina: FRI ŽU, 2017. 58 s.

The aim of the bachelor's thesis is to create a user environment, which is designed to support the decision making process of assignment of resources to customers in the public service system. Specifically in the operations centre of Emergency Health (Medical) Services (OC EHS) of District of Trenčín. Using accessible data on the positions of Emergency Health (Medical) Service (EHS) stations and hospitals providing health services, I will create a user environment for operators, particular EHS stations, and those in charge of admittance from interventions in particular health facilities in the District of Trenčín. This application will provide information support services for operators OC EHS and for the head of EHS stations who manage patient rescue by telephone or using a two-way radio via the current available EHS crews. The user environment also allows you to store medical records of each patient and the course of each particular action. I found this theme interesting because of its applicability to the decision making process of services to customers in the operation of other different logistics systems.

Key words: (support decision making, information system, Emergency Health Service)

Obsah

Zoznam obrázkov	9
Zoznam tabuliek	11
Zoznam skratiek	12
Úvod	14
1 Aktuálny stav	15
1.1 Záchranná zdravotná služba	15
1.2 Priebeh zásahu	16
1.3 KOS ZZS Trenčín	17
1.4 Nástroje pre analýzu a návrh	19
1.5 Nástroje a technológie pre tvorbu softvéru	20
1.5.1 Programovací jazyk	20
1.5.2 Java	21
1.5.3 Vývojové prostredie	22
1.6 Nástroje a technológie pre ukladanie dát	23
1.6.1 SQL Server 2012 Enterprise Edition	25
1.6.2 Nástroje pre vytvorenie databázovej schémy	25
1.7 Spracovanie geografických údajov	26
1.8 Nástroje pre administratívne výstupné správy	26
2 Ciele práce	28
3 Metodika práce	29
3.1 Analýza prevádzky ZZS	29
3.2 Analýza priebehu zásahu	30
3.3 Požiadavky na obslužný systém ZZS	32
3.4 Navrhovanie databázy	34
4 Realizácia	38
4.1 Vstupné dáta	38
4.2 Implementácia	39
4.2.1 Grafika užívateľského prostredia	39
4.2.2 Prepojenie s databázou	41
4.2.3 Využitie JasperReports šablón	42
4.2.4 Použitie Google Maps Rozhraní	44
4.3 Grafická užívateľská príručka	47
5 Výsledky práce a diskusia	53
Záver	55
Zoznam použitých zdrojov	57

Prílohy	58
Príloha A: Obsah DVD	58

Zoznam obrázkov

Obrázok 1 Trenčiansky kraj rozmiestenie staníc ZZS (4)	18
Obrázok 2 Diagramy UML(5)	20
Obrázok 3 SPRÁVA O ZÁSAHU [PDF formulár]	27
Obrázok 4 Hlavný aktéri ZZS	30
Obrázok 5 Diagram aktivít operátora KOS ZZS	30
Obrázok 6 Stavové diagramy stavu posádky [vľavo], stavu zásahu [vpravo]	31
Obrázok 7 Vývojový diagram riadenia zásahu	31
Obrázok 8 Diagram prípadov použitia ZZS	33
Obrázok 9 Schéma databázy informačného systému ZZS	37
Obrázok 10 Farebné rozlišovanie riadkov JTable podľa hodnoty bunky riadku	40
Obrázok 11 Metóda slúžiaca na naplnenie JComboBox-u dátami z databázy	40
Obrázok 12 Real-time napĺňanie ComboBox-u podľa podreťazca	41
Obrázok 13 Metóda, ktorá vracia model ComboBox-u na základe zdieľanej procedúry databázy [vpravo dole]	41
Obrázok 14 Real-time vyhľadávanie miest a obcí	41
Obrázok 15 Metóda DBConnection() na vytvorenie prepojenia s DB	42
Obrázok 16 Prepojenie na databázu z iReport Designer-a	42
Obrázok 17 Sql select skript pre výjazdový formulár	43
Obrázok 18 Navrhovanie JasperReports šablóny	43
Obrázok 19 Šablóna výjazdového formulára	44
Obrázok 20 Práca so zostavou	44
Obrázok 21 Požiadavka na Google Distance Matrix	45
Obrázok 22 Načítanie JSON výstupu z Google API	46
Obrázok 23 Rozparsovanie získaného JSON výstupu	46
Obrázok 24 Prihlasovaco-odhlasovací formulár ZZS	47
Obrázok 25 Vytvorenie zásahu operátorom KOS ZZS	48
Obrázok 26 Správa zásahov operátora KOS ZZS	49
Obrázok 27 Oprava/doplňanie údajov k zásahu	49
Obrázok 28 Okno veliteľa záchranky [nový zásah - červený]	50
Obrázok 29 Okno veliteľa záchranky [spracovanie nového rozkazu]	50
Obrázok 30 Okno veliteľa záchranky [nastavenie prevozu]	50

Obrázok 31 Okno veliteľ záchrany Administrácia [generovanie dochádzky].....	51
Obrázok 32 Okno lekár na príjme v nemocnici	51
Obrázok 33 Okno lekár otvorené na Raspberry.....	52
Obrázok 34 Prihlasovanie sa do posádky na Raspberry	53
Obrázok 35 Filtre informačného systému.....	54

Zoznam tabuliek

Tabuľka 1 : ZZS v Trenčianskom kraji	19
--	----

Zoznam skratiek

ZZS	Záchranná zdravotná služba
OS	Operačné stredisko
OS ZZS	Operačné stredisko záchrannej zdravotnej služby
KOS ZZS	Krajské operačné stredisko záchrannej zdravotnej služby
OS ZZS SR	Operačné stredisko záchrannej zdravotnej služby Slovenskej republiky
RLP	Rýchla lekárska pomoc (zloženie posádky: vodič, lekár, záchranár)
RZP	Rýchla zdravotná pomoc (zloženie posádky: vodič, záchranár)
VZZS	Vrtuľníková zdravotná záchranná služba
IZS	Integrovaný záchranný systém
HaZZ	Hasičský a záchranný zbor
s.r.o.	Spoločnosť s ručením obmedzeným
n.o.	Nezisková organizácia
OO	Objektovo orientované
OOP	Objektovo orientované programovanie
UML	Unified Modeling Language (grafický jazyk modelovanie)
SQL	Structured Query Language (štruktúrovaný vyhľadávací jazyk v DB)
T-SQL	Trancast SQL (procedurálne rozšírenie SQL od Microsoft-u)
DDL	Data Definition Language (časť SQL na definovanie dát)
DML	Data Manipulation Language (časť SQL na manipuláciu s dátami)
DAS	Data Access Language
DIS	Data Integrity Language
DB	Databáza
DBS	Databázový systém
DBMS	Database management system, (SRBD)Systém riadenia bázy dát
TDM	Toad Data Modeler (nástroj na modelovanie DB)
JDBC	Java Database Connectivity (rozhranie na prácu s DB v Jave)
JDK	Java Develepment Kit (súbor nástrojov na programovanie v Jave)
JRE	Java Runtime Enviroment (rozhranie pre beh programu v Jave)
JVM	Java Virtual Machine (virtuálny stroj na spracovanie java-bajtkódu)
JAVA SE	Java Standard Edition

WWW	web, World Wide Web (celosvetová sieť)
TCP/IP	Transmission Control Protocol/Internet Protocol
URL	Uniform Resource Locator (Jednotný vyhľadávač zdrojov)
GPS	Global Positioning System (Globálny lokalizačný systém)
API	Application programming interface (rozhranie pre tvorbu aplikácií)
XML	eXtensible Markup Language (rozšírený značkovací jazyk)
JSON	JavaScript Object Notation (JavaScriptový objektový zápis)

Úvod

Rozhodovanie v logistických systémoch (napr. obslužných, dopravných), kde ide o zabezpečenie požiadaviek zákazníkov, možno podporiť informačným systémom. Ten by predstavoval užívateľské prostredie pre operatívne a zároveň optimálne pridelovanie prostriedkov obsluhy všetkých zákazníkov. Manažment systému má pritom k dispozícii obslužné strediská, ktoré tieto služby uskutočňujú prostredníctvom staníc obsluhy.

V danej práci budem analyzovať reálny obslužný systém a navrhmem užívateľské prostredie pre podporu rozhodovania v operačnom stredisku záchranej zdravotnej služby v Trenčianskom kraji. V teoretickej časti bakalárskej práce, najskôr vysvetlím nevyhnutné termíny potrebné pre správne pochopenie problematiky ZZS a následne popíšem najdôležitejšie procesy spojené s jej činnosťou. Stručne popíšem a zdôvodnením použítie užitočných pomocných nástrojov a nevyhnutných technológií, pre vytvorenie obslužného systému. Popíšem požiadavky, ktoré daný informačný systém bude musieť spĺňať.

Praktická časť bakalárskej práce bude zameraná na riešenie problémov vznikajúcich počas navrhovania a tvorby informačného systému. Hotový informačný systém bude tvorený dvomi hlavnými časťami. Najdôležitejšou časťou bude databáza (MS SQL server 2012), nevyhnutne potrebná na ukladanie a spracovávanie dát. Druhou časťou informačného systému bude samotný program, ktorým budú jednotliví užívatelia pristupovať k dátam uloženým v databáze. Hlavnými funkciami informačného systému bude vytvorenie zásahu pomocou výberu najvhodnejšej posádky ZZS a následne riadenie takto vytvoreného zásahu. Ostatné funkcie informačného systému pomôžu jednotlivým účastníkom zásahu ľahko získať užitočné informácie, čo uľahčí prácu s bežnou administratívou.

V prvej kapitole sa budem zaoberať východiskovým stavom, rozoberiem štruktúru ZZS, opíšem priebeh zásahu a popíšem nástroje a technológie, ktoré by sa dali využiť pri tvorbe systému. V druhej kapitole si stanovím a popíšem ciele bakalárskej práce. Tretia kapitola bude zameraná na metodiku práce, od analýzy problematiky ZZS až po návrh jednotlivých častí obslužného systému pomocou diagramov UML. V štvrtej kapitole popíšem programátorsky najzaujímavejšie časti informačného systému a podrobne predstavím funkcionality vytvoreného programu. V poslednej kapitole zhrniem a vyhodnotím výsledky bakalárskej práce

1 Aktuálny stav

1.1 Záchranná zdravotná služba

Záchranná zdravotná služba (ZZS) poskytuje osobám (ďalej pacientom) neodkladnú prednemocničnú zdravotnú starostlivosť, ktorým náhla zmena zdravotného stavu ohrozuje život alebo niektorú základnú životnú funkciu. Bez urgentného poskytnutia takejto zdravotnej starostlivosti pacientom, môže hroziť vážne poškodenie zdravia, neznesiteľná bolesť. Tiež vplyvom rýchlej zmeny pacientovho správania, aj všeobecné ohrozenie seba, ako aj jeho bezprostredného okolia. O záchrannú zdravotnú starostlivosť na Slovensku sa starajú nasledujúce orgány :

- **Operačné stredisko záchrannej zdravotnej služby (OS ZZS)**

OS ZZS Slovenskej republiky je štátna príspevková organizácia, ktorú zriaďuje Ministerstvo zdravotníctva SR. Jej hlavnou úlohou je organizovať, synchronizovať a analyzovať všetky zložky ZZS, ktoré zabezpečujú neustálu a plynulú zdravotnú starostlivosť. OS ZZS ďalej zabezpečuje príjem tiesňového volania, ktoré následne spracuje a v prípade potreby ho zrealizuje. Nasleduje zásahový rozkaz pre príslušného poskytovateľa zdravotnej starostlivosti. Počas priebehu takto vytvoreného zásahu, operačné stredisko poskytuje technicko-informačnú podporu, dohliada na jeho správny priebeh a zaobstaráva dokumentáciu zásahu. Medzi ostatné funkcie OS patrí dozor nad materiálno-technickým vybavením, odbornou prípravou svojich zamestnancov, poskytovaním rôznych kurzov a iných úloh v rámci aktuálnej legislatívy. Organizačná štruktúra OS ZZS SR je rozdelená na :

- **Riaditeľstvo OS ZZS SR** - je riadiaca organizačná jednotka, ktorá plní úlohu riadenia, kooperácie a vyhodnocovania ZZS v rámci SR a aj pri poskytovaní cezhraničnej zdravotnej starostlivosti.

- **8 krajských operačných stredísk (KOS ZZS)** - sídlia v každom krajskom meste SR a ich základnou úlohou je prijímanie, spracovávanie a následné vyhodnotenie tiesňového hovoru. Tieto úlohy sú úzko spojené s celkovou koordináciou v rámci SR, aby bola zabezpečená nepretržitá a plynulá ZZS. Všetky KOS ZZS majú rovnakú personálnu štruktúru s rozdielnym počtom pracovníkov podľa potrieb daných regiónov.

- **Poskytovatelia zdravotnej starostlivosti**

Aktuálne na Slovensku pôsobí 13 poskytovateľov ZZS, ktorí tvoria základnú zložku integrovaného záchranného systému. Ich hlavnou úlohou je zriadenie stanice (záchranky) ZZS a jej následné riadenie. Podľa zákona č. 579/2004 Z. z. článku 1 § 2 odsek 3 : (1) „*Stanica záchranej zdravotnej služby je stacionárne pracovisko, na ktorom sa zdržiava zásahová skupina záchranej zdravotnej služby a ambulancia záchranej zdravotnej služby mimo času výjazdu; nie je vytvorená na príjem tiesňového volania a je vybavená komunikačnými prostriedkami v neprerušenej prevádzke na spojenie s operačným strediskom záchranej zdravotnej služby.*“ Pre dokonalé pochopenie je potrebné uviesť, že ambulanciou ZZS sa chápe ako vozidlo (sanitka) s posádkou. Posádky ZZS poznáme :

- RLP - rýchla lekárska pomoc, kde posádku tvorí vodič, lekár a záchranár,
- RZP - rýchla zdravotná pomoc, kde posádku tvorí vodič a záchranár.

V súčasnosti je možné, že na jednej stanici sídli viac ako 1 posádka rôznych typov. Príkladom je (2) stanica ZZS v Bratislave, kde sa nachádzajú 2 posádky RLP, 3 posádky RZP alebo (3) stanica v Trenčíne, ktorá má 2 posádky RZP. O nepretržitú pohotovosť dbá sieť ZZS SR, ktorú tvorí 92 ambulancií RLP, 181 ambulancií RZP a tiež 7 ambulancií vrtuľníkovej zdravotnej záchranej služby (VZZS).

1.2 Priebeh zásahu

KOS ZZS, ktorý prostredníctvom linky 115 alebo 112 prijíma tiesňové volania, vyhodnocuje ich a v prípade potreby zásahu, kontaktuje najbližšiu voľnú posádku záchranky. Uvedie vybranej záchranke adresu miesta zásahu a všetky dostupné a potrebné informácie o zdravotných ťažkostiach pacienta. Po prijatí výzvy z dispečingu (OS ZZS) na zásah, musí byť posádka sanitky schopná kedykoľvek, v dennú alebo nočnú hodinu

vyraziť na miesto zásahu, a to najneskôr do dvoch minút. Spojenie dispečingu a jednotlivých posádok záchranky je zabezpečené trojitým spôsobom. Prvý spôsob je použitie vysielacky cez rádiokomunikačnú sieť SITNO. O zvyšné spôsoby komunikácie sa starajú dve nezávislé mobilné siete.

Po príchode záchranky na miesto zásahu sa musí posádka ohlásiť dispečingu. V prípade, ak nastane problém pri lokalizovaní pacienta, môže OS ZZS navigovať posádku prostredníctvom GPS monitoringu, alebo tiež priamo prepojiť volajúceho so zasahujúcou posádkou. Na mieste zásahu prebieha vyšetrenie, liečba a následné vypísanie zdravotnej dokumentácie. Ak sa pacientov stav v priebehu zásahu RZP naďalej zhoršuje, OS ZZS môže vyslať k pacientovi aj posádku RLP. V prípade potreby, záchranári požiadajú o pomoc vrtuľník.

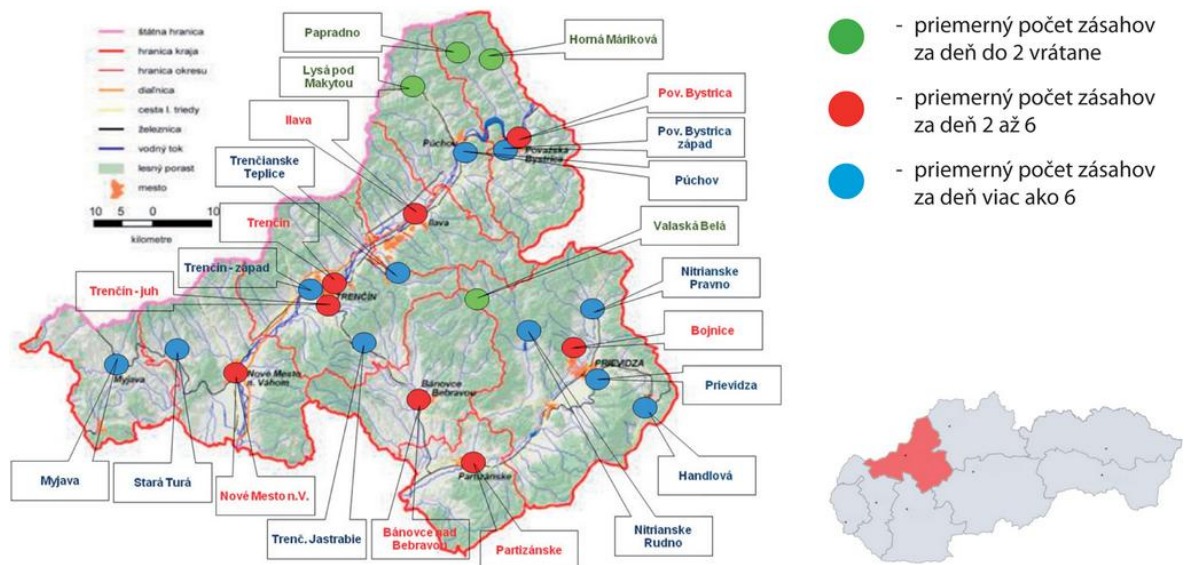
Po ukončení zásahu, nasleduje v prípade potreby transport pacienta do najbližšej nemocnice alebo špecializovaného zdravotníckeho zariadenia. V aktuálne zaužívanom systéme najlepšiu cestu do nemocnice vyberá vodič sanitky. Záchranár alebo lekár poskytujú starostlivosť pacientovi v ambulantnej časti vozidla. Posádka zároveň hlási dispečingu, kde pacienta transportujú.

Odovzdávanie pacienta zdravotníckemu zariadeniu prebieha podľa pravidiel zdravotníckeho zariadenia. Vo veľkých nemocniciach záchranári odovzdajú pacienta na centrálnom prijíme. V menších zariadeniach odovzdajú pacienta priamo na konkrétnom oddelení podľa pacientovej diagnózy. Po návrate na svoje stanovisko, záchranka potvrdí OS ZZS pripravenosť na ďalší zásah.

1.3 KOS ZZS Trenčín

Trenčianske operačné stredisko ZZS bolo založené 1.6.2006 a odvtedy sa zaoberá riadením a koordinovaním aktivít ZZS, hlavne na území Trenčianskeho samosprávneho kraja. Trenčiansky kraj leží v západnej časti Slovenskej republiky. Na západe susedí s Českou republikou, na severovýchode so Žilinským krajom, na juhovýchode s Banskobystrickým krajom, na juhu s Nitrianskym a na juhozápade s Trnavským krajom. Rozlohou približne 4 502 km² je tretím najmenším krajom SR, s počtom 599 867 obyvateľov a hustotou zaľudnenia 133,2 obyv./km². Trenčiansky kraj sa skladá z 9 okresov, 18 miest a 258 obcí. Pre názornosť uvádzam na obrázku 1 rozmiestnenie

staníc ZZS (modré a červené body) s ich priemerným počtom zásahov za deň podľa analýzy portálu HMA za rok 2010.



Obrázok 1 Trenčiansky kraj rozmiestenie staníc ZZS (4)

O zdravie a záchranu životov občanov Trenčianskeho samosprávneho kraja sa stará 21 záchranných stredísk, ktoré majú spolu 26 sanitiek s 13-timi posádkami RLP a 13-timi posádkami RZP. Na jednu sanitku pripadá približne 22 850 obyvateľov, čo v rámci SR je viac ako priemer. K dispozícii je súčasne 9 zdravotníckych zariadení, do ktorých môžu byť pacienti prevezení a sú to:

1. Nemocnica Bánovce - 3. súkromná nemocnica, s.r.o.
2. Nemocnica s poliklinikou Prievidza
3. Nemocnica Handlová - 2. súkromná nemocnica, s.r.o.
4. Nemocnica s poliklinikou Ilava, n.o.
5. Nemocnica s poliklinikou Myjava
6. Nemocnica s poliklinikou Nové Mesto nad Váhom, n.o.
7. Nemocnica s poliklinikou Partizánske, n.o.
8. Nemocnica s poliklinikou Považská Bystrica
9. Fakultná nemocnica Trenčín

Informácia o situovaní jednotlivých záchranných staníc v Trenčianskom kraji, počte a type posádok sanitiek a o počte obyvateľov, ktoré prináležia k jednotlivým posádkam záchranky sú uvedené v tabuľke 1.

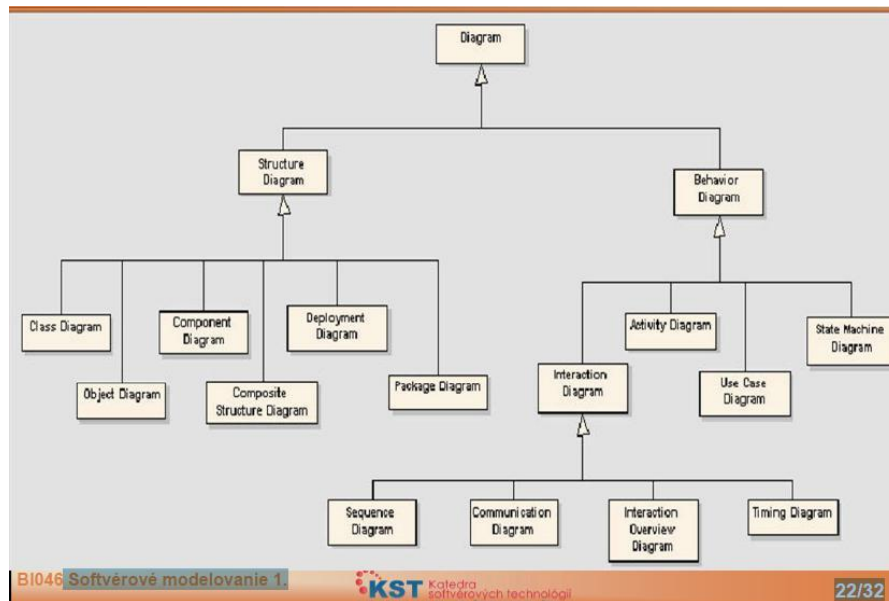
Tabuľka 1 : ZZS v Trenčianskom kraji

Domovské miesto sanitky	Obyvateľov	Počet sanitiek	Typ
Nové Mesto nad Váhom	53 700	1	RLP
Stará Turá	14 300	1	RZP
Myjava	19 000	1	RLP
Brezová pod Bradlom	6 500	1	RZP
Ilava	20 700	1	RZP
Dubnica nad Váhom	51 800	1	RLP
Trenčianske Teplice	7 200	1	RZP
Bánovce nad Bebravou	33 600	1	RLP
Ostratice	14 500	1	RZP
Trenčín + okolie	80 600	4	2 RLP +2 RZP
Púchov	35 900	1	RLP
Beluša	13 900	1	RZP
Handlová	20 900	1	RLP
Partizánske	42 000	1	RLP
Nitrianske Rudno	10 400	1	RZP
Bojnice	12 600	1	RLP
Prievidza	53 500	1	RZP
Nováky	22 400	1	RLP
Nitrianske Pravno	12 000	1	RZP
Považská Bystrica	59 700	3	2 RLP +1 RZP
Trenčianske Jastrabie	9 000	1	RZP

1.4 Nástroje pre analýzu a návrh

V nasledujúcej kapitole uvediem a stručne popíšem možné softvérové technológie pre vytvorenie užívateľského prostredia pre manažment zásahov ZZS. Pre vývoj tejto aplikácie nestačí len programátorské prostredie, ale sú potrebné aj nástroje, ktoré súvisia s návrhom riešení pre tvorby programu a návrhu databázy.

Pre popis návrhu, špecifikáciu, vizualizáciu a dokumentáciu softvéru sa všeobecne používa grafický jazyk UML (Unified Modeling Language), ktorý podporuje objektovo orientovaný prístup k analýze. Medzi najpoužívanéjšie časti UML štandardu patria diagramy, ktorých hierarchiu ukazuje nasledujúci obrázok 2:



Obrázok 2 Diagramy UML(5)

Pre tvorbu a prácu s UML, existuje veľa komerčných, aj nekomerčných nástrojov. Väčšina komerčných nástrojov je veľmi profesionálna a rozsiahla a pre efektívne používanie je nutné hlbšie štúdium. Výsledkom použitia týchto nástrojov je urýchlenie fázy návrhu programu. Patrí sem napríklad Rational Rose, aktuálne vyvíjaný firmou IBM Corporation, ktorý podporuje jazyky C, C++, Java. Softvérom s podobnými vlastnosťami a funkciami je aj I-Logix Rhapsody, ktorý napríklad využíva NASA a US Army. Enterprise Architect od spoločnosti Sparx Systems. Je komplexným nástrojom na modelovanie systémov, ktorý poskytuje podporu pre celý tím vývojárov, počas vývoju programu. K nekomerčným voľne šíriteľným nástrojom patrí Violet, ktorý je vhodný na výuku softvérového inžinierstva, OOP, ale v reálnej praxi, kde je potrebné generovanie zdrojového kódu z UML (tu jeho použitie vhodné nie je).

1.5 Nástroje a technológie pre tvorbu softvéru

1.5.1 Programovací jazyk

Jednou z dvoch najhlavnejších technológií, pre tvorbu softvéru je programovací jazyk. V súčasnosti je voľne k dispozícii veľké množstvo programovacích jazykov. Každý programovací jazyk má svoje špecifiká, ktoré vyplývajú zo svojho zamerania. Preto možno povedať, že každý jazyk je vhodný na niečo iné. Do úvahy v mojej práci, by pripadali len objektovo orientované jazyky. Sú založené na využívaní dátových štruktúr, ktoré sú nazývané objekty a ich interakcie na vývoj aplikácií. Medzi najznámejšie takého jazyky patrí C++, C#, Java.

Jazyk C++ sa nepovažuje za čisto objektovo orientovaný jazyk, lebo podporuje viacero programovacích štýlov, ako napríklad procedurálne a generické programovanie. Výhodou jazyka C++ je široká využiteľnosť, blízkosť k hardvéru a rýchlosť behu programu, ktorý vyplýva zo skutočnosti, že zdrojový kód napísaný v C++ sa najprv prekladá do strojového kódu a až potom sa môže spustiť. Nevýhodou však je jeho zložitosť, z ktorej tiež vyplýva časová náročnosť písania kódu. Jazyk neobsahuje automatickú správu pamäti, ale umožňuje používať viacnásobnú dedičnosť (na rozdiel od ďalej popísaných jazykov).

Jazyk C# je elegantný a moderný, vysokoúrovňový OOP jazyk, vytvorený firmou Microsoft na vytváranie aplikácií spustiteľných v .NET framework-u, ktorý je súčasťou každého systému Windows. Cieľom vývojárov bolo, vytvoriť podobne silný nástroj ako C++, ale doplniť ho o možnosť rýchleho programovania, ako napríklad v Jave. Jazyk C# na rozdiel od C++ nepodporuje viacnásobnú dedičnosť a neexistujú v ňom globálne metódy a premenné, ale už obsahuje automatickú správu pamäte.

Po zvážení všetkých výhod a nevýhod konkrétnych programovacích jazykov, som sa nakoniec rozhodol pre Javu. Túto voľbu najviac ovplyvnil fakt, že som mal s ňou doteraz najviac programátorských skúseností. Na internete je voľne k dispozícii dostupná a veľmi kvalitná technická podpora s množstvom návodov na riešenie problémov, ktoré by sa počas programovania mohli vyskytnúť.

1.5.2 Java

Tento programovací jazyk vytvorili Patrik Naughton, Chris Warth, Ed Frank, Mike Sheridan na čele s B.SC Jamesom Arthurom Goslingom zo spoločnosti Sun Microsystems. Po 1,5 ročnej fáze vývoja jazyka, sa mal nazývať Oak, ale vzhľadom k okolnosti, že už existoval jazyk s rovnakým názvom, bol tento produkt predstavený verejnosti 23. mája 1995 ako Java. Hlavným impulzom vzniku Javy, bola potreba trhu pre platformovo nezávislého programacieho jazyka, určeného na programovanie rôznej spotrebnej elektroniky. Aj keď sa dal program napísaný v jazyku C++ skompilovať na takmer každý procesor, bolo však potrebné mať C++ kompilátor pre daný typ procesora. Keďže vytváranie kompilátorov bolo časovo náročné a nákladné, tím okolo Jamesa Goslinga rozhodol daný problém vyriešiť.

Riešením problému s prenositeľnosťou programov je, že výstupom kompilátoru Javy nie je priamo spustiteľný kód. Výstupom kompilátora je takzvaný bajtkód, čo je vlastne vysoko optimalizovaná sada inštrukcií, navrhnutá tak, aby ju bolo možné spustiť

v interpretéri Java bajtkódu (JVM - Java Virtual Machine). Takéto riešenie umožní, že pre dané zariadenie stačí pripraviť iba JVM, ktorý už sám spracuje java bajtkód. Ďalšou výhodou je lepšia bezpečnosť počítača, kedy program napísaný v Jave, je riadený JVM a ten môže programu zabrániť vykonávať činnosti mimo tohto virtuálneho priestoru. Enormnú rýchlosť rozvoja Javy ovplyvnili hlavne Internet a WEB, kde Java našla svoje ďalšie využitie, lebo Internet je tvorený distribuovaným priestorom, v ktorom sa nachádzajú rôzne zariadenia s rôznymi operačnými systémami a rôznymi procesormi.

Úmyslom tvorcov jazyka Java bolo odvodiť množstvo vlastností od jazykov C a C++. Boli si vedomí, že ak preberú dobre známu syntax jazyka C a objektovo orientované vlastnosti z jazyka C++, ktoré následne doplnia (automatická správa pamäte), vylepšia a zjednodušia (odstránenie ukazovateľov, bezznamienkových číselných typov, príkazov), stane sa ich jazyk ľahko naučiteľný pre programátorov, ktorí pred tým používali tieto jazyky. Ostatné najdôležitejšie vlastnosti jazyka sú :

- **Robustnosť** – zahŕňa kontrolu dátových typov pri kompilácii, automatickú správu pamäte (Garbage collection), OO spracovanie výnimiek.
- **Podpora multithreading** - programy, ktoré robia naraz viac úloh, riešenie synchronizácie procesov.
- **Výkonnosť** – technológia HotSpot, ktorá obsahuje kompilátor JIT (Just-In-Time), zabezpečujúci spracovanie aktuálne potrebného bajtkódu do spustiteľného kódu.
- **Distribuovanosť** - využitie pre internet, podpora TCP/IP, URL.
- **Dynamickosť** - program zavádza triedy do pamäti podľa potreby.
- **Bezpečnosť** – ochranné mechanizmy voči škodlivému kódu.

1.5.3 Vývojové prostredie

Je softvér, ktorý programátorom uľahčuje prácu pri procese vývoja programu od písania až po jeho debugovanie a testovanie. Pri rozhodovaní výberu vývojového prostredia je dôležité zohľadniť v akom programovacom jazyku chceme programovať, lebo žiadne vývojové prostredie nie je určené pre všetky programovacie jazyky. Medzi najznámejšie Open Source vývojové prostredia jazyka Java, patria Netbeans (Oracle Corporation), Eclipse (Eclipse Foundation) a najznámejší komerčný je IntelliJ IDEA.

1.6 Nástroje a technológie pre ukladanie dát

Pre každý informačný systém je dôležité správne vyriešiť problém ukladania, triedenia a spracovávania dát. Prvý spôsob vyriešenia tohto problému je vytvorenie technológie svojho vlastného súborového systému pomocou textových súborov. Tento spôsob pre rozsiahly systém nepripadal do úvahy, lebo by musel programátor sám správne navrhnuť všetky použité údajové štruktúry, operácie, atď. Tiež by musel navrhnuť systém pre manipulovanie s uloženými dátami. Dané riešenie by si vyžadovalo extrémne navýšenie programátorského času, potrebného na vytvorenie celého informačného systému. Lepší spôsob riešenia problému by bolo použiť priamo existujúcu databázovú technológiu. Základné definície databázových technológií a databázových systémov podľa literatúry Základy databázových systémov (6) sú :

[Definícia 1: Databázový systém] „(DBS) tvorí databáza (DB) a systém riadenia bázy dát (SRBD). Zjednodušene môžeme napísať: $DBS = DB + SRBD$.“ (6 s. 24)

[Definícia 2: Databáza] „(DB) je množina perzistentných dát, ktorá je používaná v aplikačnom systéme.“ (6 s. 26) Hlavné požiadavky na databázu sú :

- Oddelenie definície dát a príkazov pre manipuláciu
 - Nezávislosť dát
 - Procedurálne a neprocedurálne rozhranie
 - Minimalizácia redundancie dát
 - Ochrana proti nekonzistentnosti dát
 - Zdieľanie dát
 - Bezpečný prístup k dátam
 - Integritu dát.“
- (6 s. 29)

[Definícia 3: Systém riadenia bázy dát] „(SRBD) je množina programov zabezpečujúcich manipuláciu s dátami, ochranu dát, paralelné spracovanie a pod., ktorá hlavne zabezpečuje:

- Definíciu dát
 - Manipuláciu s dátami
 - Riadi prístup
 - Integritu databáz
 - Obnovu databázy
 - Paralelný prístup
 - Ochranu dát
 - Riadenie katalógov
 - Ostatné služby a nástroje.“
- (6 s. 41)

[Definícia 4: Databázová technológia] „je unifikovaný súbor pojmov, prostriedkov a techník, slúžiaca pre vytváranie informačných systémov.“ (6 s. 19)

Dôvody využitia databázovej technológie, priamo vyplývajú z vyššie uvedených základných definícií a funkcií databázového systému ako celku. Pre prácu s dátami v databáze, programátor používa SQL príkazy, ktoré spracováva procesor jazykov DML a DDL. SQL (Structured Query Language) je aktuálne najpoužívanejší štruktúrovaný dopytovací jazyk v relačných databázach, ktorý je tvorený :

- DDL (Data Definition Language) a slúži na definíciu dát a štruktúr,
- DML (Data Manipulating Language) a slúži na mazanie, vkladanie a upravovanie dát,
- DAS (Data Access Language) na definovanie prístupových práv,
- DIS (Data Integrity Language) na zabezpečenie integrity dát.

Väčšina moderných databázových systémov, už pri inštalácii ponúka možnosť inštalácie vlastných nástrojov určených na vývoj a svoju správu. Napríklad svoj SQL editor kódu, správcu konfigurácie, atď. Na trhu je aktuálne veľké množstvo takýchto databázových technológií, preto je dôležité určiť, čo od databázy potrebujeme a či nám to daná databáza poskytne.

Medzi najobľúbenejšie Open Source databázy patrí MySQL, ktorá je využívaná hlavne v oblasti webových aplikácií a Internetu. Programovací jazyk PHP a MySQL databáza sú základné komponenty pre tvorbu redakčných systémov (CMS), ako napríklad WordPress, phpBB alebo MediaWiki. Užitočné nástroje pre spravovanie MySQL sú :

- MySQL QueryBrowser (vývojové prostredie),
- MySQL Administrator (konfiguračné prostredie).

Databázy Oracle sa používajú prevažne v komerčnom priestore z dôvodu potreby licencií, ktoré sú pomerne drahé, ale na druhej strane, tieto databázy sa môžu pochváliť vysokým výkonom, vyspelými technikami spracovania dát s výbornou dokumentáciou. Užitočným nástrojom na spravovanie Oracle databáz je Oracle SQL Developer (vývojové prostredie).

Pri výbere vhodnej databázovej technológie treba zohľadňovať viaceré kritéria, medzi ktoré patria očakávané požiadavky na databázu. Napríklad technická špecifikácia, efektívnosť a tiež dostupnosť technológie, s ktorou samozrejme úzko súvisí aj cena.

MS SQL Servery patria medzi spoplatnené relačné databázové systémy, vyvíjané spoločnosťou Microsoft, ktoré sú určené pre najnáročnejších používateľov.

1.6.1 SQL Server 2012 Enterprise Edition

Je kompletná dátová platforma, využitelná pre splnenie všetkých požiadaviek podnikových aplikácií, ako napríklad online spracovanie transakcií v dátových centrách (skladoch). Vďaka tejto edícii je tiež možné nastavovať servery a realizovať online veľké množstvo operácií súvisiacich so spracovaním dát. Taktiež tento server obsahuje technológiu, ktorá chráni uložené dáta pred ľudskými chybami a samozrejme zabezpečuje aj obnovu databázy pri prípadnej havárii. Technológia je tiež využitelná ako lokálny virtuálny server. SQL server 2012 nie je momentálne najnovší MS SQL server, ale je to posledná verzia, ktorá sa dala lokálne nainštalovať na operačných systém Windows 7. Túto databázovú technológiu som si zvolil hlavne preto, lebo v budúcnosti by som sa chcel venovať rozvoju informačného systému, ktorého jadro tvorí novšia verzia tohto servera. Užitočné nástroje pre spravovanie s MS SQL serverom sú :

- SQL Server Management Studio je grafické prostredie pre prácu a správu s rôznych databázových serverov od spoločnosti Microsoft. Obsahuje prostredia pre správu DB serveru, ako aj prostredie pri vytváraní a ladení SQL príkazov. V tomto prostredí je možné písať a následne testovať funkcie, procedúry, sekvencie a pod., pomocou procedurálneho rozhrania založeného na jazyku T-SQL (Transact-SQL).
- SQL Server Configuration Manager je nástroj, ktorý sa využíva pre nastavenie základných konfiguračných úloh súvisiacich so službami spúšťania, zastavovania a reštartovania databázovej technológie, ako aj nastavovania rôznych protokolov.

1.6.2 Nástroje pre vytvorenie databázovej schémy

Na vytvorenie databázovej schémy sa v praxi dajú použiť nasledujúce postupy:

- Prvý náročnejší postup je pomocou ručného zdĺhavého písania SQL príkazov pre definovanie dát (DDL).

- Ďalší jednoduchší spôsob postupu je použiť grafický nástroj (napr. DbDesigner, Toad Data Modeler, atď.) podporujúci danú databázu, ktorej schému navrhujeme. Pomocou takéhoto nástroja dokáže programátor rýchlo vytvoriť vyhovujúci model, z ktorého sám program vygeneruje skript databázovej schémy, aj s minimálnymi poznatkami z oblasti syntaxe SQL jazyka.

1.7 Spracovanie geografických údajov

Pre potrebu práce so vzdialenosťami medzi dvomi geografickými bodmi boli k dispozícii nasledujúce možnosti:

Prvou z nich bolo využitie matice vzdialeností miest a obcí SR. Matica vzdialeností je maticový zápis, kde indexy riadkov a stĺpcov sú uzly (mestá, obce) cestnej siete a prvky matice vyjadrujú vzdialenosť medzi uzlami. Nevýhodou je, že poskytuje len údaje o najkratšej cestnej vzdialenosti medzi dvoma uzlami a neobsahuje zmeny v cestnej sieti SR, ktoré sa udiali za posledné roky.

Druhou možnosťou je použitie priamo nástroja databázovej technológie pre prácu s geografickými dátami. Táto dokáže, na základe GPS súradníc vypočítať vzdušnú vzdialenosť medzi bodmi. Nevýhodou je však skutočnosť, že letecká vzdialenosť by sa dala použiť len na približný odhad cestnej vzdialenosti a v niektorých prípadoch by však mohla byť mimoriadne zavádzajúca.

Poslednou možnosťou je využitie externého nástroja, ktorý je priamo určený na zisťovanie cestnej vzdialenosti medzi 2 bodmi. Tie môžu byť popísané buď GPS súradnicami alebo presnou adresou miesta. Medzi takého nástroje môžeme zaradiť Open Source projekt OpenStreetMap, ktorý je určený na tvorbu geografických dát, ako napríklad cestných máp. Známejším nástrojom od spoločnosti Google sú však rozhrania Google Maps, ktoré poskytujú webové služby pre prácu s geografickými dátami a následne vracajú vypracované dáta (vo formáte XML alebo JSON).

1.8 Nástroje pre administratívne výstupné správy

MS SQL Server 2012 pomocou nástroja SQL Server Reporting Service umožňuje vytváranie prehľadných zostáv, doplnených rôznymi prvkami, napríklad grafmi, mapami, tabuľkami, atď. Pre tvorbu takýchto reportov sa využívajú nástroje ako napríklad Report

Builder. Poskytuje intuitívne prostredie podobné MS Office, ktorý umožní vytvoriť report aj menej skúsenému používateľovi.

JasperReports je open source nástroj na báze Javy, určený na vytváranie rôznych tlačových zostáv. Na vytváranie tlačovej zostavy sa používa, buď program iReport Designer alebo iReport plugin do Netbeans-u. Spracovávané dáta sa dajú vyberať z rôznych zdrojov, napr. z databázy, CSV, XLS. Dajú sa doplniť o rôzne parametre, ktoré môžu zahŕňať výpočtový výraz. Spomenuté nástroje, ďalej podporujú funkcie pre nastavovanie formátu strany, sekcií, atď. Medzi základné prvky tvoriace tlačovú zostavu patria statické texty, textové polia, obrázky, rôzne grafy, tabuľky, čiarové kódy a pod. Výstupným formátom takto vytvoreného dokumentu môžu byť PDF, DOCX, XML, XLS.

SPRÁVA O ZÁSAHU					
LSE - Life Star Emergency, spol. s.r.o. - Trenčín					
NEMOCNICA S POLIKLINIKOU POVAŽSKÁ BYSTRICA					
Číslo zásahu :		3179		Rodné číslo	
Dátum a čas volania :		03/05/2017 00:12:28		123699/8445	
Aktéri zásahu			Pacient : Alexandra Pekná		
Vodič zásahu :	Iveta Kákošová		OBEC :	Nová Dubnica	
Lekár zásahu :	RZP		Adresa :	Janka Kráľa 1	
Záchranár zásahu :	Lucia Biela		Info :	Bolesti v hrudníku, ťažké dýchanie spôsobené pádom na motorke	
Hovor prijal :	Jana Marcinčinová				
Prijal :	Marta Belescakova				
Uzavrel:	Jana Marcinčinová				
Potenciálna Diagnóza :			S22.- Zlomenina rebra (rebier), hrudnej kosti a hrudníka		
Konečná Diagnóza :			NEZADANA		
Popis zásahu :					
<p>Netradičná dopravná nehoda sa odohrala na križovatke ulíc Špitálska a 29. augusta vo štvrtok popoludní. Zdravotníci i hasiči mali namierené k tomu istému požiaru na Dunajskej ulici. Obe záchranárske autá uháňali so zapnutými majákmi a húkačkami, mali teda právo prednosti v jazde.</p> <p>„Také vozidlo nemusí dodržiavať jednotlivé ustanovenia zákona. Musí však jazdiť tak, aby neohrozilo iné vozidlá,“ tvrdí dopravný odborník Jozef Drahovský. Podľa neho zákon nijakým spôsobom nerieši, ktoré z vozidiel má v takom prípade prednosť. Príčinou je, že nikto nepredpokladá, že k takejto situácii by vôbec mohlo dôjsť.</p> <p>Drahovský je napriek tomu presvedčený, že jeden zo šóferov zlyhal. „Keď niekto ide na červenú, hoci má maják, mal by dávať podstatne väčší pozor,“ dodal.</p> <p>Aj podľa ďalšieho experta Jána Bazovského mal byť šófer, ktorý vchádzal do križovatky na červenú, oveľa opatrnejší. „Z laického</p>					
Výjazd	Príjazd k pacientovi	Výjazd od pacienta	Príjazd nemocnica	Začiatok návratu	Príjazd na stanicu
03/05/2017 00:24:11	03/05/2017 00:24:30	03/05/2017 00:24:44	03/05/2017 01:44:31	03/05/2017 01:44:44	03/05/2017 01:44:55

Obrázok 3 SPRÁVA O ZÁSAHU [PDF formulár]

2 Ciele práce

Cieľom bakalárskej práce je navrhnúť a následne vytvoriť informačný systém pre podporu rozhodovania pri prideľovaní prostriedkov zákazníkom v prevádzke verejného obslužného systému. Výsledný informačný systém bude určený pre krajské operačné stredisko záchranej zdravotnej služby v Trenčíne, ktorý by sa dal v prípade potreby ľahko doplniť dátami a následne použiť aj v iných krajských operačných strediskách ZZS a prípadne aj na riaditeľstve ZZS. Tento informačný systém bude poskytovať užívateľské prostredie a informačnú podporu pre riadenie zásahov jednotlivým aktérom zásahu.

Informačný systém uľahčí operátorom operačných stredísk výber najvhodnejšej posádky na zásah a súčasne im poskytne vhodný nástroj pre riadenie zásahov. Tak môže zabezpečiť plynulú a nepretržitú kooperáciu jednotlivých uzlov záchranného systému. Umožní jednotlivým nemocniciam a záchrankám aktívnejšie sa zapájať do informačného systému záchranej služby, z ktorého môžu následne čerpať a dopĺňať informácie, užitočné pri záchrane pacientov. Systém ďalej zabezpečí ukladanie záznamov o každom pacientovi, informácie o jednotlivých zásahoch a iných dôležitých skutočnostiach, ktoré by v budúcnosti mohli byť využité pri optimalizácii činností záchranej zdravotnej služby. Hotový informačný systém ZZS bude tiež použiteľný ako užitočný nástroj, uľahčujúci bežnú administráciu jednotlivých uzlov záchranného systému.

Potrebné informácie som čerpal z oficiálnej stránky Záchranej zdravotnej služby SR a Ministerstva zdravotníctva. Dáta o vzdialenostiach medzi jednotlivými mestami a obcami Trenčianskeho kraja sú online na web stránke Katedry matematických metód a operačnej analýzy FRI ŽU. (7)

Po zozbieraní a spracovaní dát a požiadaviek pre informačný systém ZZS, zanalyzujem postup realizácie, vymedzím kľúčové termíny, dôležité vlastnosti a príslušné diagramy UML, ktoré popíšu možné reálne fungovanie ZZS. Po vytvorení a naplnení databázy potrebnými dátami, navrhmem prezentačnú vrstvu aplikácie, ktorú následne doplním o potrebnú funkcionality informačného systému.

Po otestovaní informačného systému a opravení zistených chýb, ktoré môžu vzniknúť počas programovania, systém získa svoju finálnu podobu a bude možné ho otestovať v praxi počas skúšobnej prevádzky ZZS. To ukáže, či je systém, použiteľný, a či nie je potrebné ho doplniť o ďalšie funkcionality.

3 Metodika práce

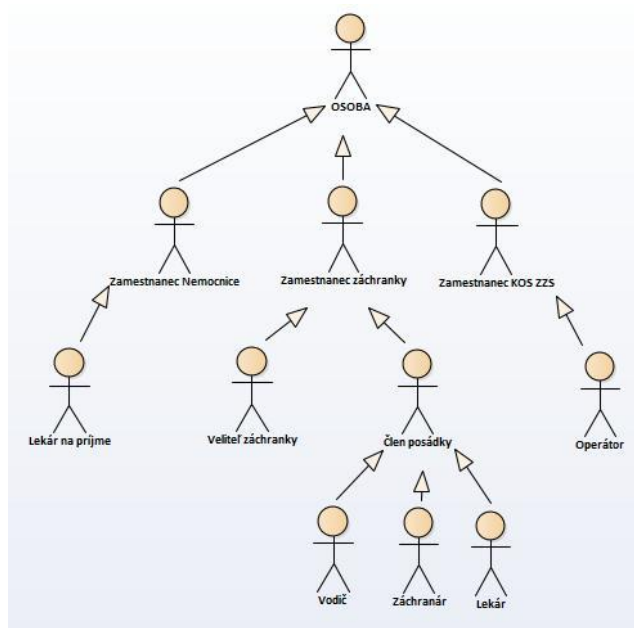
3.1 Analýza prevádzky ZZS

O zabezpečenie ZZS sa neustále starajú operačné strediská ZZS, jednotliví poskytovatelia ZZS a nemocnice. Zamestnaneckú štruktúru KOS ZZS Trenčín tvorí približne 50 vysoko kvalifikovaných profesionálnych zamestnancov. Zamestnanecká štruktúra je zložená z 37 zdravotných operátorov, a to 5 vedúcich smien, 5 technických pracovníkov, 1 referent, 1 vedúci zdravotníckych operátorov a 4 lekári –operátori. Smenu v operačnej sále vždy tvorí 6 zdravotníckych operátorov, 1 vedúci operátor smeny a jeden operátor – lekár. V Trenčianskom kraji, ako aj vo väčšine ostatných krajov SR, je obvyklé, že v rovnakej budove (sále) sídli aj dispečing iných zložiek Integrovaného záchranného systému (IZS) – operační dôstojníci Hasičského a záchranného zboru (HaZZ), 1 operátor tiesňovej linky a hlavný operátor IZS. Pri návrhu informačného systému ZZS som činnosť ďalších operátorov IZS nezahrnul. V prípade potreby by sa dalo systém rozšíriť.

Sieť staníc ZZS SR, tvorí 13 súkromných poskytovateľov, ktorí zabezpečujú vykonávanie zásahov u pacientov v zmysle platnej legislatívy. Keďže každý poskytovateľ ZZS je súkromná spoločnosť, tak jednotlivé vybavenia staníc (sanítiek), interné predpisy a zamestnanecká štruktúra sa môžu líšiť. Jednotlivé spoločnosti v rámci konkurenčného boja uvádzajú len minimálne množstvo informácií o ich fungovaní. Z toho dôvodu systém bude hlavne pracovať len s pracovníkmi posádky. V prípade typu RZP posádku tvoria: vodič a záchranár a pri type RLP posádku tvoria: vodič, záchranár a lekár. Riadenie ľubovoľnej stanice ZZS zabezpečuje veliteľ stanice.

Dôležitým článkom systému ZZS sú nemocničné zariadenia, ktoré sú v systéme zastúpené pracovníkom zodpovedným za prebratie pacientov a to je lekár na príjme.

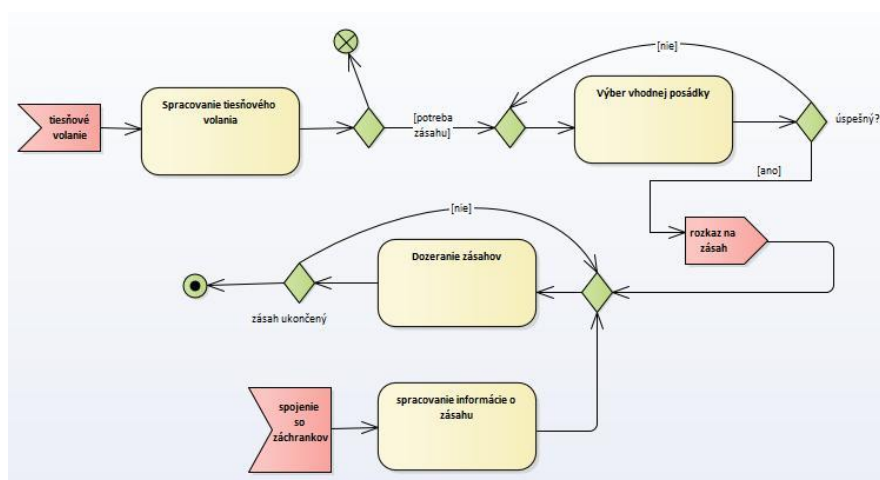
Na obrázku 4 sú pomocou časti diagramu prípadov použitia zobrazené najdôležitejšie osoby, ktoré v ZZS figurujú.



Obrázok 4 Hlavný aktéri ZZS

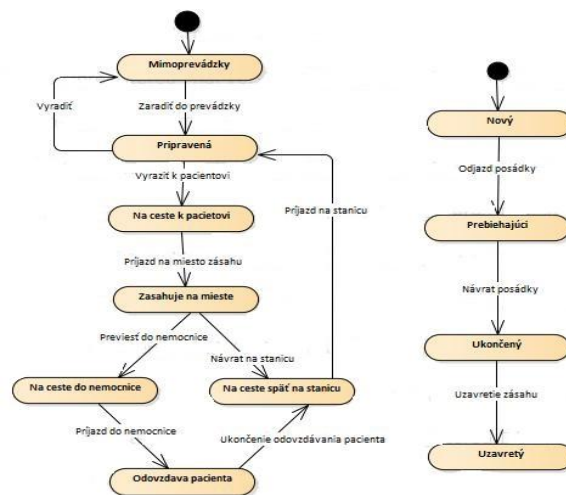
3.2 Analýza priebehu zásahu

Najdôležitejšie úlohy operátora KOS ZZS je obsluhovanie tiesňových volaní, kedy v prípade potreby poskytuje pacientovi alebo volajúcej osobe inštrukcie prvej pomoci. Počas priebehu obsluhy volania, operátor vyhodnotí dôležitosť a závažnosť volania a určí, či je v danom prípade potreba vykonať zásah. V prípade nutnosti vykonania zásahu, operátor vyberie vhodnú posádku zo systému, vydá jej zásahový rozkaz a poskytne nevyhnutné informácie pre zásah. Po ukončení tiesňového volania, operátor pomocou komunikácie s posádkou aktualizuje informačný systém a doplní dostupné informácie. Činnosti operátora KOS ZZS sú zhrnuté v nasledujúcom diagrame aktivít na obrázku 5.

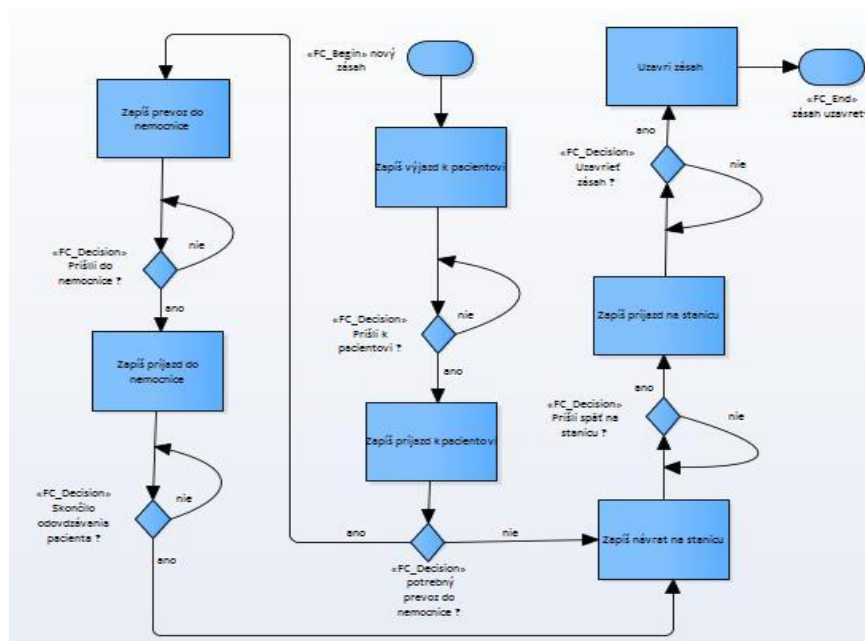


Obrázok 5 Diagram aktivít operátora KOS ZZS

Po prijatí rozkazu z dispečingu KOS ZZS, posádka odchádza na miesto zásahu. Počas jazdy k pacientom, pre sanitku ZZS neplatia bežné cestné pravidlá a obmedzenia a vozidlo so zapnutou svetelnou signalizáciou má právo prednostnej jazdy. Na základe týchto skutočností, budem považovať hodnoty získané z Google Maps rozhrania s nastavením režimu pre jazdu v bežnej premávke, za vhodné pre odhad najhoršieho času potrebného pre absolvovanie tejto trasy. Po príchode záchranky na miesto zásahu, posádka ohlásí príchod operátorovi a ten to následne zapíše do informačného systému. Podobným spôsobom fungujú aj ohlasovania dispečingu ostatných príjazdov a odjazdov. Na nižšie uvedených diagramoch (viď. Obrázok 6 a 7) je priebeh spravovania zásahu z pohľadu operátora a ako vplyvajú jednotlivé operácie operátora na zmeny stavu posádky a zásahu.



Obrázok 6 Stavové diagramy stavu posádky [vľavo], stavu zásahu [vpravo]



Obrázok 7 Vývojový diagram riadenia zásahu

Prvým krokom algoritmu spracovania nového zásahu je zapísanie výjazdu k pacientovi v momente, keď posádka opustí stanicu ZZS. Ten krok spôsobí, že sa zmení stav zásahu z „nový“ na „prebiehajúci“ a stav posádky z „pripravenej“ na „na ceste k pacientovi“. Po príchode posádky na miesto zásahu, nasleduje krok zapísania príjazdu k pacientovi a stav posádky sa zmení na „zasahuje na mieste“.

Po ukončení zásahu na mieste v prípade potreby nasleduje transport pacienta do nemocnice, ktorý zapíše operátor do systému a tým aj zmení stav posádky na „na ceste do nemocnice“. V momente príjazdu posádky do nemocnice, zapíše operátor príjazd do informačného systému a posádka prejde do stavu „odovzdáva pacienta“. Ak odovzdávanie pacienta bolo ukončené alebo transport pacienta do nemocnice nebol nutný, posádka sa vracia späť na stanicu a operátor zapíše skutočnosť do systému pomocou zmeny stavu posádky na „na ceste späť na stanicu“.

Po príjazde posádky späť na stanicu, operátor ukončí výjazd, čo znamená stav posádky sa zmení na „pripravená“ a stav zásahu na „ukončený“. Po doplnení všetkej potrebnej administratívy operátor uzavrie zásah a zásah nadobudne stav „uzavretý“.

3.3 Požiadavky na obslužný systém ZZS

Z analýzy problematiky ZZS a zásahu vyplynuli požiadavky, ktoré informačný systém pre obsluhu ZZS musí spĺňať. Požiadavky jednotlivých aktérov systému ZZS zrekapitulujem pomocou diagramu prípadov použitia na Obrázku 8.

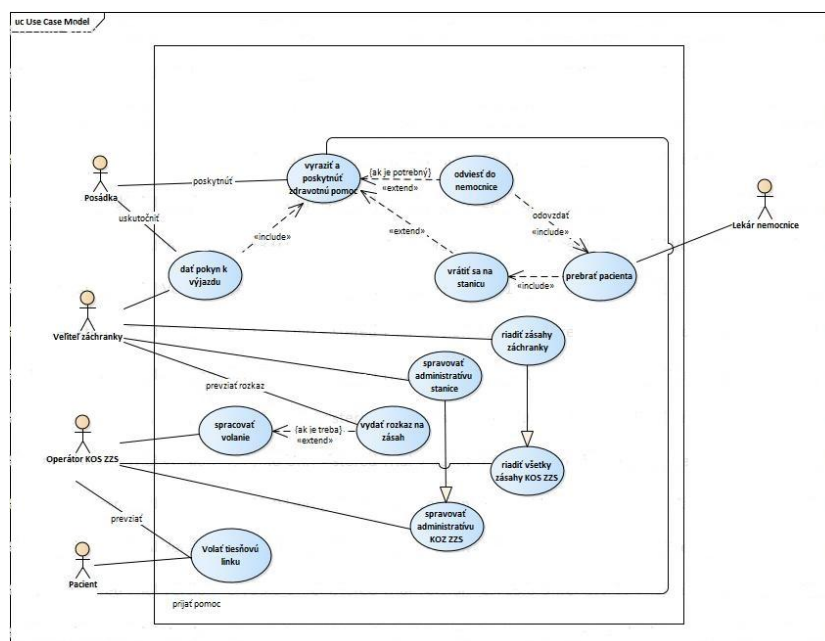
Hlavným aktérom systému ZZS je pacient, pre ktorého musia všetci ostatní aktéri zabezpečiť na základe tiesňového volania, čo najrýchlejšiu a najkvalitnejšiu zdravotnú starostlivosť. Keď operátor ZZS prevezme tiesňové volanie, tak na základe informácií volajúceho a využitím informačného systému, vyberie vhodnú posádku na vykonanie zásahu. Operátor sa následne podieľa na riadení práve prebiehajúcich zásahov, stará sa o dopĺňanie informácií a zabezpečuje ostatné činnosti bežnej administratívy (pridávanie nových staníc ZZS, nemocníc, a pod).

Veliteľ záchranky po prijatí rozkazu na zásah, vydá svojim podriadeným výjazdový formulár spolu so všetkými, dovtedy dostupnými informáciami a vyšle ich plniť rozkaz. Veliteľ ďalej plní funkcie plynúce z interných predpisov. Mojim návrhom pre vylepšenie fungovania systému ZZS, bolo jednotlivým veliteľom umožniť sa aktívnejšie zapájať

priamo do informačného systému ZZS. Docielil som to pridaním grafického prostredia určeného pre veliteľa, pomocou ktorého by veliteľ vedel spravovať zásah rovnakým algoritmom ako operátori KOS ZZS. Rovnako dôležité funkcie užívateľského prostredia veliteľa, by umožňovali uzatváranie zásahov potrebnou administratívnou dokumentáciou, riadiť zloženia jednotlivých posádok. Samozrejmosťou by bolo aj pridávanie nových sanitiek (posádok), zamestnancov a tiež možnosť spracovania dochádzky jednotlivých zamestnancov. Taký spôsob organizovania ZZS, by mohol byť výhodný hlavne v staniciach, kde sídli viacero posádok ZZS, z dôvodu užšej spolupráce veliteľa a posádky, čo by uľahčilo aj prácu KOS ZZS.

Posádka v užívateľskom prostredí nepotrebuje zastúpenie, lebo počas vykonávania zásahu musí byť neustále koncentrovaná na zásah a nemá čas sa zapodievať priamym dopĺňaním informácií do systému. Dopĺňanie informácií zásahu do informačného systému prebieha pomocou vysielacky s operátom a s veliteľom.

Posledným aktérom zásahu v prípade potreby je zamestnanec nemocnice, ktorý sa stará o príjem pacientov zo zásahov. Je to lekár na príjme, ktorý je upozornený na prichádzajúci zásah pomocou telefónu. Pre jeho potrebu som navrhol grafické prostredie, cez ktoré by bol upozornený na nový prichádzajúci zásah a mohol by o ňom získať potrebné informácie. Ďalej má možnosť zapísať do informačného systému ZZS príjazd posádky a po ukončení preberania odjazd posádky na domovskú stanicu.



Obrázok 8 Diagram prípadov použitia ZZS

3.4 Navrhovanie databázy

Pre ukladanie dát som zvolil databázu MS SQL server 2012 a pre vytvorenie schémy databázy (vid'. Obrázok 9) som použil nástroj Toad Data Modeler. Už prvý pohľad napovedal, že databázový model bude pomerne rozsiahly a zložitý, čo bolo potvrdené aj výsledkami predchádzajúcej analýzy. Databáza by mala byť tvorená tromi hlavnými navzájom prepojenými časťami, a to:

Prvá časť databázy tvorená entitami, ktoré by mali obsluhovať požiadavky týkajúce sa lokalizácie, výpočtov vzdialeností a pod. Druhou časťou sú entity, ktoré vlastne tvoria základné uzly siete ZZS, ako napríklad nemocnice, záchranky a pod. Tretia časť databázy tvorená entitami pre funkčnosť samotného systému, ktoré nie sú nevyhnutne potrebné, ale starajú sa o ostatné užitočné doplnkové funkcie. Poslednou a možno aj najdôležitejšou časťou databázy budú entity, potrebné pre obsluhovanie tiesňových volaní a následnú správu zásahov.

Keďže pri výbere vhodnej posádky je vždy nutné vedieť vzdialenosti medzi stanicou posádky a miestom zásahu, použil som online dostupnú maticu vzdialeností. Potrebné bolo navrhnúť entity pre vyriešenie tohto problému. Vyriešil som to pomocou entity OBEC, ktorá je sama sebou prepojená vzťahom M:N, čím sa automaticky vytvorila entita MATICA_VZDIALENOSTI. Primárny kľúč entity OBEC je atribút id_obce (typu Int), zvyšné atribúty sú nazov (typu nVarchar(30)) a kraj, pre ktorý som definoval obmedzenie na bežné, v praxi používané skratky napr. TN, BB, BA, atď. MATICA_VZDIALENOSTI preto obsahovala primárny cudzí kľúč, ktorý po premenovaní vyzerá nasledovne: „id_obce_z, id_obce_do“. Pre funkčnosť takto vytvorenej matice som do nej doplnil atribút, ktorý bude vyjadrovať vzdialenosti medzi obcami.

Vzhľadom k tomu, že pre správne fungovanie obslužného systému nestačilo len vedieť určiť miesto pomocou názvu mesta, prípadne obce, bolo potrebné vyriešiť problém týkajúci sa presnejšej lokácie. Ako riešenie tohto problému by som navrhol entitu ADRESA, ktorá by bola tvorená primárnym kľúčom id_adresy (Int identity), pochopiteľne s cudzím kľúčom id_obce a doplnená povinnými atribútmi ulica (nVarchar), gps (Geography) a nepovinným atribútom cislo (nVarchar). Týmto riešením sa stal problém zisťovania vzdialeností a adries vyriešený a možno pokračovať v modelovaní ostatných častí databázy.

Navrhovanie entít predstavujúcich jednotlivých aktérov ZZS neprinieslo žiadne zložité komplikácie. Entita ZDRAVOTNICKE_ZARIADENIE má primárny kľúč `id_zariadenia` (INT identity), cudzím kľúčom je len `id_adresy` a je doplnená o atribúty `tel_cislo1`, `tel_cislo2` a `nazov_zariadenia` (nVarchar). Poskytovatelia zdravotnej starostlivosti sú zastúpení entitou ZACHRANKA, ktorej primárny kľúč je atribút `id_zachranky` (INT identity), cudzí kľúč má len `id_adresy` a doplnkové atribúty sú `nazov_zachranky`, `tel_cislo1`, `tel_cislo2` (nVarchar). V bežnej praxi sa vyskytujú prípady, že v rámci jednej stanice poskytovateľ disponuje viacerými súčasne slúžiacimi posádkami. Preto bolo nutné zahrnúť takúto možnosť do systému. Riešil som to pridaním entity POSADKA a navrhnutím vzťahu medzi ZACHRANKOU a POSADKOU ako 1 : N.

Pre entitu POSADKA som navrhol primárny kľúč ako ŠPZ vozidla a preto vyzerá nasledovne `id_posadky_spz` (varchar(7)). Entita ďalej obsahuje cudzí kľúč `id_zachranky`, cudzie kľúče (`id_zamestnanca_vodic`, `id_zamestnanca_zachranar`, `id_zamestnanca_lekar`), pomocou ktorých sa dá identifikovať zloženie posádky (ak sú NULL, je tam voľné miesto), a vlastné atribúty `typ_posadky` a `stav_posadky`, z ktorých je jasné, akého typu a v akom stave sa posádka nachádza.

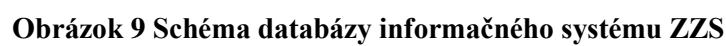
Prvou entitou tretej časti databázy je entita OS_UDAJE, tvorí ju primárny kľúč `rod_cislo` (char(11)), cudzí kľúč `id_adresy`, povinné atribúty `meno`, `priezvisko` (nVarchar) a nepovinný atribút `cislo_op`(char(8)). Pre osoby vystupujúce v systéme ako súčasť pracovnej štruktúry v ľubovoľnom uzle siete ZZS, som navrhol entitu ZAMESTNANEC_UCET. Entita bola navrhnutá pre potrebu riešenia problému prihlasovania sa do informačného systému. Každému novému zamestnancovi, musí byť pridelený takto vytvorený účet, pomocou ktorého sa prihlasuje do systému. Primárnym kľúčom je `id_zamestnanca` (Int identity), cudzím kľúčom je `rod_cislo`, ostatné povinné atribúty sú `login` a `password` (nVarchar), slúžiace na prihlasovanie do systému. Pre určenie pracovnej náplne zamestnanca napr. vodič, operátor a pod. slúži atribút `prac_pozicia`. Entita DOCHADZKA slúži na evidovanie príchodov a odchodov jednotlivých zamestnancov. Entita má primárny kľúč `id_dochadzky` (Int identity), cudzí kľúč je `id_zamestnanca` povinným atribútom je `datum_prichodu` (datetime) a nepovinný `datum_odchodu`, ktorý sa vyplňa pri odchode zamestnanca z práce.

V systéme som ďalej riešil vzťah (M:N) medzi zamestnancami a zamestnávateľmi (nemocnice, záchranky, posádky). Použil som entitu PRACUJUCI_ZAMESTNANCI, kde primárny cudzí kľúč je `id_zamestnanca` a jediný povinný atribút je `stav_zamestnanca`,

pomocou ktorého sa určuje, či je zamestnanec aktuálne v práci. Cudzie kľúče `id_zariadenia` a `id_zachranky` pomocou rôznych vzájomných kombinácií zobrazujú, u koho je osoba zamestnaná. V systéme môžu nastať tri situácie :

- ak `id_zariadenia` nie je NULL a `id_zachranky` je NULL = zamestnanec pracuje v nemocnici,
- ak `id_zariadenia` je NULL a `id_zachranky` nie je NULL = zamestnanec pracuje pre záchranku,
- ak `id_zariadenia` je NULL a `id_zachranky` je NULL = zamestnanec pracuje ako operátor KOZ ZZS.

Poslednú časť databázy tvoria entity využívané na vytváranie a následné spravovanie zásahov. Prvou entitou tejto časti je entita DIAGNOZY, ktorá slúži na spracovanie a určovanie diagnóz. Primárny kľúč `id_diagnozy` (nVarchar) zachytáva jednotlivé oficiálne používané kódy diagnóz. Atribút `typ` (nVarchar) slúži na klasifikáciu diagnóz a atribút `nazov` (text) pomenováva diagnózu. Entita PACIENT slúži na reprezentáciu pacienta potrebujúceho zdravotnú starostlivosť. Primárny kľúč entity PACIENT je `id_pacienta` (Int identity), cudzie kľúče sú `rod_cislo` a `id_diagnozy` (počiatočná diagnóza určená z tiesňového volania) a ostatné atribúty popisujúce vlastnosti entity sú `info_o_pacientovi` (text) a `tel_cislo` (varchar(10)). Ďalšou dôležitou, už vysvetlenou (odsek entity ZZS) entitou obsluhujúcou zásahy je, POSADKA. Vzťah medzi posádkou a pacientom som zadal nasledovne: „jedna posádka môže obslúžiť viacero zásahov a k jednému pacientovi môže byť zavolaných viacero posádok“ (M:N). Z toho dôvodu vznikne nová entita ZASAH, ktorá slúži na monitorovanie zásahov. Primárnym kľúčom entity je `id_zasahu` (Int identity), povinné cudzie kľúče sú `id_posadky_spz`, `id_pacienta`, `id_adresy_zasahu`, `id_diagnozy` a nepovinné `id_zariadenia` (NULL ak nebol urobený prevoz do nemocnice) a `id_diagnózy` (konečná diagnóza). Jediný povinný atribút je `stav`, ktorý vyjadruje stav zásahu (viď. Obrázok 6. Stavové diagramy [pravý]). Atribút `info_o_zasahu` (text) slúži na popisovanie zásahu, atribúty typu (Datetime) na zapisovanie časov príchodov a odchodov v jednotlivých krokoch obslužného algoritmu (viď. Obrázok 7 Vývojový diagram riadenia zásahu) a ostatné atribúty (Int) slúžia na evidenciu jednotlivých aktérov zásahu ako napr. `id_operator`.



4 Realizácia

4.1 Vstupné dáta

Prvotné informácie nevyhnutné na realizáciu práce, som získal od vedúcej bakalárskej práce, ktoré som doplnil dostupnými informáciami z oficiálnych stránok Záchrannej zdravotnej služby SR, Ministerstva zdravotníctva a VÚC Trenčín.

Hlavným zdrojom pre dáta potrebné na realizáciu informačného systému ZZS boli tri XLS súbory od vedúcej bakalárskej práce. Prvý súbor bol nemocniceSR.xls, ktorý obsahoval zoznam miest, v ktorých sa nachádzajú jednotlivé nemocnice SR. Druhý súbor TNstanice.xls obsahoval zoznam miest a obcí (aj s identifikátorom, ktorý bol v databáze použitý ako id_obce). Súbor obsahoval aj príslušný počet obyvateľov každej obce Trenčianskeho kraja a počet sanitiek sídliačich v určených mestách. V poslednom súbore boli uložené interné školské súradnice (nie GPS) miest a obcí SR, ale tie som nevyužil.

Druhým hlavným zdrojom bola (7) (školská) matica vzdialeností medzi jednotlivými mestami a obcami Trenčianskeho kraja nachádzajúca sa na web stránke KMMOA, FRI ŽU, ktorá bola rozložená do viacerých textových súborov. V prvom súbore boli identifikátory miest a obcí reprezentované v databáze atribútom id_obce_z a v druhom identifikátory id_obce_do. V poslednom potrebnom súbore boli vzdialenosti medzi jednotlivými identifikátormi. Preto som zjednotil tieto súbory do jedného súboru XLS do formátu, kde prvý stĺpec je id_obce_z, druhý stĺpec id_obce_do a tretí vzdialenosť medzi nimi. Keďže pre potrebu informačného systému mi stačila len čiastočná matica, reprezentujúca len vzdialenosti medzi miestami sídiel posádok a všetkými mestami a obcami Trenčianskeho kraja som sa rozhodol, že vyfiltrujem nepotrebné riadky takto vytvoreného súboru. Výhodou vyfiltrovaného súboru je, že prehľadávanie čiastočnej matice je rýchlejšie, ale v prípade zavedenia novej stanice v novej obci alebo meste treba maticu vzdialeností doplniť údajmi.

Doplňkovým zdrojom dát bola moja semestrálna práca zo Základov databázových systémov odkiaľ som čerpal fiktívne dáta o osobách (mená, rodné čísla a pod.) ďalším zdrojom boli internetové stránky samotných inštitúcií, kde som získal oficiálne dáta o adresách, presných názvoch a pod.

Pre importovanie dát do databázy som mohol využiť viacero postupov. Prvou možnosťou bolo vytvorenie súborov ako napr. XLS, CSV a pod., kde jednotlivé stĺpce reprezentujú príslušné atribúty tabuľky v databáze a priamo importovať dáta do príslušných tabuliek pomocou SQL Managment Studia. Takúto možnosť som zamietol z dôvodu osobných skúseností, ktoré som získal či už pri plnení pôvodnej Oracle 10g Databázy. S touto databázou som pôvodne začínal a vymenil som ju za MS SQL Server z dôvodu enormného zaťažovania procesoru. Zvoleným postupom nakoniec bolo vytvorenie obyčajných textových súborov naplnených príslušnými sql insert príkazmi pre jednotlivé tabuľky vytvorenými pomocou Excelu a jeho reťazcových funkcií. Výhodou tohto postupu je rýchlejšie hľadanie a opravovanie chýb pri importe dát do príslušných tabuliek databázy.

4.2 Implementácia

Po ukončení fázy navrhovania informačného systému vyberiem najvhodnejšiu databázovú technológiu (MS SQL server 2012), ktorú lokálne nainštalujem a správne nakonfigurujem pomocou Server Configuration Manager-a.

Z návrhu modelu databázy v TDM vygenerujem sql skript schémy databázy, ktorý spustím v SQL Server Management Studio z čoho sa vytvoria príslušne tabuľky, vzťahy, obmedzenia a pod. Tento nástroj budem používať na správu a modifikáciu databázy, ako aj na programovanie databázovej vrstvy aplikácie. Následne prepojím pomocou príslušného JDBC ovládača vytvorenú databázu s programovacím jazykom JAVA. Pre programovanie aplikačnej a prezentačnej vrstvy aplikácie využijem najnovšiu verziu JAVA SE s prekladačom Netbeans, kde naprogramujem funkcionálnu a dizajn samotného informačného systému. Pre vytvorenie šablón niektorých výstupov aplikácie, použijem Jaspersoft iReport Designer, ktoré bude informačný systém využívať pomocou knižnice JasperReports. Pre výpočet vzdialeností medzi dvoma uzlami použijem internetové rozhranie Google Distance Matrix a pre zistenie súradníc Google Geocodig API.

4.2.1 Grafika užívateľského prostredia

Pre tvorbu vzhľadu grafickej časti informačného systému som využil knižnicu Swing. Jednotlivé komponenty ako napr. okná, tlačidlá, textové polia atď. som vytiahol z palety nástrojov Netbeans-u. Vývojové prostredie vygenerovalo základný kód takto vytiahnutých komponentov, do ktorého som v prípade potreby doplnil pomocou grafického

prostredia Netbeans-u rozširujúci kód. Dobrým príkladom (viď. Obrázok 10) je Jtable, ktorej základni vlastnosti som rozšíril o farebné rozlišovanie riadkov tabuľky podľa hodnoty bunky (napr. pohľad lekára na príjme posádky idúce do nemocnice sú červené, posádky odovzdávajúce pacienta sú zelené).

```
JTabNemocnicaZasah = new javax.swing.JTable() {
    public Component prepareRenderer(TableCellRenderer renderer, int row, int column){
        Component c = super.prepareRenderer(renderer, row, column);
        //nastavenie farby riadku podľa hodnoty bunky v riadku
        //stav_posadky
        if (!isRowSelected(row)) {
            c.setBackground(getBackground());
            int modelRow = convertRowIndexToModel(row);
            String type = (String) getModel().getValueAt(modelRow, 6);
            if ("NaCesteDoNemocnice".equals(type))
                {c.setBackground(Color.red);}
        }
    }
}
```

Obrázok 10 Farebné rozlišovanie riadkov JTable podľa hodnoty bunky riadku

Ďalším príkladom (viď. Obrázok 11) sú inicializácie jednotlivých JComboBox–ov, ktoré naplňam dátami priamo z databázy na základe select-u (zoznam nemocníc využívaný pri zapisovaní prevozu pacienta do nemocnice).

JComboBox-y, ktoré sú vyplňané real-time podľa zapisovaného podreťazca ako napríklad zoznam miest a obcí. Fungujú (viď. Obrázok 12 a 13) na základe ASCII kódu získaného z tlačenia klávesy klávesnice, ak je platný (t.j. znaky A-Z, a-z, Backspace), nastaví sa model príslušného JComboBox-u (zoznam položiek) pomocou metódy getNazvyObce (podreťaz), ktorá vracia ComboBox model, naplnený pomocou zdieľanej procedúry databázy PROC_SELECT_MESTO. Ak zoznam položiek takto získaného zoznamu obsahuje položky a nebol použitý Backspace, doplní sa názov na najviac abecedne podobný, inak sa zaznamená len odobratie znaku. Vyhľadávanie je zobrazené na obrázku 14.

```
public void naplCBJNemocnice() {
    try {
        conn = DBConnection.DBConnection();
        st = conn.createStatement();
        rs = st.executeQuery("select nazov_zariadenia from zdravotnicke_zariadenie");
        while (rs.next()) {
            jCB_FOP_Nemocnice.addItem(rs.getString(1));
            jCBNemocnice1.addItem(rs.getString(1));
            jCBNemocnice2.addItem(rs.getString(1));
            jCB_FOP_Nemocnice1.addItem(rs.getString(1));
        }
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, e.getMessage());
    }
}
```

Obrázok 11Metóda slúžiaca na naplnenie JComboBox-u dátami z databázy


```

jCB_NZ_obce.getEditor().getEditorComponent().addKeyListener(new KeyAdapter() {
@Override
public void keyReleased(KeyEvent evt) {
    String pom = jCB_NZ_obce.getEditor().getItem().toString();
    if (evt.getKeyCode() >= 65 && evt.getKeyCode() <= 90
        || evt.getKeyLocation() > 96 && evt.getKeyCode() <= 105
        || evt.getKeyCode() == 8) {
        jCB_NZ_obce.setModel(getNazvyObce(pom));
        if (jCB_NZ_obce.getItemCount() > 0) {
            jCB_NZ_obce.showPopup();
            if (evt.getKeyCode() != 8) {
                ((JTextComponent) jCB_NZ_obce.getEditor().getEditorComponent())
                    .select(pom.length(), jCB_NZ_obce.getEditor().getItem().toString().length());
            } else {jCB_NZ_obce.getEditor().setItem(pom);}
        } else {jCB_NZ_obce.addItem(pom);}
    }
}});

```

Obrázok 12 Real-time napĺňanie ComboBox-u podľa podreťazca

```

public DefaultComboBoxModel getNazvyObce(String str) {
    DefaultComboBoxModel mod = new DefaultComboBoxModel();
    try {
        conn = DBConnection.DBConnection();
        stored_pro = conn.prepareCall("{call PROC_SELECT_MESTO (?)}");
        stored_pro.setString(1, str);
        rs = stored_pro.executeQuery();
        while (rs.next()) {
            mod.addElement(rs.getString(1));
        }
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, e.getMessage());
    }
    return mod;
}

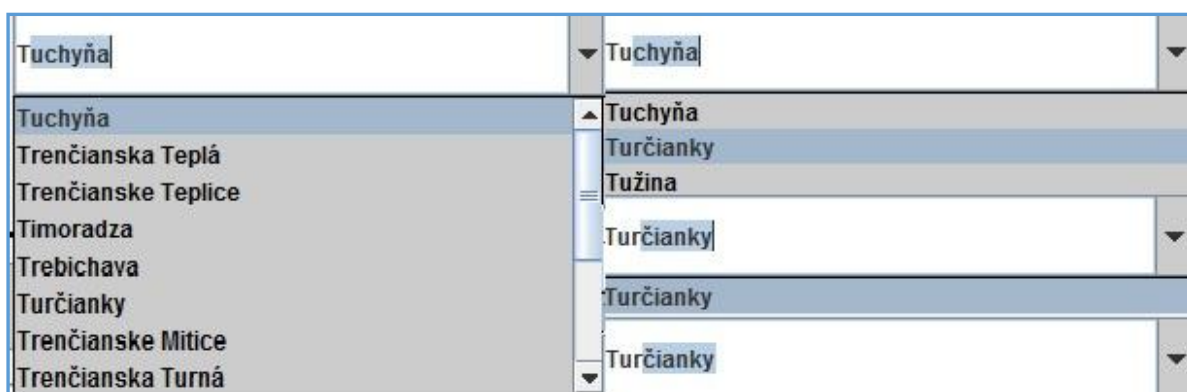
```

```

ALTER PROCEDURE [dbo].[PROC_SELECT_MESTO]
    @pomnazovObce nvarchar(30)
AS
BEGIN
    select nazov from obec
    where nazov like @pomnazovObce+'%'
END

```

Obrázok 13 Metóda, ktorá vracia model ComboBox-u na základe zdieľanej procedúry databázy [vpravo dole]



Obrázok 14 Real-time vyhľadávanie miest a obcí

4.2.2 Prepojenie s databázou

Komunikácia medzi databázou a programom funguje na základe JDBC ovládača, ktorý umožňuje programu pracovať a využívať databázu, nachádzajúca sa mimo programu. Pre potrebu vytvorenia spojenia medzi programom a databázou, som vytvoril triedu

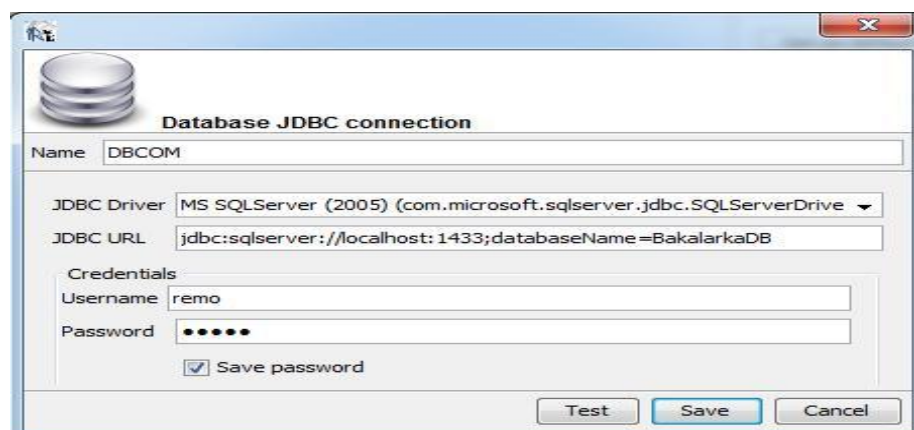
DBConnection, ktorá obsahuje metódu DBConnection(). Táto metóda na základne štandardných parametrov (názov databázy, login, heslo, port atď.) potrebných pre vytvorenie url adresy, potrebnej pre vytvorenie prepojenia s lokálne nainštalovanou databázou. Na obrázku 15 sa nachádza zdrojový kód tejto metódy, ktorý obsahuje aj ošetrovanie výnimiek, ktoré počas behu programu môžu vzniknúť. Túto metódu využívajú všetky metódy, ktoré spolupracujú s databázou.

```
public static Connection DBConnection1() {
    try {
        String driverName = "com.microsoft.sqlserver.jdbc.SQLServerDriver";
        Class.forName(driverName); String DBName = "BakalarkaDB";
        String username = "remo"; String password = "rendo";
        String port = "1433;"; String url = "jdbc:sqlserver://192.168.1.32:"
        + port + "databaseName=" + DBName + "user=" + username +
        "password=" + password;
        Connection con = DriverManager.getConnection(url);
        System.out.println("Succesfully Connected to the database !!! ");
        return con;
    } catch (ClassNotFoundException e) {
        JOptionPane.showMessageDialog(null, "Nenašiel sa databázový ovládač \n"
        + e.getMessage(), "Chyba", JOptionPane.ERROR_MESSAGE);
        return null;
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(null, "Nepodarilo sa pripojiť k databáze"
        + "\n" + e.getMessage(), "Chyba", JOptionPane.ERROR_MESSAGE);
        return null;
    }
}
```

Obrázok 15 Metóda DBConnection() na vytvorenie prepojenia s DB

4.2.3 Využitie JasperReports šablón

Prvým krokom potrebným pri vytvorení takejto šablóny je prepojenie dizajnéra šablón s databázou. Z databázy sa vyberú dáta, tie sa následne spracujú pri tvorbe šablóny. Prepojenie sa vytvára rovnakým postupom, aký je uvedený v triede DBConnection. Grafický nástroj na vytvorenie takého pripojenia v Jaspersoft iReport Designer-i vyzerá:



Obrázok 16 Prepojenie na databázu z iReport Designer-a

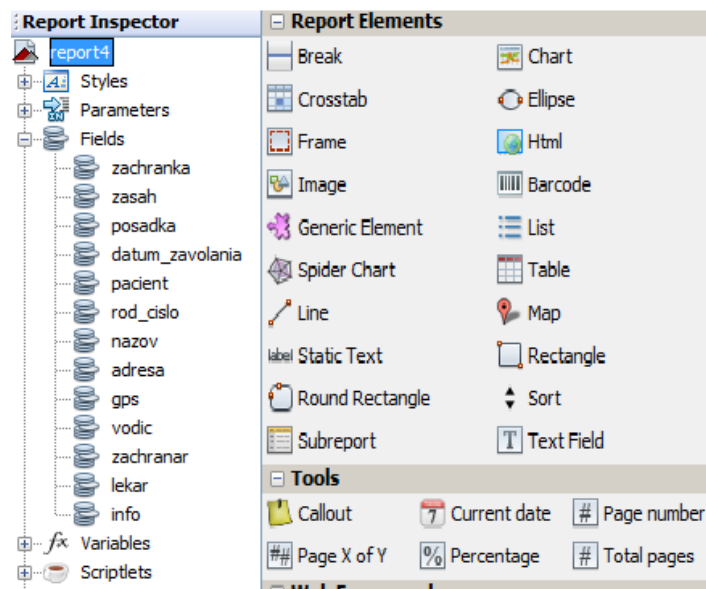
Druhým krokom je vytvorenie SQL select príkazu na získanie dát, ktoré budú spracované vo výslednej správe. Na nižšie uvedenom obrázku je uvedený takýto príkaz, ktorý vracia dáta potrebné pri tvorbe výjazdového formulára.

```
[1]
select
(zach.nazov_zachranky + ' - ' + obl.nazov) as zachranka ,
id_zasahu as zasah,
zas.id_posadky_spz as posadka,
(CONVERT(varchar(19),FORMAT(zas.zavolanie,'dd/MM/yyyy
HH:mm:ss'))) as datum_zavolania,
(ou.meno + ' ' + ou.priezvisko) as pacient,
ou.rod_cislo,
ob.nazov,
(adr.ulica+ ' ' + adr.cislo) as adresa,
('[' + substring(CONVERT(VARCHAR(20),CAST(adr.gps.Lat AS
decimal(18,14))),0,10) + ' , '
+ substring(CONVERT(VARCHAR(20),CAST(adr.gps.Long AS
decimal(18,14))),0,10) + ']' ) as gps,
(ouv.priezvisko+ ' ' + ouv.meno) as vodiac ,
(ouz.priezvisko+ ' ' + ouz.meno) as zachranar,
(ISNULL(oul.priezvisko,'') + ' ' + ISNULL(oul.meno,'')) as lekar,
pac.info_o_pacientovi as info
from zasah zas

[2]
join posadka pos on (zas.id_posadky_spz = pos.id_posadky_spz)
join zachranka zach on (pos.id_zachranky = zach.id_zachranky)
join adresa adr1 on (zach.id_adresy = adr1.id_adresy)
join obec obl on (adr1.id_obce = obl.id_obce)
join pacient pac on (zas.id_pacienta = pac.id_pacienta)
join os_udaje ou on (pac.rod_cislo = ou.rod_cislo)
join adresa adr on (zas.id_adresy_zasahu = adr.id_adresy)
join obec ob on (adr.id_obce = ob.id_obce)
full outer join zamestnanec_ucet zul
on (pos.id_zamestnanca_lekar = zul.id_zamestnanca)
full outer join os_udaje oul
on (zul.rod_cislo = oul.rod_cislo)
join zamestnanec_ucet pzv
on (pzv.id_zamestnanca = pos.id_zamestnanca_vodic)
join zamestnanec_ucet pzz
on (pzz.id_zamestnanca = pos.id_zamestnanca_zachranar)
join os_udaje ouv on (ouv.rod_cislo = pzv.rod_cislo)
join os_udaje ouz on (ouz.rod_cislo = pzz.rod_cislo)
where zas.id_zasahu=95
```

Obrázok 17 Sql select skript pre výjazdový formulár

Po spustení skriptu sa v ponuke Report Inspector objaví nové prvky (viď obrázok 18.), ako napríklad Fields (zachranka, posadka atď.), prezentujúce jednotlivé výstupy reportu. Pomocou už existujúcich motívov šablón takto získaných prvkov a pomocou palety Report Elements, sa len potiahnutím myšou dajú vytvárať rôznorodé výstupy.



Obrázok 18 Navrhovanie JasperReports šablóny

Výjazdový formulár		Číslo zásahu :	Dátum a čas volania :				
		\$V{zasah_1}	\$F{datum_zavolania}				
\$F{zachranka}							
Zloženie zasahujúcej posádky :		Pacient :	<table border="1"> <tr> <td>\$F{pacient}</td> <td>Rodné číslo</td> </tr> <tr> <td></td> <td>\$F{rod_cislo}</td> </tr> </table>	\$F{pacient}	Rodné číslo		\$F{rod_cislo}
\$F{pacient}	Rodné číslo						
	\$F{rod_cislo}						
Vodič zásahu :	\$F{vodic}	\$F{nazov}					
Záchranár zásahu :	\$F{zachranar}	\$F{adresa}					
Lekár zásahu :	\$F{lekar}	\$F{gps}					
Info o pacientovi je : "+\$F{info}							

Obrázok 19 Šablóna výjazdového formulára

Vytvorenú šablónu zostavy vo formáte jrxml stačí už len uložiť a je pripravená na ďalšie použitie. Aby bolo možné využívať JasperReports na tlačenie alebo ukladanie zostáv do rôznych typov dokumentov v Jave, je potrebné stiahnuť a následne nainštalovať samotnú knižnicu, voľne dostupnú na internete(8).

Prvou fázou použitia vytvorenej šablóny zostavy je načítanie. Následne sa pripraví skript pre výber dát z databázy (viď. Obrázok 20) a následné samotné skompilovanie šablóny zostavy, pomocou JasperCompileManager-a (zostava.jasper) do tlačovej zostavy. Naplnením výsledných dát z databázy pomocou JasperFillManager získa tlačová zostava finálnu podobu a tak je pripravená na zobrazenie, tlačenie alebo ukladanie podľa potreby užívateľa.

```
JasperDesign jd = JRXmlLoader.load("SpravaOZasahu.jrxml");
JRDesignQuery nQ = new JRDesignQuery();
nQ.setText(query);
jd.setQuery(nQ);
JasperReport jr = JasperCompileManager.compileReport(jd);
JasperPrint jp = JasperFillManager.fillReport(jr, null, conn);
JasperViewer.viewReport(jp, false);
```

Obrázok 20 Práca so zostavou

4.2.4 Použitie Google Maps Rozhraní

Rozhrania Geocoding API a Distance Matrix API, ktoré som použil, fungujú na rovnakom princípe. Na odoslaní požiadavky pomocou URL adresy, na ktorú im jednotlivé rozhrania odpovedajú, buď vo formáte XML alebo JSON. Pre prácu

s rozhraniami, je najprv potrebné požiadať o api key podľa (9) postupu uvedeného vo voľne dostupnej programátorskej (10) dokumentácii umiestnenej na internete. Bezplatné denné limity na používanie rozhraní sú až 2500 požiadaviek. V jednej požiadavke môže byť max 25 východísk alebo 25 cieľov. Pre potrebu informačného systému v rámci jedného kraja, je plne postačujúce. V prípade potreby sa dajú požiadavky dokúpiť až do limitu 100 000 požiadaviek za deň, kde 1000 stojí 0,5 USD za deň.

Komunikácia medzi internetovými Google Maps rozhraniami a samotným programom prebieha na základe dotazu, v presne definovanom formáte, pomocou takto vytvorenej URL adresy. URL požiadavka (viď. Obrázok 21) pre zisťovanie jednotlivých časov a vzdialeností medzi stanicami ZZS a miestom zásahu musí obsahovať nasledovné parametre:

- json? – parameter, ktorý zabezpečí že výstup bude v JSON formáte,
- stanice – je zoznam adries všetkých staníc ZZS, reprezentovaný reťazcom, kde medzery v názvoch adries sú nahradené znamienkom „+“ a jednotlivé adresy sú oddelené znamienkom „|“ v tvare Ilava+Nemocničná+742|Trenčín+Jána+Derku+18,
- ciel – je adresa zásahu v rovnakom tvare ako pri staniach,
- mode=driving – znamená, že výsledná trasa je určená pre jazdu autom,
- language=sk-SK – zabezpečí použitie slovenských názvov miest vo výstupnom formáte,
- key – google api key, slúži na identifikovanie žiadateľa požiadavky.

```
String urls = "https://maps.googleapis.com/maps/api/distancematrix/"
+ "json?origins="
+ stanice
+ "&destinations="
+ ciel
+ "&mode=driving"
+ "&language=sk-SK"
+ "&key=AIzaSyChgYn2IO8qL1iZXuyC4Ly7wn5DKmmCIVY";
```

```
String stanice = "";
for (int i = 0; i < poleStanic.length; i++) {
    stanice += poleStanic[i] + "|";
}
stanice = (String) stanice.substring(0, stanice.length() - 1);
```

Obrázok 21 Požiadavka na Google Distance Matrix

Pre ilustráciu práce s rozhraniami som sa rozhodol popísať metódu, pomocou ktorej program zisťuje vzdialenosti a časy medzi jednotlivými stanicami ZZS a miestami zásahov. Základom metódy je Google Maps Distance Matrix API, čo je internetová služba

poskytujúca cestnú vzdialenosť a potrebný čas pre absolvovanie odporúčanej trasy medzi miestami začiatkov a cieľov. Vstupnými parametrami metódy je reťazcové pole adries jednotlivých staníc posádok ZZS získaných z databázy, a cieľ (adresa miesta zásahu). Vstupné parametre (viď. Obrázok 21 – v obdĺžniku) spracujem do potrebného formátu pre URL požiadavku, vytvorím spojenie s rozhraním a následne (viď. Obrázok 22) pomocou načítania po riadkoch načítam výstupný JSON výstup do reťazca result. Poslednou časťou metódy je (viď. obrázok 23) rozparsovanie (roztriedenie) reťazca result pomocou metód JSONParsera (voľne dostupná knižnica) a následné uloženie dát do nového poľa pole[pocetStanic][4]. Výstupným parametrom tejto metódy je :

[adresa stanice 1.posádky][adresa miesta zásahu][vzdialenosť medzi adr.][čas],
 [adresa stanice 2.posádky][adresa miesta zásahu][vzdialenosť medzi adr.][čas],
 [adresa stanice n.posádky][adresa miesta zásahu][vzdialenosť medzi adr.][čas].

```
URL url = new URL(urls);
URLConnection urlConnection = (URLConnection) url.openConnection();
String formattedAddress = "";
try {
    InputStream in = url.openStream();
    BufferedReader reader = new BufferedReader(new InputStreamReader(in));
    String result, line = reader.readLine();
    result = line;
    while ((line = reader.readLine()) != null) {
        result += line;
    }
}
JSONParser parser = new JSONParser();
JSONObject jb = (JSONObject) parser.parse(result);
JSONArray rows = (JSONArray) jb.get("rows");
JSONArray origin_addresses = (JSONArray) jb.get("origin_addresses");
JSONArray destination_addresses = (JSONArray) jb.get("destination_addresses");
```

Obrázok 22 Načítanie JSON výstupu z Google API

```
int k = rows.size();
int i = 0;
String[][] pole = new String[k][4];
while (i < k) {
    JSONObject row_i = (JSONObject) rows.get(i);
    JSONObject json_row_i = (JSONObject) parser.parse(row_i.toString());
    JSONArray elements = (JSONArray) json_row_i.get("elements");
    JSONObject element0 = (JSONObject) elements.get(0);
    JSONObject json_element0 = (JSONObject) parser.parse(element0.toString());
    JSONObject duration = (JSONObject) json_element0.get("duration");
    JSONObject distance = (JSONObject) json_element0.get("distance");
    pole[i][0] = destination_addresses.get(0).toString();
    pole[i][1] = origin_addresses.get(i).toString();
    pole[i][2] = duration.get("text").toString();
    pole[i][3] = distance.get("text").toString();
    i++;
}
```

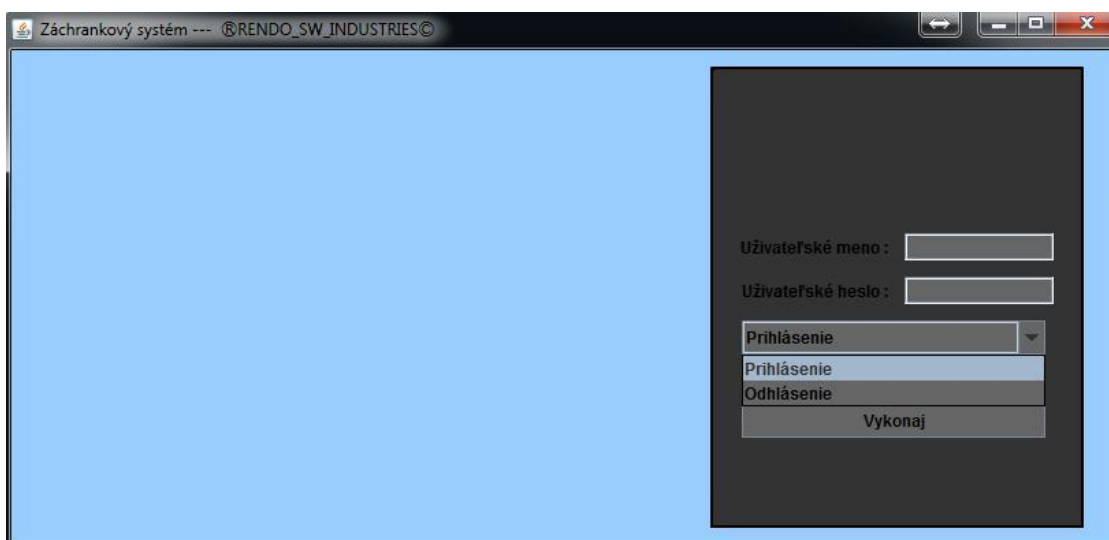
Obrázok 23 Rozparsovanie získaného JSON výstupu

4.3 Grafická užívateľská príručka

Hlavným zámerom pri tvorbe užívateľského prostredia informačného systému ZZS bolo vytvoriť čo najjednoduchšie, najintuitívnejšie a najrýchlejšie ovládanie aplikácie.

Prvým nevyhnutným krokom pre prácu s informačným systémom, je prihlásenie sa do systému pod užívateľským menom (login) a heslom, ktoré každý zamestnanec dostane od svojho nadriadeného pri registrácii do informačného systému. Prihlasovaco-odhlasovací formulár (viď. Obrázok nižšie) slúži aj ako nástroj obsluhujúci a spravujúci dochádzku jednotlivých zamestnancov. Pri prihlasovaní zamestnanca môžu nastať situácie:

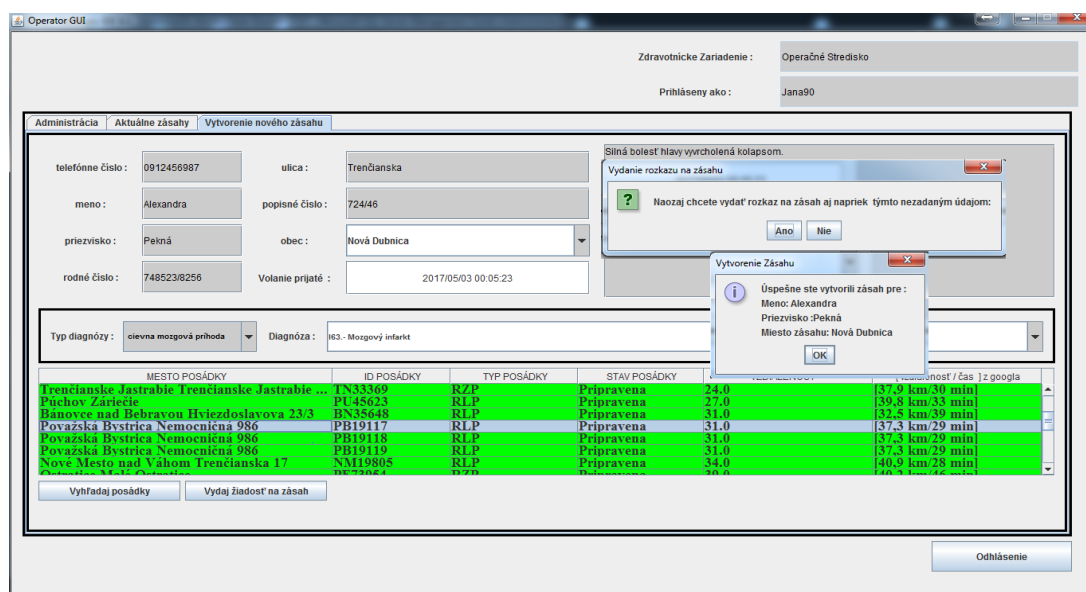
- ak zamestnanec vyplní nesprávne údaje, aplikácia ho upozorní na chybu a zamestnanec sa môže opäť skúsiť prihlásiť do systému,
- ak zamestnanec vyplní správne údaje a je členom ľubovoľnej posádky (vodič, lekár, záchranár) záchranky môžu nastať dve situácie :
 - ak záchranka ma práve jednu posádku, zamestnanec nahradí pôvodného zamestnanca (ak bol a zapíše sa odchod),
 - ak záchranka ma viacero posádok, systém umožní výber posádky, do ktorej sa zamestnanec môže prihlásiť.
- ak zamestnanec nie je členom posádky (operátor, lekár v nemocnici, veliteľ záchranky) tak po úspešnom prihlásení sa zamestnancovi zobrazí príslušný užívateľský pohľad.



Obrázok 24 Prihlasovaco-odhlasovací formulár ZZS

Informačný systém disponuje tromi rôznymi užívateľskými pohľadmi. Hlavný užívateľský pohľad je určený pre operátora KOS ZZS. Okno operátora je zložené z troch záložiek a to :

- administrácia – slúži na vytváranie nových zamestnancov daného operačného strediska, vytváranie nových staníc záchraniek, zamestnancov a správu administrácie,
- vytvorenie nového zásahu – vypíšu sa tu základné údaje o pacientovi (minimálne telefón, obec, ulicu) a následne pomocou tlačidla „Vyhľadaj posádky“ sa v tabuľke zobrazia voľné posádky s príslušnými informáciami o type, čase a vzdialenosti k miestu zásahu. Po označení vybranej posádky, operátor pomocou tlačidla „Vydaj žiadosť na zásah“ dokončí vytvorenie zásahu,
- aktuálne zásahy – slúži na spravovanie práve prebiehajúcich zásahov podľa algoritmu riadenia zásahu, ďalej sa tu tiež dajú dopĺňať k zásahom a pacientom nové informácie zo zásahov.



Obrázok 25 Vytvorenie zásahu operátorom KOS ZZS

Administrácia	Aktuálne zásahy	Vytvorenie nového zásahu								
ID ZÁSahu	STANICA POSÁDKY	ID POSÁDKY	STAV POSÁDKY	INFO O ZÁSahu	MIESTO ZÁSahu	ID PACIENTA	RODNÉ ČÍSLO	INFO O PACIENT...	ID ZARIADENIA	STAV
3178	Považská Bystrica	PB19117	NaCesteNa...		Nová Dubni...	3219	748523/8256	Silná bolesť...	0	prebiehajúci
3179	Trenčín	TN71126	NaCesteDo...		Nová Dubni...	3220	123699/8445	Bolesť v hr...	8	prebiehajúci
3180	Považská Bystrica	PB19118	ZasahujeNa...		Belusa	3221	124596/7956	Pad do stre...	0	prebiehajúci
3181	Považská Bystrica	PB19119	NaCesteKP...		Považská B...	3222	358723/1456	neznáme	0	prebiehajúci
3182	Handlová	PD99954	NaCesteKP...		Trenčianske...	3223	NE 200/22/44	Podozrenie ...	0	nový

Zapiš výjazd k pacientovi

Zapiš prevoz do nemocnice

NEMOCNICA S POLIKLINIKOU POVAŽSKÁ BYSTRICA

Zapiš príjazd k pacientovi

Zapiš príjazd do nemocnice

Zapiš príjazd na stanicu

Vytlač správu zo zásahu

Doplň údaje k zásahu

Zapiš návrat na stanicu

Uzavri zásah

Zobraz históriu pacienta

[748523/8256, Alexandra, Pekná]

refresh

MimoPrevádzky / Priprave...

Pôvodná diagnóza :
I63 - Mozgový infarkt

Pôvodné info o pacientovi :
Silná bolesť hlavy vyvrcholená kolapsom.

Bánovce nad...	BN35648	RLP	Pripravená
Dubnica nad...	IL59623	RLP	Pripravená
Ilava	IL78945	RZP	Pripravená
Myjava	MY18963	RLP	Pripravená
Brezová pod...	MY88225	RZP	Pripravená
Nové Mesto ...	NM19805	RLP	Pripravená
Stará Turá	NM23569	RZP	Pripravená
Považská By...	PB19117	RLP	NaCesteNaS...
Považská By...	PB19118	RLP	ZasahujeNa...
Považská By...	PB19119	RLP	NaCesteKPa...
Nitrianske R...	PD19546	RZP	Pripravená
Prievidza	PD33114	RZP	Pripravená
Bojnice	PD88223	RLP	Pripravená
Handlová	PD99954	RLP	NaCesteKPa...
Nitrianske P...	PD99977	RZP	Pripravená
Partizánske	PE65327	RLP	Pripravená
Ostratice	PE73954	RZP	Pripravená

Obrázok 26 Správa zásahov operátora KOS ZZS

Oprav Zsah

telefónne číslo : 0912456987
 Ulica: Trenčianska
 ID Zásahu: 3178

Meno: Alexandra
 Popisné číslo: 724/46

Priezvisko: Pekná
 Obec: Nová Dubnica

Rodné číslo: 748523/8256

Pravdepodobná Diagnóza: I63 - Mozgový infarkt

Info o pacientovi :

Mozgové krvácanie vzniká najčastejšie po prasknutí cievy typicky v oblasti bazálnych ganglií u hypertonikov, zriedkavejšie cievnej rozšíreniny (aneurizmy) alebo malformácie. Častejšie vzniká po námahe, rozčúlení a zvýšení krvného tlaku, môže sa vyskytnúť aj porucha vedomia, bolesti hlavy, napínanie na zvracanie alebo zvracanie.

Mozgová ischémia (nedokrvenie) vzniká najčastejšie pri zníženom prietoku krvi v ateroskleroticky zužujúcich mozgových cievach, môže byť však aj

Info k zásahu :

Celý systém je veľmi jednoduchý a v priebehu roka 2018 ho budú mať integrované všetky nové autá v Európskej únii povinne. Takýto systém má byť schopný zachrániť ročne tisíce životov.

Aktuálne si ho však môžete doinštalovať do auta len dodatočne. Jedno z takých riešení ponúka aj poisťovňa Generali, ktorej produkt som mal v aute. Je to malá biela krabička na čelnom sídle.

Konečná Diagnóza : NEZADANA

Prevoz: NEMOCNICA S POLIKLINIKOU POVAŽSKÁ BYSTRICA

Refresh
 Uložiť

Obrázok 27 Oprava/doplňanie údajov k zásahu

Druhý užívateľský pohľad je zameraný pre potreby veliteľa záchrannej stanice. Okno veliteľa sa tiež skladá z dvoch hlavných záložiek, a to :

- aktuálne zásahy záchranky – hlavná tabuľka zásahov, ktorá sa aktualizuje každých 60 sekúnd. Služi na zobrazenie a následnú správu zásahov pomocou algoritmu riadenia zásahu. Podobne ako pri pohľade operátor je tiež možné dopĺňať jednotlivé informácie k zásahom. Doplnil som aj možnosť vytlačenia výjazdového a uzáverového formulára zásahu.
- administrácia – služi na spravovanie zloženia posádok, pridávanie nových sanitiek, posádok a umožňuje tiež generovať dochádzku jednotlivým zamestnancom.

Zdravotnícke Zariadenie : Falck Záchraná a.s., Košice - Považská Bystrica

Prihlásený ako : Monika79

ID ZÁSAHU	ŠPZ POSÁDKY	STAV POSÁDKY	NEMOCNICA	PACIENT	KONTAKT	ADRESA	INFO	STAV
3178	PB19117	NaCesteNaSt...	NEMOCNICA S...	Alexandra Pekná	0912456987	Nová Dubnica , Tr...	Silná bolesť hlavy...	prebie...
3180	PB19118	Odovzdavan...	NEMOCNICA S...	František Karika	1425635871	Belusa , Farská , 1	Pád do strechy 3...	prebie...
3183	PB19119	NaCesteKPa...	žiadny	Monika Caputová	P125478335	Púchov , Kinekus ,	odpadla pri naku...	prebie...

Vytlač výjazdový formulár Zapiš prevoz do nemocnice Nemocnica Bánovce - 3. súkromná nemocnica, s.r.o.

Zapiš výjazd k pacientovi Zapiš príjazd do nemocnice Uzavri zásah [Datum tiesnového volania: 03/05/2017 00:05:23]
[748523/8256, Alexandra, Pekná]

Zapiš príjazd k pacientovi Zapiš návrat na stanicu refresh Pôvodná diagnóza : NEZADANÁ NEZADANÁ

Doplň údaje k zásahu Zapiš príjazd na stanicu Zobraz históriu pacienta Pôvodné info o pacientovi : Silná bolesť hlavy vyvrcholená kolapsom.

Odhlásenie

Obrázok 28 Okno veliteľ'a záchranky [nový zásah - červený]

Zdravotnícke Zariadenie : Falck Záchraná a.s., Košice - Považská Bystrica

Prihlásený ako : Monika79

ID ZÁSAHU	ŠPZ POSÁDKY	STAV POSÁDKY	NEMOCNICA	PACIENT	KONTAKT	ADRESA	INFO	STAV
3178	PB19117	NaCesteNaSt...	NEMOCNICA S...	Alexandra Pekná	0912456987	Nová Dubnica , Tr...	Silná bolesť hlavy...	prebie...
3180	PB19118	Odovzdavan...	NEMOCNICA S...	František Karika	1425635871	Belusa , Farská , 1	Pád do strechy 3...	prebie...
3183	PB19119	NaCesteKPa...	žiadny	Monika Caputová	P125478335	Púchov , Kinekus ,	odpadla pri naku...	prebie...

Vytlač výjazdový formulár Zapiš prevoz do nemocnice Nemocnica Bánovce - 3. súkromná nemocnica, s.r.o.

Zapiš výjazd k pacientovi Zapiš príjazd do nemocnice Uzavri zásah [Datum tiesnového volania: 03/05/2017 00:05:23]
[748523/8256, Alexandra, Pekná]

Zapiš príjazd k pacientovi Zapiš návrat na stanicu refresh Pôvodná diagnóza : NEZADANÁ NEZADANÁ

Doplň údaje k zásahu Zapiš príjazd na stanicu Zobraz históriu pacienta Pôvodné info o pacientovi : Silná bolesť hlavy vyvrcholená kolapsom.

Odhlásenie

Obrázok 29 Okno veliteľ'a záchranky [spracovanie nového rozkazu]

Zdravotnícke Zariadenie : Falck Záchraná a.s., Košice - Považská Bystrica

Prihlásený ako : Monika79

ID ZÁSAHU	ŠPZ POSÁDKY	STAV POSÁDKY	NEMOCNICA	PACIENT	KONTAKT	ADRESA	INFO	STAV
3178	PB19117	NaCesteNaSt...	NEMOCNICA S...	Alexandra Pekná	0912456987	Nová Dubnica , Tr...	Silná bolesť hlavy...	prebie...
3180	PB19118	Odovzdavan...	NEMOCNICA S...	František Karika	1425635871	Belusa , Farská , 1	Pád do strechy 3...	prebie...
3183	PB19119	ZasahujeNa...	žiadny	Monika Caputová	P125478335	Púchov , Kinekus ,	odpadla pri naku...	prebie...

Vytlač výjazdový formulár Zapiš prevoz do nemocnice Nemocnica Bánovce - 3. súkromná nemocnica, s.r.o.

Zapiš výjazd k pacientovi Zapiš príjazd do nemocnice Uzavri zásah [Datum tiesnového volania: 03/05/2017 00:05:23]
[748523/8256, Alexandra, Pekná]

Zapiš príjazd k pacientovi Zapiš návrat na stanicu refresh Pôvodná diagnóza : NEZADANÁ NEZADANÁ

Doplň údaje k zásahu Zapiš príjazd na stanicu Zobraz históriu pacienta Pôvodné info o pacientovi : Silná bolesť hlavy vyvrcholená kolapsom.

Odhlásenie

Obrázok 30 Okno veliteľ'a záchranky [nastavenie prevozu]

Dochádzka

Falck Záchraná a.s., Košice-Považská Bystrica
58 Monika Markova

príchod	odchod	nadať min
02/05/2017 12:47	02/05/2017 12:48	2
02/05/2017 12:51	02/05/2017 01:09	18
02/05/2017 01:09	02/05/2017 01:09	50
02/05/2017 02:07	02/05/2017 02:09	2

Strana 1 z 1

Uprav Zamestnanca: Monika Markova

Vytvor novú sanitku

Vygeneruj dochádzku

OD: 01.04.2017 DO: 03.05.2017

Odhlásenie

Obrázok 31 Okno veliteľ záchrany Administrácia [generovanie dochádzky]

Posledný užívateľský pohľad je určený pre lekára na príjme v nemocnici. Tento pohľad je zo všetkých najjednoduchší, v tabuľke sú zobrazené len zásahy, ktoré práve smerujú do daného zariadenia. Lekár môže z neho priamo zistiť informácie o konkrétnom zásahu a do systému doplniť informácie o príjazde a odjazde posádky.

Zdravotnícke Zariadenie: NEMOCNICA S POLIKLINIKOU POVAŽSKÁ BYSTRICA

Prihlásený ako: Marta72

Administrácia Aktuálne zásahy zariadenia

ID ZÁSAHU	ID POSÁDKY ŠPZ	ID PACIENTA	RODNÉ ČÍSLO	INFO O ZÁSAHU	INFO O PACIENTOVI	STAV POSÁDKY
3179	TN71126	3220	123699/8445		Bolesti v hrudník...	NaCesteDoNemoc...
3180	PB19118	3221	124596/7956		Pad do strechy 3...	OdovzdavaniePac...
3182	PD99954	3223	NEZ00:22:44		Podozrenie na žln...	NaCesteDoNemoc...

Zapiš príjazd do nemocni... Zobraz info k zásahu

Zapiš odchod z nemocnice Zobraz históriu pacienta

refresh Tlač

Pôvodná diagnóza :
S22.- Zlomenina rebra (rebrí), hrudnej kosti a hrudníkovej chrbtice

Pôvodné info o pacientovi :
Bolesti v hrudníku, ťažké dýchanie spôsobené pádom na motorke

Konečná diagnóza diagnóza :

Info o zasahu :
null

Odhlásenie

Obrázok 32 Okno lekár na príjme v nemocnici

Zdravotnícke Zariadenie : NEMOCNICA S POLIKLINIKOU POVAŽSKÁ BYSTRICA

Prihlásený ako : Simona72

ID ZÁSAHU	ID POSÁDKY ŠPZ	ID PACIENTA	RODNÉ ČÍSLO	INFO O ZÁSAHU	INFO O PACIENTOVI	STAV POSÁDKY
3190	PB19118	3231	NEZ16:05:21	fdffdfsd	fdffdfsd	NaCesteDoN...

Zapiš príjazd do n... Zobraz info k zásahu

Zapiš odchod z ne... Zobraz históriu pa...

refresh Tlač

[Dátum tiesňového volania: 04/05/2017 16:05:21][Id zásahu :3190]
[Filip,Stastny,NEZ16:05:21]

Pôvodná diagnóza :
NEZADANÁ NEZADANÁ

Pôvodné info o pacientovi :
fdffdfsd

Konečná diagnóza diagnóza :

Odhlásenie

Obrázok 33 Okno lekár otvorené na Raspberry

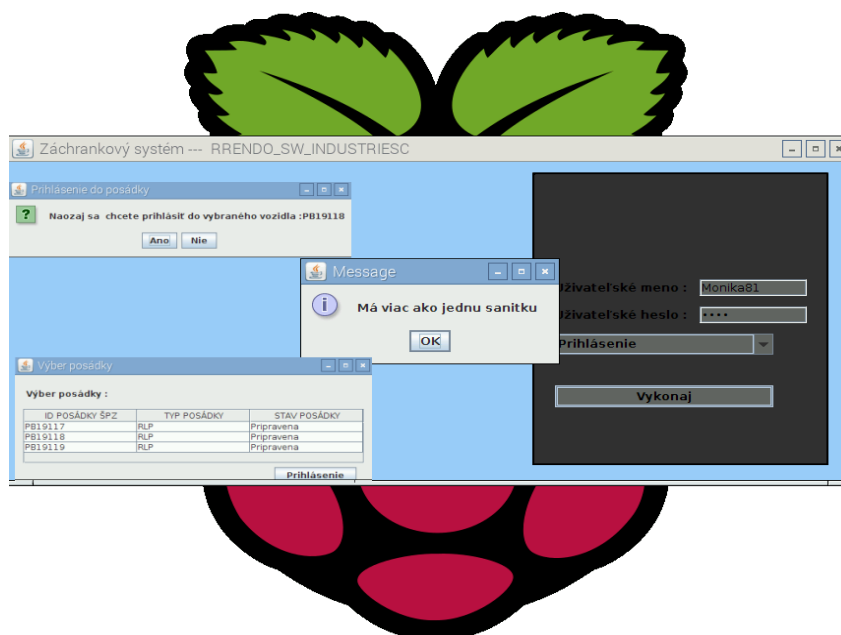
Zamestnanec po ukončení služby sa zo systému musí odhlásiť. Zamestnanci, ktorí majú vytvorený vlastný užívateľský pohľad sa odhlasujú pomocou tlačidla „Odhlásenie“ umiestneného v pravom dolnom rohu okna. Následne musia tento krok potvrdiť v prihlasovaco-odhlasovacom formulári ako členovia jednotlivých posádok a tak sa zapíše dátum a čas odchodu zamestnanca do systému.

Video návod pre jednotlivých užívateľov na používanie informačného systému ZZS je k dispozícii na priloženom CD, kde sa nachádza samotný program, dokumentácia a pod.

5 Výsledky práce a diskusia

Po ukončení programovania informačného systému som začal s jeho testovaním. Pre komfortné testovanie informačného systému, bolo potrebné testovať ho na viacerých počítačoch súčasne. Drobné chyby, ktoré som zistil pri testovaní, boli postupne odstraňované, pokiaľ som samotný informačný systém kompletne nevyhladil.

Pri simulácii činnosti jednotlivých aktérov ZZS, som musel používať viacero rôznych typov počítačov, súčasne umiestnených v lokálnej sieti. Na serverový počítač, kde bola umiestnená databáza, pristupujú jednotlivé aplikácie pomocou ip adresy serverového počítača a príslušného portu databázy. Funkčnosť informačného systému som preveril pri prevádzke troch súčasne zapojených užívateľov. Prvý užívateľ bol operátor KOS ZZS, druhý veliteľ záchranky a tretí aktér zásahu, bol lekár na príjme v nemocnici. Pri skúšobnej prevádzke sa problémy neprejavili a obnovovanie tabuľky aktuálnych zásahov u jednotlivých aktérov fungovalo bezproblémovo. Zo zvedavosti som do takejto skúšobnej prevádzky pridal aj zariadenie s iným operačným systémom. Išlo o mikropočítač Raspberry PI 3, na ktorom aplikácia informačného systému bežala v obmedzenej rýchlosti. To by v reálnej praxi nebolo možné využiť v operačnom stredisku ZZS. Pri šetrení na výdavkoch pri obstarávaní hardvéru pre potrebu lekára na príjme, by aplikácia nainštalovaná na Raspberry spĺňala požiadavky. Vhodnejšie je využitie na pohodlnú evidenciu dochádzky jednotlivých členov posádok.



Obrázok 34 Prihlasovanie sa do posádky na Raspberry

Vzhľadom k rozsiahlosti informačného systému ZZS, bude v databáze uložené veľké množstvo dôležitých dát, ktoré sa môžu využiť v budúcich analýzach pre zefektívnenie činnosti jednotlivých zložiek ZZS. Z toho dôvodu som vytvoril nástroj na získavanie dát z databázy ZZS, s ktorým pomocou navrhnutých filtrov a ich vzájomnou kombináciou dokáže užívateľ (operátor ZZS) získať dáta využiteľné pre rôzne štatistiky.

Pomocou filtrov je možné jednotlivým aktérom ZZS (napr. záchrankám, nemocniciam) vygenerovať prehľad o ich zásahoch, ktorý je možné pomocou kalendára časovo ohraničiť. Ďalej je možné bližšie špecifikovať, napr. typ diagnózy, zhody/nezhody predbežných a konečných diagnóz. Pri potrebe riešenia sporných zásahov, je z databázy možné získať veľa užitočných informácií, ako napr. časový priebeh zásahu, zistenie jednotlivých aktérov zásahu od členov posádky, cez operátora až po osobu, ktorá zásah uzavrela. Takto vybrané dáta sa zobrazia v tabuľke a je možné ich ďalej exportovať do rôznych typov súborov.

Obrázok 35 Filtre informačného systému

Pre potrebu informácií o histórii pacientov, som vytvoril jednotlivým aktérom v záložke aktuálnych zásahov, možnosť zistenia histórie pacienta, s prehľadom jeho predošlých zásahových udalostí. V budúcnosti, v prípade ďalšej požiadavky, by sa dal informačný systém a databáza doplniť o nové možnosti, napr. ukladanie nahrávky tiesňového volania, audio alebo video záznam o priebehu zásahu. V aplikácii sú ošetrené výnimky, ktoré sa počas bežnej prevádzky môžu vyskytnúť, čím je docielená rýchlosť a jednoduchosť práce s aplikáciou.

Záver

Pri vypracovávaní bakalárskej práce som využil veľa poznatkov získaných počas bakalárskeho štúdia na Fakulte riadenia a informatiky Žilinskej univerzity. Pre naplnenie cieľov bakalárskej práce som si musel doštudovať aj iné informatické nástroje a technológie, ktoré v praxi môžem využiť.

Hlavným cieľom bakalárskej práce bolo navrhnúť a vytvoriť užívateľské prostredie ako nástroj na podporu rozhodovania pre zabezpečenie prevádzky obslužného alebo distribučného systému so zameraním na operatívne pridelovanie prostriedkov obsluhy zákazníkom.

Na začiatku vypracovávania praktickej časti bakalárskej práce, som zozbieral požiadavky a informácie. Následne som analyzoval jednotlivé zložky ZZS a zo zistených skutočností som podrobnejšie vyšpecifikoval požiadavky pre informačný systém ZZS. Pristúpil som k návrhu informačného systému, kde najdôležitejšou a najzložitejšou časťou navrhovania bolo vytvorenie databázovej schémy. Pri navrhovaní schémy vznikali mnohé komplikácie a problémy, ktoré súviseli hlavne s rozsiahlosťou a rôznorodosťou samotného zloženia ZZS.

Po vyriešení problémov, ktoré vznikali pri navrhovaní jednotlivých častí informačného systému, som pristúpil k tvorbe softvéru. Tak som vytvoril informačný systém, ktorý poskytuje jednotlivým aktérom ZZS informačnú podporu a obsahuje tri užívateľské pohľady prispôbosené pre potreby jednotlivých aktérov ZZS.

Hlavným aktérom informačného systému je operátor, ktorý obsluhuje tiesňové volania a následne v prípade potreby pomocou programu vyberie vhodnú posádku záchranky na zásah. Ďalej pomocou telefonickú, prípadne rádiovú komunikáciu s posádkou manažuje riadenie zásahu a zapisuje jednotlivé skutočnosti do systému.

Druhým aktérom je veliteľ záchranky, ktorý pomocou informačného systému prijme rozkaz na zásah a následne vyšle posádku plniť rozkaz. Podobne ako operátor môže pomáhať pri organizácii zásahu s možnosťou dopĺňania informácií do systému.

Posledný užívateľ informačného systému je lekár na príjme v nemocnici, ktorý získava zo systému informáciu o priebehu zásahu a zdravotnom stave pacienta prichádzajúceho zo ZZS a do systému dopĺňa informáciu o príchode a odjazde sanitky.

Hotový informačný systém možno rozširovať a dopĺňať o rôzne funkcie, vylepšenia, aplikácie a pod. V prípade potreby, by sa dala urobiť aplikácia pre posádku, ktorá by na základe komunikácie s databázou informačného systému ZZS poskytovala informácie priamo na zásahu. V prípade nainštalovania lokalizačného systému do sanitiek ZZS, by sa dal informačný systém prerobiť tak, aby sa využívala pri vyberaní najvhodnejšej posádky aktuálna poloha vozidla ZZS. Čím by sa ešte viac zoptimalizovala stratégia výberu posádky záchrany a skrátil sa čas príchodu k pacientovi ZZS. Veľakrát je čas najdôležitejší faktor pri záchrane ľudského života.

Na základe uvedených skutočností, môžem povedať, že informačný systém po drobných úpravách by bolo možné využívať i v iných záchranných zložkách, napr. Hasičský a záchranný zbor, Policajný zbor. Čiastočnou úpravou entít zabezpečujúcich zásah, by bolo možné vytvoriť systém aj na prevádzku iných logistických systémov so zameraním na operatívne pridelovanie prostriedkov obsluhy zákazníkov (rozvoz tovaru zo skladov, doprava, a pod).

Zoznam použitých zdrojov

1. Ekonomicko právne informácie. [Online] <http://www.epi.sk/zz/2004-579#c11>.
2. **Záchranná zdravotná služba Bratislava** . [Online] <http://emergency-ba.sk/zdzs/zachranna-zdravotna-sluzba>.
3. **Life Star Emergency spol. s.r.o.** [Online] <http://www.lse.sk/>.
4. **HPI - Stredoeurópsky inštitút pre zdravotnú politiku.** [Online] <http://www.hpi.sk/2010/01/analyza-systemu-zachrannej-zdravotnej-sluzby-po-reforme/>.
5. **Ing. Ján Ružbarský, PhD.** *Prednášky softvérové modelovanie*.
6. **Matiaško J., Vajsová M., Zábovský M.n Chochlík M.** *Základy databázových systémov*. Žilina : Edis, 2008. 9788080708207.
7. **KMMOA FRI UNIZA.** Matica vzdialenost miest a obcí SR. [Online] <http://frdsa.fri.uniza.sk/~betka/BenchmarksSR.html>.
8. **JasperReport.** *JasperReports Library.* [Online] <https://sourceforge.net/projects/jasperreports/files/>.
9. **Google.** Get API key. [Online] <https://developers.google.com/maps/documentation/geocoding/get-api-key>.
10. —. Matrix distance documentation. [Online] <https://developers.google.com/maps/documentation/distance-matrix/intro>.
11. **Schildt, Herbert.** *Mistrovství - Java*. Brno : Computer Press, 2014. 978-80-251-4145-8.
12. **Záchranná zdravotná služba ZaMED.** [Online] <http://www.zamed.sk/ako-funguje-zachranka/>.
13. **Operačné Stredisko ZZS SR.** [Online] <http://www.emergency-slovakia.sk/>.
14. **Google.** Dokumentácia Distance Matrix. [Online] <https://developers.google.com/maps/documentation/distance-matrix/start>.

Prílohy

Príloha A: Obsah DVD

Priložené DVD obsahuje:

- Práca v elektronickej podobe (formát PDF)
- Zdrojový kód aplikácie
- Skript databázovej schémy
- Export databázy
- Video ukážky použitia programu
- Diagramy