

UNIVERZITA PARDUBICE

Fakulta elektrotechniky a informatiky

Software pro dobrovolné hasiče generující zprávy
o zásahu

Roman Petrásek

Bakalářská práce
2015

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2014/2015

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Roman Petrásek**
Osobní číslo: **I11148**
Studijní program: **B2646 Informační technologie**
Studijní obor: **Informační technologie**
Název tématu: **Software pro dobrovolné hasiče generující zprávy o zásahu**
Zadávající katedra: **Katedra informačních technologií**

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

Cílem práce je navrhnout a implementovat software pro vytváření zpráv o výjezdech pro dobrovolné hasiče. V teoretické části bude provedena analýza požadavků dobrovolných hasičů na software pro vytváření zpráv. Dále porovnání alternativních systémů umožňující generování zpráv a popis využití technologií .NET/C# pro vývoj uvedené aplikace. V praktické části bude vytvořen software pro podporu vytváření a rozesílání zpráv o výjezdech pro dobrovolné hasiče. Aplikace umožní interaktivní vytvoření zprávy s využitím předem definovaných hodnot personálu, nasazené techniky aj. Vytvořenou zprávu bude možné vytisknout a rozeslat e-mailem. Program bude implementován v jazyce C#.

Rozsah grafických prací:

Rozsah pracovní zprávy:

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

Seznam odborné literatury:

[1] NAGEL CH. et al. C# 2008. Programujeme profesionálně. Brno, 2009. ISBN 978-80-251-2407-7.

[2] PETZOLD, Charles. Programování Microsoft Windows v jazyce C#. Praha : SoftPress, 2003. 1208 s. ISBN 80-86497-54-2

[3] Standard ECMA-334: C# Language Specification. ECMA International [online]. 2006 [cit. 2014-09-23]. Dostupné z:

<http://www.ecma-international.org/publications/standards/Ecma-334.htm>

Vedoucí bakalářské práce:

Ing. Roman Diviš

Katedra softwarových technologií

Datum zadání bakalářské práce:

20. prosince 2014

Termín odevzdání bakalářské práce:

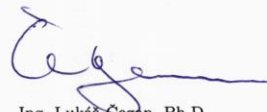
11. května 2015



prof. Ing. Simeon Karamazov, Dr.
děkan



L.S.



Ing. Lukáš Čegan, Ph.D.
vedoucí katedry

V Pardubicích dne 31. března 2015

Prohlášení autora

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

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

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 26. 4. 2015

Roman Petrásek

Poděkování

Rád bych poděkoval vedoucímu práce Ing. Romanu Divišovi, za trpělivost a pomoc při tvorbě bakalářské práce kdykoli jsem potřeboval, svému strýci Miroslavu Barešovi za konzultace a poskytnutí veškerých potřebných informací pro dokončení mé bakalářské práce, své přítelkyni Zuzaně Milické, za pevné nervy, podporu a pomoc po celou dobu studia, celé své rodině za zázemí a podporu při studiu a svým kamarádům, zejména Ondřeji Krausovi, Josefu Hrodkovi, Jitce Pomikálkové, Janu Burešovi, Zdenku Nouzovi, Nice Beranové a Tereze Šťastné za pomoc a podporu po celou dobu studia.

Anotace

Cílem práce je vytvoření softwaru pro dobrovolné hasiče, pomocí kterého budou moci jednoduše vytvářet a spravovat zprávy o zásahu. Zprávu o zásahu musí tvořit sbor po každém výjezdu. Software bude umožňovat předepsat hodnoty, které se automaticky vygenerují do každé nové zprávy. Při vytváření nové zprávy bude po zadání minimálního počtu potřebných údajů, které nelze automatizovat, vygenerována zpráva připravená pro tisk, kterou software odešle na předem definované e-maily. Práce bude implementována v jazyce C#.

Klíčová slova

C#, zpráva o zásahu, dobrovolní hasiči, generování zprávy, NDatabase, PDFSharp

Title

Software for volunteer firefighters generating intervention reports.

Annotation

The aim of this thesis is to create a software for volunteer firefighters that would help them with report's writing and administrating. These reports have to be created for every single action. The main advantage of this software is that it enables the user to predefine some information, which are later generated automatically. That means that when somebody writes the report, only a few information are needed (only the data which can't be automatized). Right after the finishing the report, the generated report is ready for printing or for sending to predefined email addresses. The program is written in C# programming language.

Keywords

C#, report intervention, volunteer firefighters, generating report's, NDatabase, PDFSharp

Obsah

Seznam zkratk.....	8
Seznam obrázků.....	9
Seznam tabulek.....	9
Úvod.....	10
Použité konvence	10
1 Zprávy o zásahu.....	11
1.1 Základní charakteristika	11
1.2 Typy zpráv.....	11
1.2.1 Zpráva o zásahu	11
1.2.2 Dílčí zpráva o zásahu.....	11
1.2.3 Zpráva o činnosti	11
1.3 Zpráva o zásahu	12
1.3.1 Základní informace.....	12
1.3.2 Hasiči u zásahu	12
1.3.3 Automobily.....	12
1.3.4 Čas a datum	12
1.3.5 Použité prostředky	13
1.3.6 Ostatní informace	13
1.4 Psaní zprávy.....	13
1.5 Definice pojmů	17
2 Programovací jazyk C#.....	19
2.1 Základní informace.....	19
2.2 Historie	19
2.3 Vlastnosti jazyka	19
2.4 Konvence jazyka.....	20
2.5 DLL	20
2.6 Prvky jazyka	21
2.6.1 Třída	21
2.6.2 Pole	21
2.6.3 Rozhraní	21
2.6.4 Operátory.....	21

2.6.5	Výjimky	22
3	NDatabase	23
3.1	Objektová databáze.....	23
3.2	Základní informace.....	23
3.3	Princip.....	23
3.4	Produktivita	24
3.5	Použití.....	24
4	PDFSharp.....	25
4.1	PDF.....	25
4.2	Základní informace.....	25
4.3	Vlastnosti	26
4.4	Práce s knihovnou.....	26
5	Implementace aplikace.....	28
5.1	Návrh aplikace.....	28
5.2	Třídy aplikace	31
5.2.1	Objekty zprávy	31
5.2.2	HlavniProgram	32
5.2.3	Konfigurator	33
5.2.4	NovaZprava	33
5.2.5	OdeslaniNaMail.....	34
5.2.6	Tutorial	35
5.2.7	Zpráva.....	35
5.3	Práce s databází	35
5.4	Export do PDF.....	36
5.5	Možnost úpravy zprávy	37
5.6	Zabezpečení	38
5.6.1	Správné formáty	38
5.6.2	Povinné položky	38
5.6.3	Zabezpečení databáze	38
5.6.4	Smazání zprávy.....	38
5.6.5	Odesílání e-mailu.....	38
5.6.6	Odstranění souborů.....	38
5.7	Uživatelská přívětivost	39

Závěr	41
Literatura	42
Příloha A – UML Diagram	43
Příloha B – Manuál k aplikaci.....	43

Seznam zkratek

ACID	Atomicity consistency isolation durability
CLR	Common language runtime
DDL	Data definition language
DLL	Dynamic-link library
ECUD	Evidenční číslo události
HZS ČR	Hasičský záchranný sbor České republiky
IZS	Integrovaný záchranný systém
JPO	Jednotka požární ochrany
LINQ	Language integrated query
LZW	Lempel-Ziv-Welch
MSDN	Microsoft developer network
PDF	Portable document format
SDH	Sbor dobrovolných hasičů
SHČMS	Sdružení hasičů Čech, Moravy a Slezska
SMTP	Simple mail transfer protocol
SODA	Simple object database access
ZOZ	Zpráva o zásahu

Seznam obrázků

Obrázek 1 – Formulář pro zprávu o zásahu	14
Obrázek 2 – Tabulkový editor pro ZOZ	15
Obrázek 3 – CZOZ offline	16
Obrázek 4 – Kostra programu vygenerovaná při vytvoření konzolové aplikace	20
Obrázek 5 – Výstup ve formátu PDF	27
Obrázek 6 – Úvodní tutoriál aplikace	28
Obrázek 7 – Hlavní program	29
Obrázek 8 – Formulář Nové zprávy o zásahu	30
Obrázek 9 – Pole pro automobil	30
Obrázek 10 – Odeslání na e-mail	31
Obrázek 11 – Odeslání na e-mail	31
Obrázek 12 – Výstup ve formátu PDF	37

Seznam tabulek

Tabulka 1 – Porovnání možnosti psaní ZOZ	17
Tabulka 2 – Operátory	22

Úvod

Dobrovolní hasiči v dnešní době musí psát zprávu o zásahu po každém výjezdu, ať se jedná o požár, povodně, nebo uvíznutí kočky na stromě. Hasiči musí psát zprávy o zásahu i v případě, že se jedná o planý poplach, nebo že jednotka k zásahu vůbec nevyjela. Formuláře jsou dlouhé a obsáhlé, což je pro hasiče, kteří tuto činnost vykonávají dobrovolně zdoluhavá, a velmi často i zbytečná práce.

Po konzultaci s vedením SHČMS jsem se dozvěděl, že v současné době žádný program určený pro tvorbu zprávy o zásahu pro dobrovolné hasiče neexistuje. Ulehčení v současné době umožňuje předvyplněný formulář v tabulkovém editoru Microsoft Excel, ten je však také časově náročný. Další z možností je program CZOZ vytvořený firmou RCS Kladno. Tento software je ale určen pro profesionální jednotky a dobrovolným hasičům je umožněno psaní zpráv v offline verzi programu, která má jen omezené možnosti a pro potřebu dobrovolných hasičů je nedostačující.

V teoretické části jsou probrány požadavky dobrovolných hasičů na aplikaci, porovnání alternativních systémů umožňující psaní zpráv o zásahu, popis technologií NET/C# s využitím PDFSharp a objektové databáze NDatabase.

Hlavním cílem práce bylo vytvořit aplikaci umožňující interaktivní vytvoření zprávy o zásahu s využitím předem definovaných hodnot personálu, techniky aj. Vytvořený program umožní generování zpráv do formátu PDF a odesílání zpráv elektronickou poštou na předem definované e-mailové adresy.

Bakalářská práce má následující strukturu. V první části jsou probrány požadavky hasičů na aplikaci a porovnání alternativních systémů. Ve druhé části je probrán programovací jazyk .NET/C#. Třetí část obsahuje informace o použité objektové databázi NDatabase. Ve čtvrté části je probrána knihovna určená pro generování formátu PDF. V páté části je probrána vlastní implementace aplikace pro generování zpráv o zásahu.

Použité konvence

Kurzíva – použita pro pojmy v anglickém jazyce, pro které neexistuje adekvátní český překlad, nebo není z důvodu srozumitelnosti překládán.

`Neproporcionální styl písma` – části kódu, programové příkazy, apod.

1 Zprávy o zásahu

1.1 Základní charakteristika

Zpráva o zásahu, dílčí zpráva o zásahu nebo zpráva o činnosti tvoří součást dokumentace o událostech a činnostech jednotek dobrovolných hasičů sloužící pro potřebu HZS ČR. Dobrovolní hasiči, na rozdíl od hasičů profesionálních, jsou neziskovou organizací a všichni její členové vykonávají tuto činnost dobrovolně ve prospěch druhých, bez nároku na finanční odměnu. Dobrovolní hasiči kromě hašení požárů zasahují i u živelních pohrom či jiných mimořádných událostech.

Zprávu o zásahu zpracovává velitel zásahu příslušné jednotky sboru dobrovolných hasičů. Jejím obsahem jsou informace o použitých prostředcích a technice vyslané na místo zásahu, seznam zúčastněných hasičů a jednotek včetně speciálních skupin, hodnocení situace při příjezdu první ze zasahujících jednotek na místo zásahu, postup v místě zásahu a záznamy o poskytnuté osobní a věcné pomoci. Ve zprávě jsou uvedeny veškeré časové údaje o činnosti jednotky při zásahu. Zpráva se odesílá na operační středisko, pod které spadá sbor dobrovolných hasičů. Operační středisko je u všech sborů, kde působí profesionální hasiči.

Povinnost napsat zprávu nastává i v případě, že byl sbor povolán k výjezdu, ale k zásahu nedorazil. V tomto případě je nutné uvést důvod absence jednotky u zásahu.

1.2 Typy zpráv

1.2.1 Zpráva o zásahu

Zprávu o zásahu píše velitel zásahu. Velitel zásahu se určí podle nadřazenosti sboru, v případě, že ke sboru vyjeli pouze dobrovolní hasiči, velitel zásahu je určen podle velikosti sboru dobrovolných hasičů. Pokud jsou u zásahu i jednotky profesionálních hasičů, velitel požáru je z řad profesionálních hasičů. Zpráva o zásahu je hlavní zpráva o celé události složená z dílčích zpráv o zásahu popsanych níže.

1.2.2 Dílčí zpráva o zásahu

Dílčí zprávu o zásahu píše velitel každého sboru, který se zúčastnil výjezdu a zároveň nebyl velitelem zásahu. Zprávu píše z pohledu své jednotky a tu potom předá veliteli zásahu. Dílčí zpráva se následně použije do hlavní zprávy o zásahu.

1.2.3 Zpráva o činnosti

Zprávu o činnosti musí sbory psát v případě, že provádí nějakou činnost, ale nejedná se o zásah. Jedná se o předem plánované akce, např. pro obec nebo firmu, které musí být předem nahlášeny na operační středisko, odkud musí být schváleny. Činností, při které se nejedná o zásah, se rozumí například kácení nebezpečných stromů nebo mytí komunikací. Tato činnost je počítána do statistiky zásahů, ale není třeba kvůli ní psát zprávu o zásahu.

1.3 Zpráva o zásahu

1.3.1 Základní informace

Každá zpráva o zásahu musí obsahovat veškeré informace o sboru a zásahu:

- typ zprávy, v současné době jsou definovány tři typy zpráv (zpráva o zásahu, zpráva o činnosti a dílčí zpráva),
- název jednotky požární ochrany a její evidenční číslo,
- ECUD (evidenční číslo události), je u každé zprávy jedinečné,
- typ události charakterizující událost, ke které byl sbor povolán,
- adresa, objekt a prostor, kde byl proveden zásah,
- vzdálenost, kterou bylo nutno urazit k zásahu,
- pořadí, v jakém sbor na místo dorazil,
- počet zásahů se vyplní v případě, že byl sbor ze zásahu odvolán k jinému, aniž by se vrátil na základnu,
- zda bylo zapotřebí nasadit speciální skupiny,
- seznam činností jednotky, které byly u zásahu provedeny.

1.3.2 Hasiči u zásahu

Ve zprávě musí být uvedeni všichni hasiči sboru, kteří se zásahu zúčastnili a jejich hodnosti u zásahu (hasič, strojník, velitel zásahu, velitel jednotky, nebo velitel družstva). V případě zranění hasiče je nutné uvést vážnost zranění a podrobný popis události, která k němu vedla.

1.3.3 Automobily

Zpráva musí obsahovat seznam všech automobilů, které byly povolány k výjezdu, počet ujetých kilometrů a motohodin (počet hodin, kdy byl motor v činnosti). Dále je třeba uvést čas, kdy byl automobil povolán, kdy přijel k zásahu a kdy se vrátil zpět na základnu.

1.3.4 Čas a datum

Součástí zprávy je přesný časový rozpis činností souvisejících se zásahem:

- ohlášení zásahu,
- vyhlášení poplachu,
- výjezd sboru k zásahu,
- příjezd k zásahu,
- zahájení zásahu,
- odjezd zpět na základnu,

- návrat na základnu.

1.3.5 Použité prostředky

Zásadní je také uvedení všech použitých hasiv a sorbentů použitých u zásahu, včetně typu hasiva a spotřebovaného množství. Do použitých prostředků spadají také ochranné prostředky (zde jsou vypsána jména hasičů, kteří ochranný prostředek použili a jak dlouho jej používali) a veškerá pomocná technika, které bylo využito při zásahu. Přesná specifikace prostředků je samozřejmostí.

1.3.6 Ostatní informace

Na konci zprávy je uveden popis zásahu, využitá osobní a věcná pomoc od civilních osob nebo společností, případná zranění příslušníků SDH či dalších osob zraněných během zásahu. Do ostatních informací je dále zahrnut seznam členů IZS, kteří se zúčastnili zásahu.

1.4 Psaní zprávy

Dobrovolní hasiči v současné době mají možnost psát zprávu o zásahu pomocí papírových formulářů, které jsou předtištěné. Vše potřebné je vyplněno a následně odesláno poštou na velitelství HZS spravujícího daný sbor dobrovolných hasičů. Tento způsob psaní zprávy se používá od doby, kdy se zprávy o zásahu začaly psát. Nicméně v dnešní době počítačů je tento způsob zdlouhavý a neefektivní. Na následujícím obrázku (Obrázek 1) je zobrazen papírový formulář vyplňovaný dobrovolnými hasiči při každém zásahu.

(DÍLČÍ) ZPRÁVA O ZÁSAAHU						CENNOST – pořadí jednotky PO					CENNOST – pořadí jednotky PO					
						1	2	3	4	5	1	2	3	4	5	
DATUM	ADRESA (obec, ulice, č.p., SPZ)					000 Jednotka nezastavovala (odvolána při cestě k zásahu)						445 Ověření místa zásahu				
Evidenční číslo události						100 Požární asistence						450 Práce na vodě				
UDALOST						110 Asistence při vyhlídce likvidace (např. výbuš. jvyt.)						460 Práce ve vodě a pod vodou				
12 Požár						120 Prásknutí						470 Obnova nebezpečného zařízení				
Dopravní nehoda	Měsíční pomoc					130 Použití hasicích přístrojů						480 Provizorní oprava				
21 – Silniční	OBJEKT					140 Použití jednoduších hasicích prostředků						490 Rozebírání konstrukcí				
22 – Silniční hromada						145 Použití vody dodávané proudem D – počet proudů						500 Uzavírání vody, plynu, elektřiny				
23 – Zeleništní vř. metra						150 Použití vody dodávané proudem C – počet proudů						510 Vnikání do uzavřeného prostoru				
24 – Letecká	PROSTOR					160 Použití vody dodávané proudem B – počet proudů						520 Zásah ve výš. pomocí lezec. tech.				
25 – Ostatní*						170 Použití vody a odtah. proudnice – počet						530 Zásah ve výš. a nad vol. hloubkou				
Unik. nebezpeč. chem. látky						180 Použití vysokotlaké vody – počet proudů						540 Vyhledávání osob				
41 – Plynu aerosolu	CASOVÉ ÚDAJE					190 Použití plynů lehké – počet agregátů						550 Vyhledávání, záchrana osob z vody				
42 – Kapalin	Měs. Den Hod. Min.					200 Použití plynů střední – počet proudů						560 Vyprošťování osob a kloubek				
43 – Ropného produktu	Ohřívání					210 Použití plynů těžké – počet proudů						570 Vyprošťování osob a vytek.				
44 – Pevné látky	Vyhášení poplachu					220 Použití smetadla						580 Vyprošťování osob a havarov. voz.				
45 – Ostatní*	Výjezd jednotky					230 Použití prachu a mobilní techniky						590 Vyprošťování osob a vytahu				
Technická havárie	Příjezd k zásahu					240 Použití laserových plynů a mobilní techniky						600 Vyprošťování se skřesocen. staveb				
51 – Technická havárie	Zásahy na zásahu					250 Hádání avistami technickými prostředky a hasičy						610 Záchrana osob (jiná)				
52 – Technická pomoc	Odpad na základě					260 Čerpání a odčerpávání vody						620 Předkvalifika. pomoc				
53 – Technologická pomoc	Příjezd na základě					270 Dálková doprava vody hadicemi						630 Vyprošťování předmětů				
54 – Ostatní pomoc*	VZDALENOST OD ZÁKLADNY (km)					280 Dálková doprava vody kyvadlová						640 Odchyt a likvidace vř. vyhledávání				
61 Radiac. havárie a nehoda	CENNOST PŘED PŘÍJEZDEM JEDNOTEK PO-					290 Dopřívání vody						650 Odchyt a likvidace obut. hmyzu				
71 Opat. mimořádn. uděl.*	Samohlášení					300 Chlazení						660 Evaluace osob objektů				
81 Plavby poplach	Zásahy občanů, zaměstnanců					310 Odvětrání prostorů plynem						670 Evaluace osob plochy				
PRÍZNÁK – hřivní pohroma	Zásahy jiných služeb					320 Odvětrání prostorů znečištění						680 Evaluace předmětů				
Porodní, záplava, dešť	Pořadí jednotky PO*					330 Izolace, separace látek						690 Evaluace zvířat				
Směš. náprava	Evidenční číslo jednotky PO					340 Neutralizace						700 Zřízení a záj. provozu evakuačního				
Větrná smršť	Název jednotky PO					350 Řázení						710 Opatření nebezpečných oblastí				
Savon. plyn						360 Přetřepávání látek						720 Dekontaminace osob vř. hasičů				
Ostatní*						370 Ochránění, znečištění unáše látek						730 Dekontaminace techniky a prost.				
						380 Jímání, sběr unáše látek (mimo rop. produktů)						740 Zprovoznění úkrytů				
						390 Zjišťování druhu unáše látek						750 Doprava pitné vody, potravin				
						391 Odběry vzorků						760 Výdej a rozdělení pitné vody, potravin				
						395 Měření koncentrací plynů						770 Tělové práce				
						400 Zajištění místa nehody						780 Čekání na speciální služby				
						410 Odstraňování následků dopravní nehody						790 Záloha na místě události				
						415 Řízení provozu na komunikacích						800 Pohotovost na vlastní stanici				
						420 Odstraňování překážek a komunikací a. prostorů						810 Záloha na stanici				
						430 Odstraňování unáše rop. látek – provoz. náplní vozidel						820 Jine.*				
						440 Ochrana státek										

Obrázek 1 – Formulář pro zprávu o zásahu

Z důvodu úspory času byla založena šablona v tabulkovém editoru Microsoft Excel. Tato šablona umožňuje vyplnění zprávy o zásahu na počítači a její odeslání na operační středisko pomocí elektronické pošty. Zpráva musí být po odeslání vytisknuta a uložena do archivu. Tento způsob psaní zprávy zjednodušil velitelům sborů psaní zpráv, ale i tak je tento způsob zdoluhavý, poněvadž při každé zprávě je zapotřebí znovu vyplňovat většinu polí ručně. V obrázku 2 je zobrazena část šablony určené pro zprávy o zásahu v tabulkovém editoru.

	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AAA	ACA	AE	AF	AG	AH	AI	AJ	AK	AL	AN	AO	AP
2	Dílčí zpráva o zásahu																																					
4																																						
6	<u>Základní údaje události</u>																																					
8	Typ události SSU:															Evidenční číslo události:																						
10	Popis typu:															Adresa:																						
12	Objekt:																																					
14	Prostor:																																					
17	Datum:															Čas:																						
19	Ohlášení:															Počet jednotek:																						
21	Lokalizace:															Místní JPO:																						
23	Likvidace:															Výkon proudů:																						
26	<u>Základní údaje jednotky</u>																																					
28	Jednotka PO:															Velitel jednotky:																						
30	Druh činnosti, vč. počtu proudů:															#																						
32																Vložit datum ohlášení události do všech polí																						
34																Datum:																						
36																Čas:																						
38																Pořadí jednotky:																						
40																Vzdálenost [km]:																						
42																Počet osob:																						
44																Počet techniků:																						
46																Počet zásahů:																						
49	<u>Technika jednotky u zásahu</u>																																					
51	Název techniky:															Volací znak:																						
53																Řidič:																						
55																Velitel:																						
57	Povolání:															Ujeto [km]:																						
59	Výjezd:															Motohodiny:																						
61	Na místě:															Č. 1																						
63	Odjezd:															Č. 2																						
65	Na základně:															Č. 3																						
67	Pohotovost:															Č. 4																						
70																Č. 5																						
72																Č. 6																						
74	<u>Časové údaje k jízdě</u>																																					
76	Povolání:															Ujeto [km]:																						
78	Výjezd:															Motohodiny:																						
80	Na místě:															Č. 1																						
82	Odjezd:															Č. 2																						
84	Na základně:															Č. 3																						
86	Pohotovost:															Č. 4																						
																Č. 5																						
																Č. 6																						

Tisk formuláře

Vymazání
formuláře

Obrázek 2 – Tabulkový editor pro ZOZ

Další z možností je program CZOZ vytvořený firmou RCS Kladno. Nejnovější verze 5.0 (údaj z 23. 3. 2015) umožňuje efektivní psaní zpráv o zásahu. Program si pamatuje údaje, které byly použity již dříve, a při psaní další zprávy je možné je použít. Program umožňuje uložení rozpracovaných zpráv, úpravu vytvořených zpráv, odesílání zpráv na předem zadané e-maily, export do několika různých formátů a možnost tisku. Bohužel, tento program není určen pro dobrovolné hasiče, ale výhradně pro profesionální složky IZS. Dobrovolní hasiči mohou používat po zakoupení offline verzi programu, ta však neumožňuje odesílání na e-maily, má omezené možnosti exportu a jeho hlavní slabinou je možnost psaní pouze dílčí zprávy o zásahu. Na obrázku 3 je ukázka hlavní stránky programu CZOZ ve verzi offline.

Agenda jSDH - [Offline klient]							
	Stav	Datum	ECUD	Ev. č. jednotky PO	Typ události	Místo události	
Seznam ZOZ	► D ZOZ	Rozpracovaná	5.11.2009	2109000001	223344		
	D ZOZ	Rozpracovaná	5.11.2009	2109000001	223344		
	D ZOZ	Rozpracovaná	5.11.2009	9000000			
	D ZOZ	Rozpracovaná	5.11.2009	2109000001	223344		
Nová ZOZ	D ZOZ	Rozpracovaná	25.8.2009	2109000001	223344		
	D ZOZ	Rozpracovaná	25.8.2009	2109000001	12121244	Dopravní nehoda silniční	122
Import ZOZ	D ZOZ	Exportovaná	24.8.2009	2109000001	223344	Dopravní nehoda silniční hromadná	12
Detail ZOZ							
Uložit ZOZ							
Uzavřít ZOZ							
Exportovat ZOZ							
Náhled ZOZ (tisk)							
Smazat ZOZ							

Obrázek 3 – CZOZ offline

V následující tabulce je srovnání všech možností vyplnění zpráv o zásahu vzhledem k jednoduchosti, uživatelské přívětivosti a časové náročnosti. Časová náročnost byla ozkoušena psaním zpráv několika typů událostí a byla prodiskutována s vedením SHČMS a členy výboru SDH Uhlířské Janovice.

Tabulka 1 – Porovnání možnosti psaní ZOZ

	Papírový formulář	Tabulkový formulář	CZOZ Offline	Nový software
Možnost tvorby celé ZOZ	Ano	Ano	Ano	Ano
Možnost tvorby na PC		Ano	Ano	Ano
Možnost tisku		Ano	Ano	Ano
Možnost psaní všech druhů ZOZ	Ano			Ano
Interaktivní vyplnění			Ano	Ano
Export do PDF			Ano	Ano
Elektronické uchování zprávy			Ano	Ano
Předdefinované hodnoty				Ano
Odeslání na e-mail				Ano
Průměrná doba vyplnění	45 min	30 min	16 min	7 min

1.5 Definice pojmů

Jednotka požární ochrany (jednotka) – „síly a prostředky soustředěné a pracující pod jednotným velením velitele jednotky a náležející ke stejnému druhu jednotky (HZS kraje, hasičského záchranného sboru podniku, sboru dobrovolných hasičů obce, sboru dobrovolných hasičů podniku) označené vlastním evidenčním číslem jednotky.

Za jednotku je považována každá stanice HZS kraje, tj. veškeré síly a prostředky jednoho druhu jednotky ze stejného místa dislokace soustředěné na místě zásahu nezávisle na době příjezdu na místo zásahu. “

Zásah – „činnost jednotky a složky IZS u události včetně prověřovacího cvičení nezávisle na tom, zda je za ni požadována finanční náhrada nebo ne.

Za zásah jednotky se považuje také předem dohodnutá pomoc jiným službám (např. policii při vyhledávání osob, předmětů, otvírání prostor apod.) nebo požár či záchranné práce, které vyvstanou z účasti jednotky (která není za zásah považována) při zajištění sportovní či kulturního akce.“

Požár - definice viz § 1 písm. m) vyhlášky č. 246/2001 Sb. „Za požár se považují případy naplňující znaky požáru dle citované vyhlášky, přitom není rozhodující, zda a kým byl požár likvidován, došlo-li k samouhašení nebo byl-li požár nahlášen příslušnému HZS kraje dodatečně. Za požár se považují také výbuchy hořlavých par, plynů a prachů bez následného plamenného hoření, pokud se nejednalo o fyzikální výbuch.“

Dopravní nehoda – „zásah u události mající charakter činností spojených s odstraňováním následků kolize dopravních prostředků.

Druh dopravní nehody se zatřídí podle převažujícího charakteru nehody.

Pokud by v činnosti jednotky převládaly jiné práce, např. z důvodu úniku nebezpečné látky do životního prostředí, klasifikuje se tento zásah rovněž podle převažujícího charakteru.“

Planý poplach – „činnost jednotky vyvolaná z důvodu ohlášení požáru nebo jiné události jednotce, která se nepotvrdila.“

2 Programovací jazyk C#

2.1 Základní informace

C# je vysokoúrovňový programovací jazyk, založen na jazycích C++ a Java, čerpající syntaxi z jazyka C, je jeho nepřímým potomkem. Programovací jazyk C# vyvinula firma Microsoft, zároveň s platformou .NET Framework.

C# se nejčastěji používá k formulářovým aplikacím ve Windows, tvorbě databázových programů, webových aplikací a stránek, webových služeb a další. Jazyk je vhodný pro vývoj softwarových komponent distribuovaných v různých prostředích. Obsahuje nativní podporu komponentového programování, je navržen pro psaní aplikací pro zařízení s operačními systémy, i pro zařízení s omezenými možnostmi. C# je také integrován do Visual Studio.NET.

2.2 Historie

První verze C# 1.0 byla vydána v roce 2002 společně s .NET Framework. První verze obsahovala podporu objektového programování, ve které vycházela z jazyka C++ a zkušeností s jejich aktualizací v jazyce Java.

Verze 2.0 vydaná na konci roku 2005 obsahovala částečné a statické třídy, iterátory, anonymní metody, nullovatelné hodnotové typy aj.

C# 3.0 vydaný na konci roku 2007 společně s Visual Studií 2008 a s .NET Framework 3.5. Obsahuje velké změny, aplikace v něm psané půjdou spouštět i na počítačích vybavených druhým Frameworkem, ponesou-li si s sebou patřičné knihovny. Verze 3.0 obsahuje LINQ (Language Integrated Query), integrovaný dotazovací jazyk, který umožňuje nový způsob dotazování nad jakýmkoliv daty, usnadňuje tvorbu, třídění a vyhledávání dat. LINQ to Objects umožňuje dotazování nad kolekcemi objektů, LINQ to SQL zjednodušuje práci s databázemi a LINQ to XML umožňuje práci se soubory XML.

Verze 4.0 vyšla v roce 2010. Vylepšuje zejména práci s dynamickými aspekty programování a frameworky. Mezi nové vlastnosti patří volitelné a pojmenované parametry, dynamicky typované objekty, nebo kovariance a kontravariance.

V srpnu roku 2012 byla vydána prozatím nejnovější verze jazyku 5.0 společně s .NET Framework 4.5 a vývojovým prostředím Visual Studio 2012. Novinkou v této verzi je možnost asynchronního programování pomocí klíčových slov *async* a *await*.

2.3 Vlastnosti jazyka

Jazyk C# je čistě objektově orientovaný, obsahuje nativní podporu komponentovaného programování, poskytuje hlídání hranic polí, detekci použití neinicializovaných proměnných, automatickou správu paměti, o korektní uvolňování zdrojů se

stará *garbage collector*. Poskytuje pouze jednoduchou dědičnost s možností násobné implementace rozhraní.

Zajišťuje zpracování chyb pomocí výjimek, typovou bezpečnost, podporu řízení verzí a atributové programování. Samozřejmostí je zpětná kompatibilita se stávajícím kódem na binární i zdrojové úrovni.

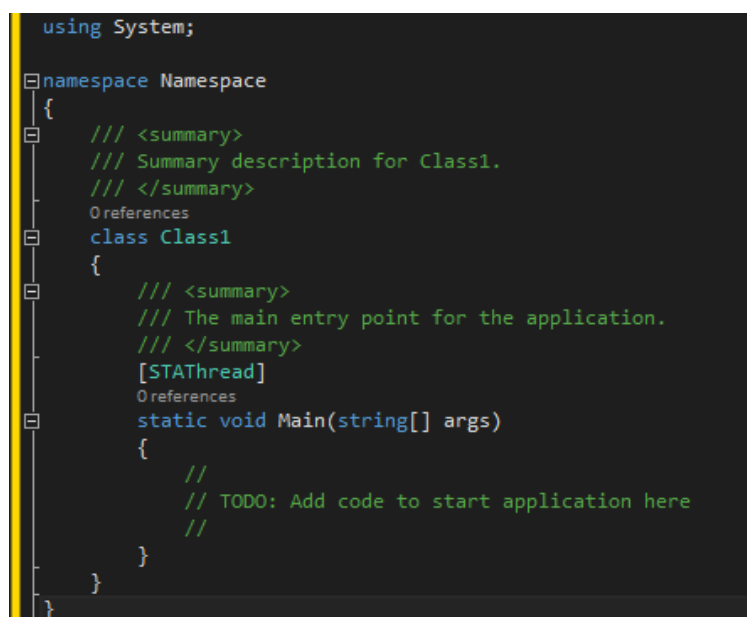
Překladač rozlišuje velká a malá písmena, tedy je case sensitive.

2.4 Konvence jazyka

Jako v jiných jazycích, bylo i v jazyce C# zavedeno několik konvencí.

Jména tříd, rozhraní, balíků a dalších položek začínají velkým písmenem. Malým písmenem začínají lokální proměnné, parametry, chráněné (*protected*) a privátní atributy.

Na obrázku (Obrázek 2) je zobrazena jednoduchá kostra programu, vygenerována Visual Studiem po vytvoření konzolové aplikace.



```
using System;

namespace Namespace
{
    /// <summary>
    /// Summary description for Class1.
    /// </summary>
    References
    class Class1
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        References
        static void Main(string[] args)
        {
            //
            // TODO: Add code to start application here
            //
        }
    }
}
```

Obrázek 4 – Kostra programu vygenerovaná při vytvoření konzolové aplikace

2.5 DLL

DLL (Dynamická knihovna) je jeden spustitelný soubor, pracující jako sdílená knihovna funkcí. Umožňuje procesu volat funkci, která není součástí jeho kódu, ale je umístěna v knihovně DLL. Může obsahovat jednu nebo více funkcí, které se kompilují odděleně od procesů, které tyto funkce využívají. Pomocí DLL lze jednoduše sdílet data a prostředky. Přistupovat k obsahu jedné knihovny DLL v paměti může současně několik aplikací. Dynamické propojení obsahuje pouze informace nutné k nalezení spustitelného kódu pro funkci knihovny DLL při běhu programu. Knihovna DLL šetří paměť, zabírá méně místa na disku a podporuje vícejazyčné programy.

2.6 Prvky jazyka

2.6.1 Třída

Třída umožní uživateli vytvořit vlastní typ. Jde o spojení proměnných jiného typu, metod a událostí. Třídou je definováno chování typu a její data. Třída slouží k vytvoření objektů nebo instancí (v případě že se nejedná o statickou třídu), které jsou přiřazeny k proměnným. Objekt nebo instance zůstane v paměti, dokud na ni existuje odkaz. V případě, že již neexistuje odkaz, CLR povolí uvolnění paměti. Jestliže je třída deklarována jako statická, v paměti existuje jen jedna.

2.6.2 Pole

Pole je množina proměnných stejného datového typu, tato množina proměnných tvoří celek. K jednotlivým prvkům polí (jednotlivým proměnným v poli) se přistupuje pomocí indexů pole a následně s nimi můžeme pracovat jako s jednoduchými proměnnými. První prvek pole má index 0, délka pole se značí N, poslední prvek pole se tedy nachází na indexu N-1. V jazyce C# existují pole jednorozměrné a vícerozměrné. Operace určené pro práce s polem se nachází ve třídě `System.Array`.

Deklarace jednorozměrných polí se může provést dvěma způsoby:

- `typ[] nazev_pole = new typ [velikost];`
- `Array nazev_pole = Array.CreateInstance(typeof(typ), velikost);`

Deklarace vícerozměrných polí lze provést následujícím způsobem:

- `typ[,] nazev_pole = new typ[velikost_x, velikost_y];`

2.6.3 Rozhraní

Jazyk C# umožňuje pouze jednoduchou dědičnost. Pomocí rozhraní můžeme třídám přidat určité charakteristiky, které nebudou závislé na hierarchii tříd. Rozhraní definujeme klíčovým slovem *interface*, po kterém následuje jméno rozhraní. Za definicí ve složených závorkách je poté napsáno tělo rozhraní, ve kterém můžou být výčty vlastností, událostí a metod (hlavička metody bez těla metody). Všechny členy rozhraní jsou automaticky veřejné (`public`). C# umožňuje třídám implementovat více rozhraní. Za pomoci dědičnosti můžeme rozhraní rozšířit a kombinovat.

2.6.4 Operátory

Jazyk C# obsahuje množinu operátorů téměř stejnou jako v jiných programovacích jazycích, např. Java. V následující tabulce 1 je uvedena množina operátorů obsahujících jazyk C#. Operátory jsou seřazeny podle jejich priority.

Tabulka 2 – Operátory

Primární	new, typeof, sizeof, checked, unchecked, (x) x.y f(x) a[x] x++ x--
Unární	+ - ! ~ ++x, --x (T)x
Multiplikativní	* / %
Aditivní	+, -
Bitové posuny	<< >>
Relační	< > <= >= is as
Rovnost	== !=
Logické AND (bitové)	&
Logické XOR (bitové)	^
Logické OR (bitové)	
AND	&&
OR	
Podmíněný výraz	? :
Přiřazení	= *= /= %= += -= <<= >>= &= ^= =

2.6.5 Výjimky

Výjimka je objekt, který obsahuje informaci o problému. Je odvozena ze třídy *Object*, z této třídy odvozuje základní třídu výjimek *Exception*. Z této třídy dědí ostatní třídy výjimek. Výjimku může vyvolat prostředí .NET Framework, nebo můžeme výjimku odchytit pomocí příkazu *throw*. Výjimka se objevuje v případě, že program objeví neočekávaný běh programu, např. dělení nulou, nebo že se snažíme zapsat prvek na pole mimo hranice pole.

Následující kód ukazuje strukturu bloku *try* pro odchyťování výjimek.

```
try
{
    //Příkazy pro kontrolu
}
catch (System.NullPointerException e)
{
    //Odchycení výjimky NullPointerException, a reakce na ni
}
catch (System.Exception e)
{
    //Odchycení výjimky Exception, a reakce na ni
}
```


3 NDatabase

3.1 Objektová databáze

Databázový objekt je databázový systém, v němž jsou informace reprezentovány v podobě objektů, jako jsou používány v objektově orientovaném programování. Objektové databáze umožňují vyššímu programovacímu jazyku uložit data jako objekty, a nástroje pro správu, úpravu existujících objektů.

3.2 Základní informace

NDatabase je novou generací databází, pro objektově orientované programování. Jedná se o reálně transparentní stálou vrstvu, která umožňuje komukoliv uchovat původní objekty .NET s jediným řádkem kódu. NDatabase může být použita jako vložená objektová databáze, integrována do jakékoliv aplikace využívající .NET Framework.

Využívá jednoduché rozhraní API a nevyžaduje žádné mapovací techniky, jako třeba mapování mezi původními současnými objekty. NDatabase ukládá veškeré objekty tak jak jsou, bez jediného zásahu, nebo jeho úpravy. NDatabase nevyžaduje žádnou správu ani instalaci.

NDatabase je šířena pod licencí LGPL, což je licence svobodného softwaru, publikovaná nadací Free Software Foundation.

3.3 Princip

NDatabase je distribuován jako jediný soubor DLL a její běh zabere méně než 400 kB operační paměti. Databázi lze snadno zabalit do jakéhokoli typu .NET. Podporuje ACID transakce pro zajištění integrity dat z databáze. Všechny příkazy budou provedeny v databázi i v případě selhání hardwaru, a to díky automatické obnově přenosu při dalším spuštění. Pro modifikaci dat pomocí API lze využít tyto typy modifikace:

- přidání nového pole,
- odebrání pole,
- přejmenování třídy,
- přejmenování pole.

NDatabase používá jen jeden soubor pro uložení všech dat (meta-model, indexy, objekty). Pracuje na poli s názvem indexy, založené na binárních stromech, které jsou zpřístupněny API.

V současné době podporuje 4 druhy triggerů:

- trigger výběrový,
- trigger vložení,

- trigger odstranění,
- trigger aktualizací.

3.4 Produktivita

NDatabase umožňuje uchovat data s velmi málo řádky kódu. Není potřeba upravovat třídy, které musí být stálé a není potřeba žádné mapování. Nejnovější verze NDatabase podporuje použití LINQ jako dotazovacího jazyka a zobrazení dat s nejnovější verzí LinqPadu. Linq je integrovaný jazyk pro dotazování, představený s jazyky C# 3.0 a Visual Basic 9 spolu s .NET Frameworkem 3.5. LINQ přináší nové možnosti pro dotazy nad jakýmkoliv daty, usnadňuje tvorbu, třídění, propojování a vyhledávání dat.

3.5 Použití

V první ukázce práce s databází je zobrazeno, jak lze vložit nový objekt do databáze. V kódu je použit implicitní datový typ pomocí slova *var*, které umožňuje od verze C# 3.0 deklarovat proměnné v rozsahu metody, aniž bychom zadali jeho typ. Typ objektu určuje kompilátor. Následující dvě deklarace proměnné jsou ekvivalentní:

- `var i = 1;`
- `int i = 1;`

V ukázce níže se provede vytvoření nové instance, následně je otevřen přístup do databáze a objekt je do ní vložen.

```
var objekt = new TypObjektu("nazev_objektu");
using (var odb = OdbFactory.Open(Jmeno_databaze))
    odb.Store(objekt);
```

Objekty z databáze lze dostat pomocí API rozhraní S.O.D.A., což je nástroj pro jednoduchou komunikaci s objektovými databázemi. Na začátku je třeba otevřít databázi, ze které chceme objekt získat. V dalším kroku pomocí příkazu `odb.Query<Osoba>()` získáme z databáze všechny objekty typu „Osoba“. Pomocí příkazu `query.Descend("Jmeno").Constrain("Marie").Equal()` vytřídíme ze seznamu jen ty objekty, jejichž atribut „Jmeno“ obsahuje jméno Marie. V posledním řádku je tento seznam osob se jménem Marie uložen do proměnné `seznam_osob`.

```
using (var odb = OdbFactory.Open(Jmeno_databaze))
{
    var query = odb.Query<Osoba>();
    query.Descend("Jmeno").Constrain("Marie").Equal();
    var seznam_osob = query.Execute<Osoba>();
}
```

4 PDFSharp

4.1 PDF

PDF je souborový formát založen na jazyce PostScript, který vyvinula firma Adobe. Jedná se o přenosný formát dokumentů (není závislý na softwaru ani hardwaru) a je určený pro ukládání dokumentů. Formát PDF zajišťuje, že se dokument zobrazuje na všech zařízeních stejně, přičemž může obsahovat texty i obrázky. Formát může obsahovat video, zvuk, nebo další aktivní typy, ty však nejsou podporovány ve všech PDF prohlížečích. PDF soubory mají příponu .pdf. PDF umožňuje také možnost uložení částí dokumentu do jediného souboru s použitím komprese, ke komprimování je použit algoritmus LZW.

4.2 Základní informace

PDFsharp je .NET knihovna pro programové vytváření a úpravy dokumentů PDF za chodu programu. Je kompletně napsán v jazyce C# a může být použit z libovolného .NET jazyka. Byl vyvinut s Visual Studiem 2012 a nová verze s Visual Studiem 2013, ale je kompatibilní se všemi C# kompilátory. Podporuje verze 1.2 – 1.4 pro čtení a zápis a verze 1.2 – 1.7 pro vytváření dokumentů. Všechny grafické třídy jsou potomky tříd jazyka .NET/C#. Podporuje PDF Form objekty jako šablony a XGraphics třídu pro vytváření PDF objektů formuláře. Grafické objekty jsou přidávány pomocí návrhových vzorů rozhraní .NET Framework. PDFSharp využívá následující sady funkcí z knihovny System.Drawing.Graphics:

- čáry, křivky, oblouky
- obdélníky, zaoblené obdélníky, elipsy, mnohoúhelníky, uzavřené drážkování, dráhy,
- RGB, CMYK, a šedé stupnice barev s průhledností,
- pera a štětce pro kreslení a plnění,
- GDI + jako předdefinované barvy, pera a štětce,
- maticové transformace,
- BMP, PNG, GIF, TIFF a JPEG obrázky,
- formulář XObjects,
- první dva typy čárových kódů (kód 2of5 Interleaved a Code3of9 Standard).

4.3 Vlastnosti

Primárním účelem knihovny je tvorba dokumentů formátu PDF za běhu programu, dále umožňuje měnit, slučovat a rozdělovat existující soubory, posílat soubory na tiskárnu, číst a zobrazovat soubory formátu PDF. PDFSharp umožňuje i vykreslování grafických prvků a obrázků s průhledností, barevnou maskou, nebo monochromatickou maskou. Knihovna podporuje šifrování dokumentů s 40/128 bitovým heslem a nastavování oprávnění.

4.4 Práce s knihovnou

Prvním krokem pro úspěšné vytvoření dokumentu PDF za pomoci PDFSharpu je vytvoření nového dokumentu a přidání nové stránky.

```
PdfDocument dokument = new PdfDocument();  
PdfPage page = document.AddPage();
```

Po vytvoření stránky je potřeba vytvořit objekt XGraphics pro vykreslování do dokumentu a font pro texty. Font v následující ukázce je nastaven na Verdana, velikost fontu na 20px a texty budou vykresleny tučně a zkoseně.

```
XGraphics gfx = XGraphics.FromPdfPage(page);  
XFont font = new XFont("Verdana", 20, XFontStyle.BoldItalic);
```

V následující části kódu je proveden zápis textu Hello World! Do dokumentu, je použit font vytvořený výše, a následně vytvořen nový obdélník pro vypsání textu, a text nastaven na střed stránky.

```
gfx.DrawString("Hello, World!", font, XBrushes.Black, new XRect(0, 0,  
page.Width, page.Height), XStringFormat.Center);
```

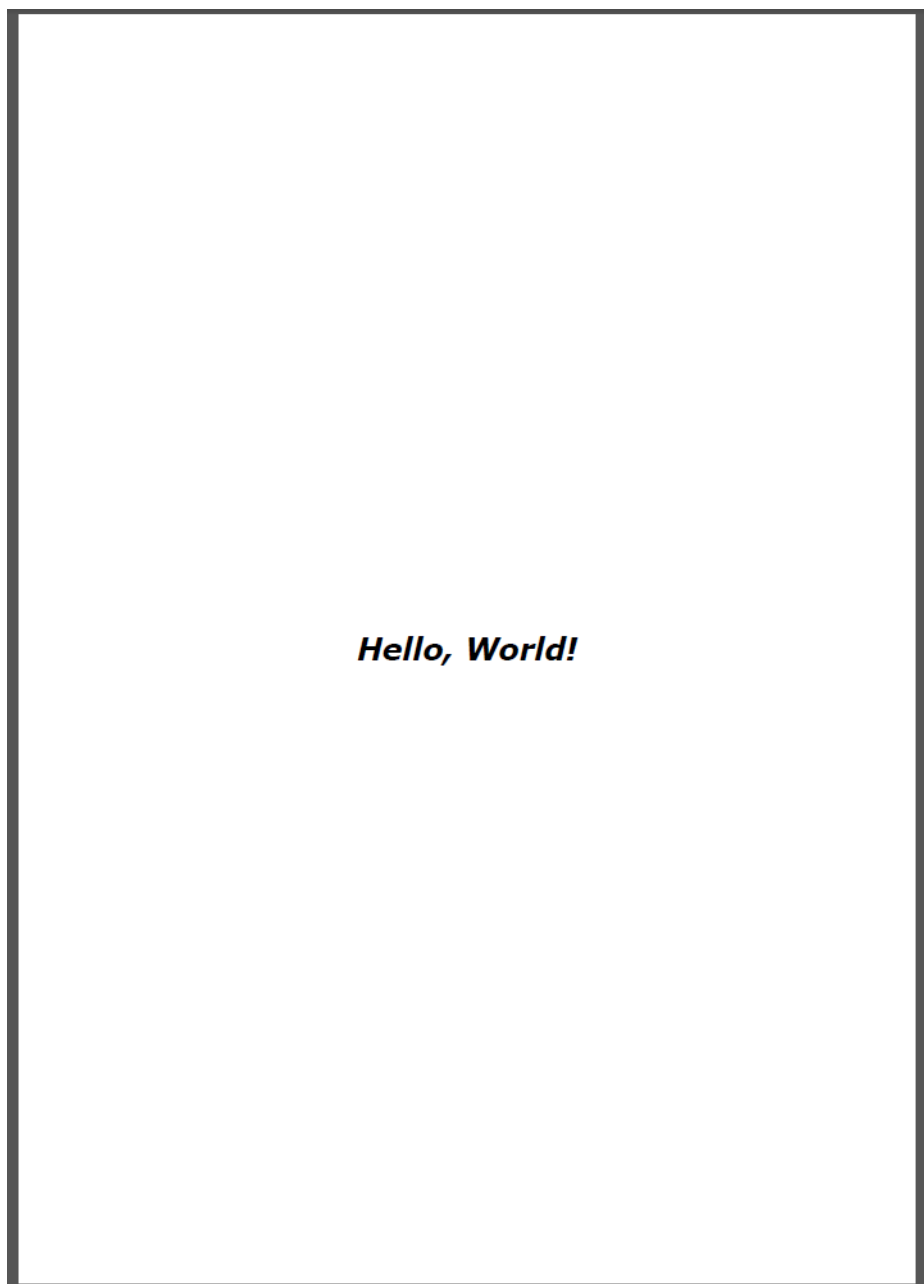
Následující kód slouží pro uložení hotového dokumentu.

```
string jmenoSouboru = "HelloWorld.pdf";  
document.Save(jmenoSouboru);
```

Jazyk C# umožňuje zobrazení hotového PDF dokumentu ve výchozí aplikaci počítače, určenou pro otevírání dokumentů.

```
Process.Start(filename);
```

Na následujícím obrázku 5 je zobrazen výstupní dokument PDF vytvořený pomocí knihovny PDFSharp kódem uvedeným výše.



Obrázek 5 – Výstup ve formátu PDF

5 Implementace aplikace

5.1 Návrh aplikace

Aplikace je implementována pomocí 17 tříd, které zpracovávají logiku programu. Data se ukládají do objektové databáze a výstup aplikace do vytvořené složky v dokumentech. Vývoj byl zaměřen na přehlednost a jednoduchost ovládání programu. Aplikace při startu programu načte z databáze konfigurační soubor, který uchovává veškeré nastavení aplikace, včetně předem vyplněných hodnot. V případě, že konfigurační soubor v databázi neexistuje, spustí se úvodní tutoriál, ve kterém se nastaví základní informace o sboru, které jsou potřebné pro zprávu o zásahu, jména hasičů, technika sboru, ochranné prostředky, ale také nastavení e-mailové adresy pomocí přihlašovacího jména, hesla, SMTP serveru a portu. Dále je při prvním spuštění vytvořena složka „ZpravyOZasahu“ do dokumentů uživatele. Po vyplnění úvodního tutoriálu se již spustí hlavní program, ze kterého se celý program ovládá. Hodnoty z úvodního tutoriálu lze upravit později, spuštěním okna tutoriálu v hlavním programu.

Na následujícím obrázku 6 je zobrazena první strana úvodního tutoriálu aplikace.

Emaily pro odeslání ZOZ	
operacni.stredisko@hasici.cz	Odstranit
velitel.zasahu@hzs.cz	Odstranit

Obrázek 6 – Úvodní tutoriál aplikace

Hlavní program obsahuje seznam zpráv o zásahu vygenerovaných aplikací, které jsou uloženy v databázi a ovládání programu. Z hlavního programu se tvoří nové zprávy, nebo upravují hodnoty databáze zapsané v úvodním tutoriálu. Zprávy v seznamu lze upravovat, mazat, zobrazit náhled a odesílat na e-mail. U každé zprávy v seznamu jsou zobrazeny nejdůležitější informace; a to datum zprávy, identifikační číslo zprávy, o jaký typ události se jednalo, na jaké adrese proběhl zásah a jestli byla zpráva po exportu odeslána e-mailem (obrázek 7).

Datum Zprávy	Zpráva	Typ Události	Adresa	Odeslána	Náhled	Upravit	Odstranit
12.03.2015	120315_001	Požár	Palackého 280, Pardubice	<input type="checkbox"/>	Náhled	Upravit	Odstranit
26.03.2015	260315_001	Dopravní nehoda letecká	Jungmannova 25, Sezemice	<input type="checkbox"/>	Náhled	Upravit	Odstranit

Obrázek 7 – Hlavní program

Vytvoření nové zprávy se provede pomocí tlačítka „Vytvořit zprávu“. Otevře se formulář pro vytváření zprávy, ve kterém jsou již připraveny předem definované seznamy hasičů, automobilů, atd. Jsou také vyplněny kolonky Název JPO a Evidenční číslo JPO z konfiguračního souboru, které byly nastaveny v úvodním tutoriálu (obrázek 8). Formulář se ovládá pomocí záložek v horní liště (*tabs*). Po vyplnění všech informací o zásahu se zpráva dá zobrazit pomocí tlačítka „Náhled zprávy“. Po kliknutí na tlačítko se vytvoří PDF ve složce určené pro dočasné zprávy, která se následně zobrazí a odstraní. Po vytvoření zprávy se zpráva uloží do databáze, exportuje do PDF a otevře se formulář s nabídkou odeslání na e-mail.

Obrázek 8 – Formulář Nové zprávy o zásahu

Při vytváření nové zprávy u zásahu se musí uvést veškeré automobily, a u každého z nich datum a čas kdy byly povolány k výjezdu, kdy na výjezd dorazily a kdy se vrátily na základnu. Na následujícím obrázku 9 je vyobrazen vyplněný řádek pro automobil.

Účast	Automobil	Motohodiny	Ujeté Km	Povolána	Příjezd	Návrat
<input checked="" type="checkbox"/>	Tatra	55	27	12-04-15 12:00	12-04-15 13:00	12-04-15 17:00

Obrázek 9 – Pole pro automobil

Pro ulehčení zadávání dat a časů se při kliknutí na libovolné pole pro zadávání data a času otevře nové okno, ve kterém se vyplní všechny informace najednou. V následujícím obrázku je znázorněno okno pro vyplnění dat a časů (obrázek 10). Data a časy, které se zobrazí ve formuláři, jsou odvozena od výjezdu jednotky, příjezdu k zásahu a návratu na základnu. Tyto časy se mění i v případě, že datum bylo změněno, změní se však jen ta data a časy, které nebyly již dříve upraveny na jiné hodnoty.

Obrázek 10 – Doplnění dat a časů

Okno pro možnost odeslání elektronickou poštou obsahuje seznam e-mailů, které jsou nastaveny v konfiguračním souboru. Po kliknutí na tlačítko „Odeslat na e-mail“ se program pokusí přihlásit do e-mailové schránky pomocí předem definovaných informací nastavených v tutoriálu, v případě úspěšného přihlášení program odešle zprávy na všechny označené e-maily, do kterých připojí zprávu o zásahu ve formátu PDF jako přílohu (obrázek 11).

Obrázek 11 – Odeslání na e-mail

5.2 Třídy aplikace

Program obsahuje 17 základních tříd. Celý program se ovládá přes třídu *HlavniProgram*, uživatel ovládá program pomocí tlačítek (*buttons*).

5.2.1 Objekty zprávy

Soubor *ObjektyZpravy.cs* obsahuje následující seznam tříd. Tyto třídy jsou potřebné pro vytváření nové zprávy a pro přednastavené hodnoty v nastavení aplikace. Každá třída také obsahuje gettery a settery pro kontrolu nad úpravami atributů jednotlivých tříd zvenčí.

- *Auto* – obsahuje základní informace o automobilech.
- *Datумы* – obsahuje všechna data a časy, které vyžaduje zprávy o zásahu.
- *Hasic* – jméno, příjmení a hodnost hasiče.
- *HasiciUZasahu* – obsahuje seznam hasičů, kteří se zásahu účastnili.
- *Hasiva* – informace o hasivech a sorbentech, které sbor používá.
- *ZakladniInfo* – obsahuje základní informace o sboru a o zásahu.
- *OchranneProstredky* – obsahuje základní informace o ochranných prostředcích sboru.
- *Ostatni* – třída obsahující informace o spolupráci s ostatními složkami IZS, zranění ostatních osob a využitou osobní a věcnou pomoc.
- *PomocnaTechnika* – pomocná technika sboru.
- *Technika* – obsahuje seznam pomocné techniky, automobilů, použitých hasiv a sorbentů.

5.2.2 Třída HlavníProgram

Hlavní páteří třída programu, přes kterou se lze dostat ke všem částem programu. Třída obsahuje bezparametrický konstruktor, který kromě načítání komponent provádí kontrolu prvního spuštění a načítá seznam zpráv.

- `private void kontrolaPrvnihoSpusteni()` – Metoda provádějící kontrolu při každém startu programu. V první fázi otevře databázi a zjistí, zda v ní existuje konfigurační soubor. V případě že neexistuje, je vytvořen, poté zjistí, jestli existuje složka v dokumentech, kam se ukládají zprávy o zásahu. V případě, že neexistuje, je složka vytvořena a je spuštěn úvodní tutoriál. Pokud konfigurační soubor existuje, je spuštěn hlavní program.
- `private void NactiZpravy()` – Funkce se volá při načítání zpráv do seznamu v hlavním programu. Metoda otevře databázi a všechny zprávy vloží do rozhraní `IList` a následně seřadí seznam podle data a názvu zprávy. Zprávy jsou poté načteny do `DataGridView` v hlavním programu.
- `private void btn_upravit_click(object sender, EventArgs e)` – Otevře nastavení předdefinovaných hodnot a mailů pro úpravu.
- `private void btn_konec_click(object sender, EventArgs e)` – Ukončí program.
- `Private void dataGridView_seznamZprav_click(object sender, dataGridViewCellEventArgs e)` – Kontroluje kliknutí na `DataGridView`.

Hlavní program umožňuje upravit existující zprávy, odstranit je ze seznamu, zobrazit náhled ve formátu PDF a odeslání elektronickou poštou.

5.2.3 Třída Konfigurator

Obsahuje základní nastavení programu uložené v databázi, které je načteno při spuštění programu. Obsahuje přednastavené hodnoty pro novou zprávu a nastavení údajů e-mailu. Konfigurator obsahuje bezparametrický konstruktor pro případ, že se uživatel rozhodne při prvním spuštění nevyplňovat informace o sboru a konstruktor obsahující seznam informací o sboru a nastavení elektronické pošty. Třída obsahuje gettery a settery pro všechny atributy třídy.

5.2.4 Třída NovaZprava

Třída NovaZprava je určena pro vytváření nové zprávy o zásahu, ale i pro úpravu již existujících zpráv. Obsahuje bezparametrický konstruktor, který načte komponenty, načte konfigurační soubor a nastaví data a časy nové zprávy. Konstruktor s jedním parametrem určeným pro zprávu se volá v případě, že se jedná o úpravu zprávy. Od bezparametrického konstruktoru se liší tím, že časy a data zprávy nastaví podle zprávy předané v parametru a načtení existující zprávy do formuláře.

- `private void zmenaDatum(DateTime datum)` – Nastaví data a časy ve formuláři, na datum předané v parametru zprávy.
- `public static string Show_Text_Dialog(string text, string caption, string puvodni)` – Metoda pro otevření dialogu, do kterého se zadá textový řetězec, který je následně vrácen jako návratová hodnota metody.
- `private bool kontrolaNepovinne(List<Hasic> hasici, List<Auto> technika, ZakladniInfo zakladniI)` – Kontrola nepovinných údajů zprávy o zásahu, metoda provede kontrolu, jestli zpráva obsahuje všechny informace o sboru a zásahu. Jestliže zpráva neobsahuje všechny potřebné informace, je uživatel upozorněn na tuto skutečnost pomocí dialogu. Pokud bude uživatel pokračovat, metoda vrátí `true` a program bude pokračovat dál.
- `private bool kontrola()` – Na rozdíl od předchozí kontroly, je tato kontrola povinná a nelze ji ignorovat. Metoda provede kontrolu, zda je zadáno evidenční číslo události a je vybrán typ události. Pokud jedna z hodnot není zadána, metoda vrátí `false`, uživatel je upozorněn dialogem proč nelze pokračovat a odkázán na místo kde je potřeba zadat hodnoty. V další fázi metoda kontroluje zadané číselné formáty; v případě, že uživatel zadá do pole, ve kterém je očekáváno číslo text, je uživatel upozorněn a vyzván k opravě. Program pokračuje až do doby, kdy jsou zadány povinné údaje a pole očekávající čísla jsou prázdná, nebo obsahují číselné hodnoty.

- `private void comboBox_typUdalosti_SelectedIndexChanged(object sender, EventArgs e)` – V ojedinělých případech může nastat, že typ události nebude uveden v seznamu událostí. Pomocí této metody, lze změnit *ComboBox* na *Textbox*, a typ události do něj vepsat jako textový řetězec.
- `private void btn_vytvor_click(object sender, EventArgs e)` – Zavolá metodu pro vytvoření nové zprávy a následně zjistí, zda se jedná o úpravu zprávy, nebo její vytvoření. V případě vytvoření nové zprávy o zásahu volá metody pro export do formátu PDF a odeslání zprávy elektronickou poštou. Zpráva je následně uložena do databáze. V případě úpravy, je zpráva exportována do PDF, otevřen dialog pro odeslání zprávy elektronickou poštou a podle názvu zpráva načtena z databáze a následně nahrazena zprávou novou. Po uložení zprávy do databáze je formulář pro vytváření nové zprávy o zásahu ukončen.
- `private Zprava vytvorZpravu()` – Nejprve provede kontrolu povinných hodnot zavoláním metody `kontrola()`, která je popsána výše. Vytvoří proměnné a seznamy potřebné pro zprávu o zásahu, následně do nich načte hodnoty z formuláře pro vytváření nové zprávy o zásahu, zavolá metodu pro kontrolu nepovinných údajů, vytvoří jedinečný identifikátor zprávy o zásahu a vrátí nově vytvořenou instanci zprávy o zásahu.
- `private void nactiKonfig()` – Metoda se volá v konstruktoru třídy, načte z databáze konfigurační soubor a do formuláře pro vytvoření nové zprávy o zásahu z něj vloží název jednotky požární ochrany, evidenční číslo jednotky, seznamy hasičů, automobilů, ochranných prostředků, hasiv a sorbentů a pomocné techniky.
- `private void exportPDF(Zprava zp, bool nahled)` – Provádí generování zprávy předané v parametru do formátu PDF. Podle druhého parametru metoda pozná, zda se jedná jen o náhled, nebo o vytvoření zprávy. V případě, že se jedná jen o náhled, je zpráva vygenerována do složky pro dočasné soubory a následně smazána. V případě vytvoření se zpráva vloží do složky určené pro zprávy o zásahu. Pokud se jedná o generování zprávy, která byla upravována, je zpráva přepsána novou zprávou.
- `private void nactiExistujici(Zprava zp)` – Metoda načte již existující zprávu o zásahu předanou parametrem do formuláře pro vytváření zpráv o zásahu, zaškrtně a vyplní všechny pole tak, jako byla vyplněna v původní zprávě.

5.2.5 Třída OdeslaniNaMail

Třída má za úkol přihlásit se do e-mailové adresy uživatele pomocí adresy a hesla zadaného v nastavení aplikace a následně odeslat zprávu o zásahu elektronickou poštou, na všechny uživatelem vybrané e-mailové adresy. V příloze každého e-mailu je připojena

zpráva o zásahu ve formátu PDF. Třída obsahuje konstruktor s parametrem zprávy o zásahu, která má být odeslána.

- `public bool nactiEmaily()` – Proveďte načtení konfiguračního souboru z databáze a přidejte všechny e-mailové adresy do seznamu. Jestliže konfigurační soubor neobsahuje žádnou předem zadanou e-mailovou adresu, zobrazí se hláška, že nebyla přidána žádná adresa, odesílání elektronickou poštou se přeskočí a program se vrátí do hlavního okna.
- `public void posliMail()` – Za pomoci údajů z konfiguračního souboru o nastavení e-mailové schránky se pokusí přihlásit uživatele a odeslat zprávu o zásahu na všechny vybrané e-mailové adresy. V případě, že se nepodaří odeslat na všechny adresy, je vypsáno chybové hlášení se seznamem e-mailů, na které se zprávu nepodařilo odeslat.

5.2.6 Třída Tutorial

Spouští se při prvním startu programu a v případě každé uživatelské úpravy nastavení. Je určen pro vyplnění informací o sboru dobrovolných hasičů, tyto informace zahrnují seznamy hasičů v zásahové jednotce, automobily, které má sbor k dispozici, ochranné prostředky, hasiva, sorbenty a pomocnou techniku. Dále jsou zde nastaveny všechny údaje potřebné pro přihlášení do elektronické pošty a seznam příjemců pro zprávy o zásahu. Obsahuje bezparametrický konstruktor, který zavolá metodu `nactiHodnoty()` popsanou níže a inicializuje komponenty.

- `private bool kontrola()` – Provádí kontrolu, zda jsou všechny číselné údaje a e-mailové adresy zadané uživatelem ve správném formátu.
- `private void nactiHodnoty()` – Načte konfigurační soubor z databáze a do formuláře vyplní všechny údaje, které obsahuje nastavení programu.
- `private void btn_ulozit_Click(object sender, EventArgs e)` – Metoda volaná po kliknutí na *button* uložit nastavení, provede kontrolu, zda jsou údaje ve správném formátu zavoláním metody `kontrola()`, viz. výše. Pokud kontrola neproběhne v pořádku, zobrazí uživateli chybovou hlášku s popisem, z jakého důvodu nebyla kontrola úspěšná. Pokud kontrola proběhne úspěšně, je načten z databáze konfigurační soubor, který je následně upraven hodnotami z formuláře.

5.2.7 Třída Zprava

Obsahuje veškeré informace o zprávě o zásahu a obsahuje všechny metody `get` a `set`, pomocí kterých lze zprávu upravovat zvenčí.

5.3 Práce s databází

V programu je použita objektová databáze *NDatabase*. Při spuštění se z databáze načte konfigurační soubor. V případě, že konfigurační soubor neexistuje, je vytvořen a je

spuštěn úvodní tutoriál pro nastavení parametrů. Jestliže se nejedná o první spuštění, načte z databáze seznam zpráv, který se uloží do *IListu* (kolekce objektů) a následně je seřazen dle data vypsání. V následujícím kódu je zobrazené načtení všech zpráv z databáze do *IListu*.

```
IList<Zprava> seznamZprav =  
odb.Query<Zprava>().Execute<Zprava>().ToList();
```

Při odstranění zprávy ze seznamu zpráv nebo jejím upravení je z databáze načtena zpráva dle názvu. Při úpravě je následně aktualizována, při odstranění se zpráva smaže z databáze. V ukázce kódu je zobrazeno načtení zprávy dle jména zprávy.

```
using (var odb = OdbFactory.Open(dbname))  
{  
    zprava = (from zpr in odb.AsQueryable<Zprava>()  
              where (zpr.NazevZpravy) == nazevZpravy  
              orderby zpr.NazevZpravy  
              select zpr).Last();  
}
```

5.4 Export do PDF

Pro export do formátu PDF je použita knihovna *PDFSharp*. V ukázce kódu je vytvořen nový dokument formátu PDF, následně je do něj vložena nová prázdná stránka, získání objektu pro vykreslení, nastavení písma a následně vytvořeny dva fonty; jeden určený pro prostý text a druhý pro nadpisy.

```
PdfDocument document = new PdfDocument();  
PdfPage page = document.AddPage();  
XGraphics gfx = XGraphics.FromPdfPage(page);  
XTextFormatter tf = new XTextFormatter(gfx);  
XPdfFontOptions options = new XPdfFontOptions(PdfFontEncoding.Unicode,  
PdfFontEmbedding.Always);  
XFont font = new XFont("Arial", 16, XFontStyle.Underline);  
XFont fontNadpis = new XFont("Arial", 11, XFontStyle.Bold);
```

Při vytváření dokumentu je vytvořeno textové pole, do kterého je následně vložen text, nastavený font, barva fontu a vytvořen nový obdélník v závorce je nejprve počáteční bod, šířka a výška a zarovnání, v tomto případě vlevo nahoru.

```
gfx.DrawString(zp.ZakladniI.TypZpravy, font, XBrushes.Black, new  
XRect(20, 20, page.Width, page.Height), XStringFormats.TopLeft);
```

Pro vykreslení čáry je použit následující kód, který vykreslí čáru, nejprve se nastaví nové pero pro vykreslení černé barvy, a následně počáteční a koncový bod.

```
fx.DrawLine(new XPen(XColor.FromArgb(000000)), new XPoint(20, 60), new  
XPoint(page.Width - 20, 60));
```

Na následujícím obrázku 12 je vidět část výstupu pdf.

Dílčí zpráva o zásahu

ECUD: 2113000001
Datum: 2.6.2013

Technická pomoc

Složka IZS: **HZS ČR a jednotky PO**
Jednotka PO: **Sbor dobrovolných hasičů Uhliřské Janovice 215142**
Velitel: **Petr Barták**

Adresa: **Fanta Eduard Křečovice 29**
Objekt: **Rodinný dům**
Prostor: **Zahrada**
Specifikace zásahu: **čerpání vody**
Důvod nepřítomnosti:

Den ohlášení: **2.června 2013**
neděle
Čas ohlášení: **09:14**

Ohlášení: **2.6 2013 09:14**
Vyhlášení poplachu: **2.6 2013 09:14**
Výjezd: **2.6 2013 09:23**
Příjezd k zásahu: **2.6 2013 09:35**
Zahájení zásahu: **2.6 2013 09:35**
Odjezd na základnu: **2.6 2013 21:50**
Příjezd na základnu: **2.6 2013 22:00**

Vzdálenost [km]: **8**
Počet zásahů: **1**
Nasazení spec. skupin: **1**
Pořadí jednotky PO: **14**
Počet osob: **5**
Počet techniky: **5**

Činnosti jednotky PO

Průzkum
Evakuace osob objektová
Čerpání a odčerpávání vody
Jiné

Zasahující technika jednotky PO

Hlavní technika	Typ	Povolána	Příjezd k zásahu	Odjezd na zákl.	Poznámka
DA Citroen Jumper 2,5 TDui	dopravní automobil 8-xxx	3.6 2013 09:38	3.6 2013 09:58	3.6 2013 15:28	
Škoda pickup	technický automobil-xx	3.6 2013 10:15	3.6 2013 10:25	3.6 2013 21:50	
Tatra 148	cisternová automobilová stříkačka 32/x/x-xxx	3.6 2013 09:14	3.6 2013 09:35	3.6 2013 13:20	
VW Transporter 2,5 Syncro	rychlý zásahový automobil-xxx	3.6 2013 13:52	3.6 2013 14:15	3.6 2013 14:30	
Škoda felicia kombi	technický automobil-xx	3.6 2013 20:36	3.6 2013 20:50	3.6 2013 21:50	
Pomocná technika	Typ	Poznámka			
Kalové čerpadlo HONDA 800	kalová čerpadla se spalovacím motorem				
plovoucí čerpadlo KRAKEN	plovoucí motorová stříkačka 8				
motorová stříkačka PS-12	přenosná motorová stříkačka 8				
elektrické kalové čerpadlo	kalová čerpadla s elektrickým motorem				

Obrázek 12 – Výstup ve formátu PDF

5.5 Možnost úpravy zprávy

Pokud zpráva z nějakého důvodu není dokončena, nebo vytvořená zpráva obsahuje chybu, je možné zpětně zprávu otevřít ve formuláři pro novou zprávu, do kterého se načtou data rozpracované zprávy z databáze, případně zpráva, která obsahuje chybu. Zpráva se z databáze načte dle názvu zprávy, veškeré informace, které původní zpráva obsahovala, jsou vyplněny do formuláře Nová zpráva (viz obrázek 5) a po následném uložení zprávy se zpráva v databázi aktualizuje.

V případě, že zpráva již byla exportována do formátu PDF, zpráva se ze složky odstraní a je exportována znovu s aktualizovanými informacemi.

5.6 Zabezpečení

5.6.1 Správné formáty

Program je zabezpečen proti zadání špatných formátů čísel např. u portu, nebo čísla jednotky a e-mailových adres. V případě, že správné formáty nejsou dodrženy, program nedovolí uživateli pokračovat.

5.6.2 Povinné položky

Zprávu nelze dokončit bez zadání údajů, které musí být uvedeny v každé zprávě o zásahu. Pokud položky nejsou vyplněny, zobrazí program chybové hlášení s informacemi, které položky nebyly vyplněny a odkáže na toto místo v programu.

5.6.3 Zabezpečení databáze

Do databáze nejde zasahovat jinak než přes okna programu, pokud však dojde k nečekanému zásahu do databáze, její úpravě, nebo výskytu chyby, je vymazána celá databáze včetně konfiguračního souboru, čímž je program resetován do původního nastavení a je znovu spuštěn úvodní tutoriál.

5.6.4 Smazání zprávy

V případě, že byla zpráva o zásahu exportovaná do PDF odstraněna ze složky určené pro zprávy. Po zjištění v běhu programu je zpráva odstraněna z databáze a ze seznamu zpráv v hlavním programu.

5.6.5 Odesílání e-mailu

Odesíláním elektronickou poštou jsou odchyťávány výjimky. V případě, že se nepodařilo zprávu odeslat na nějakou z vybraných e-mailových adres, seznam adres je vypsán v chybovém hlášení do vyskakovacího okna.

5.6.6 Odstranění souborů

V případě zasažení uživatele smazáním, nebo úpravou potřebných souborů a složek je při každém startu programu provedena kontrola. Tato kontrola primárně slouží k tomu, aby aplikace zjistila, jestli již byl program dříve spuštěn na počítači. V první fázi zkontroluje funkce, zda existuje složka určená pro dočasné zprávy (tyto zprávy se tvoří při náhledu před exportem zprávy). Pokud složka existuje a obsahuje nějaké soubory, jsou všechny tyto soubory odstraněny. V další fázi program otevře databázi a zjistí, jestli v ní existuje konfigurační soubor. V případě, že konfigurační soubor neexistuje, je vytvořen a vložen do databáze. Pokud neexistují složky určené pro zprávy o zásahu, jsou tyto složky vytvořeny. Program následně spustí úvodní tutoriál popsáný v kapitole výše (5.1). Pokud existuje cesta do složky pro zprávy o zásahu, ale nebyl nalezen konfigurační soubor, je smazána databáze a znovu spuštěna kontrola prvního spuštění, při kterém je znovu spuštěno úvodní nastavení programu.


```

private void kontrolaPrvnihoSpusteni()
{
    bool prvniSpusteni = false;
    if (Directory.Exists(cestaDocasne))
    {
        string[] filePaths = Directory.GetFiles(cestaDocasne);
        foreach (string filePath in filePaths)
            File.Delete(filePath);
    }
    using (var odb = OdbFactory.Open(dbname))
    {
        IList<Konfigurator> konfigurator =
            db.Query<Konfigurator>().Execute<Konfigurator>().ToList();
        if (konfigurator.Count == 0)
        {
            prvniSpusteni = true;
            Konfigurator konf = new Konfigurator();
            odb.Store(konf);
            if (!Directory.Exists(cestaZOZ))
                System.IO.Directory.CreateDirectory(cestaZOZ);
        }
    }
    if (prvniSpusteni)
    {
        Tutorial uvodniTut = new Tutorial();
        uvodniTut.ShowDialog();
    };
}

```

5.7 Uživatelská přívěťivost

Hlavním cílem programu bylo co nejvíce zjednodušit proces psaní zpráv o zásahu, jejich ukládání a odesílání na operační střediska. Uživatel si proto na začátku v úvodním tutoriálu (viz. obrázek 2) zadá veškeré důležité informace o sboru, které se musí uvádět ve zprávách o zásahu do příslušných kolonek *textboxů*, tyto informace může kdykoli upravit, odstranit, nebo přidat kliknutím na tlačítko upravit informace *button* v hlavním programu (viz. obrázek 3). Veškeré informace, které uživatel zadá, jsou uloženy v konfiguračním souboru, který je uložen od prvního spuštění programu v databázi.

Při vytváření nové zprávy jsou informace o názvu sboru a evidenčním čísle JPO vypsaný jako text *label* na úvodní stránce nové zprávy o zásahu, tyto informace jsou po uložení

zprávy uvedeny v nové zprávě o zásahu. Informace o zásahové jednotce, automobilech, pomocné technice, hasivech a ochranných prostředcích jsou vypsány do bloku *datagridview* jako seznam. Před každým prvkem seznamu je zaškrťovací pole *checkbox*, v případě že uživatel toto pole zaškrtně, bude tato informace obsažena ve zprávě o zásahu. Po zaškrtnutí je nutno doplnit potřebné informace, pokud uživatel zapomene, informaci nezná, nebo se rozhodne doplnit ji později, informaci nevyplní a ta se ve výsledné zprávě projeví u textové informace jako prázdné textové pole, u číselné informace jako 0 a u data je defaultně nastavena uživatelem zadaná hodnota, kdy byl vyhlášen poplach. V případě, že uživatel nenastaví data a časy potřebné pro zprávu o zásahu, jsou všechna data nastavena na čas otevření formuláře pro novou zprávu o zásahu. Jestliže v seznamu nějaká z informací chybí, nebo se jedná o informaci týkající se pouze určité události, je uživateli povoleno přidat tuto informaci za běhu programu. Tato informace bude uvedena ve zprávě o zásahu, bude načtena při případné úpravě zprávy, ale nebude uvedena jako předdefinovaná hodnota v konfiguračním souboru, tedy při vytváření nové zprávy tato informace opět nebude uvedena v seznamu.

V úvodním tutoriálu je možné vyplnit svou e-mailovou adresu s heslem, nastavením SMTP serveru a seznamem e-mailů, na který má být zpráva o zásahu odeslána. To umožňuje uživateli odeslat zprávy z programu a nemusí se přihlašovat v prohlížeči nebo v e-mailovém správci. Zpráva je odeslána na všechny uvedené e-maily, v případě, že nastane chyba, je uživateli vypsán seznam e-mailů, na který se nepodařilo zprávu o zásahu odeslat. Zpráva se odešle s připojenou zprávou o zásahu jako přílohou. Pokud uživatel nezná, neví jak zjistit, nebo si nepamatuje SMTP server a port své e-mailové adresy a schránku má buď u poskytovatelů Google nebo Seznam, je možné vybrat ze seznamu *ComboBox* jeden z nich a port a SMTP server se následně vyplní automaticky.

Program nevyžaduje žádné pokročilé znalosti ovládání počítače, ani znalosti ohledně formátu psaní zprávy o zásahu. Je navržen tak, aby ho dokázali ovládat všichni, i méně zkušené uživatelé.

Závěr

Cílem bakalářské práce, bylo vytvořit aplikace generující zprávy o zásahu pro dobrovolné hasiče. Aplikace byla implementována pomocí jazyka .NET/C#, s pomocnou knihovnou PDFSharp, určenou pro generování dokumentů ve formátu PDF. Pro ukládání dat byla zvolena objektová databáze NDatabase. Při psaní programu bylo nutné seznámit se s požadavky dobrovolných hasičů na zprávy o zásahu a možnosti, jak nejlépe velitelům sborů ulehčit a zefektivnit psaní zpráv o zásahu a jejich odesílání na určená operační střediska. Požadavky a náležitosti zpráv o zásahu byly konzultovány s vedením SHČMS a se členy sboru dobrovolných hasičů v Uhlířských Janovicích, včetně velitele, který zprávy o zásahu vytváří.

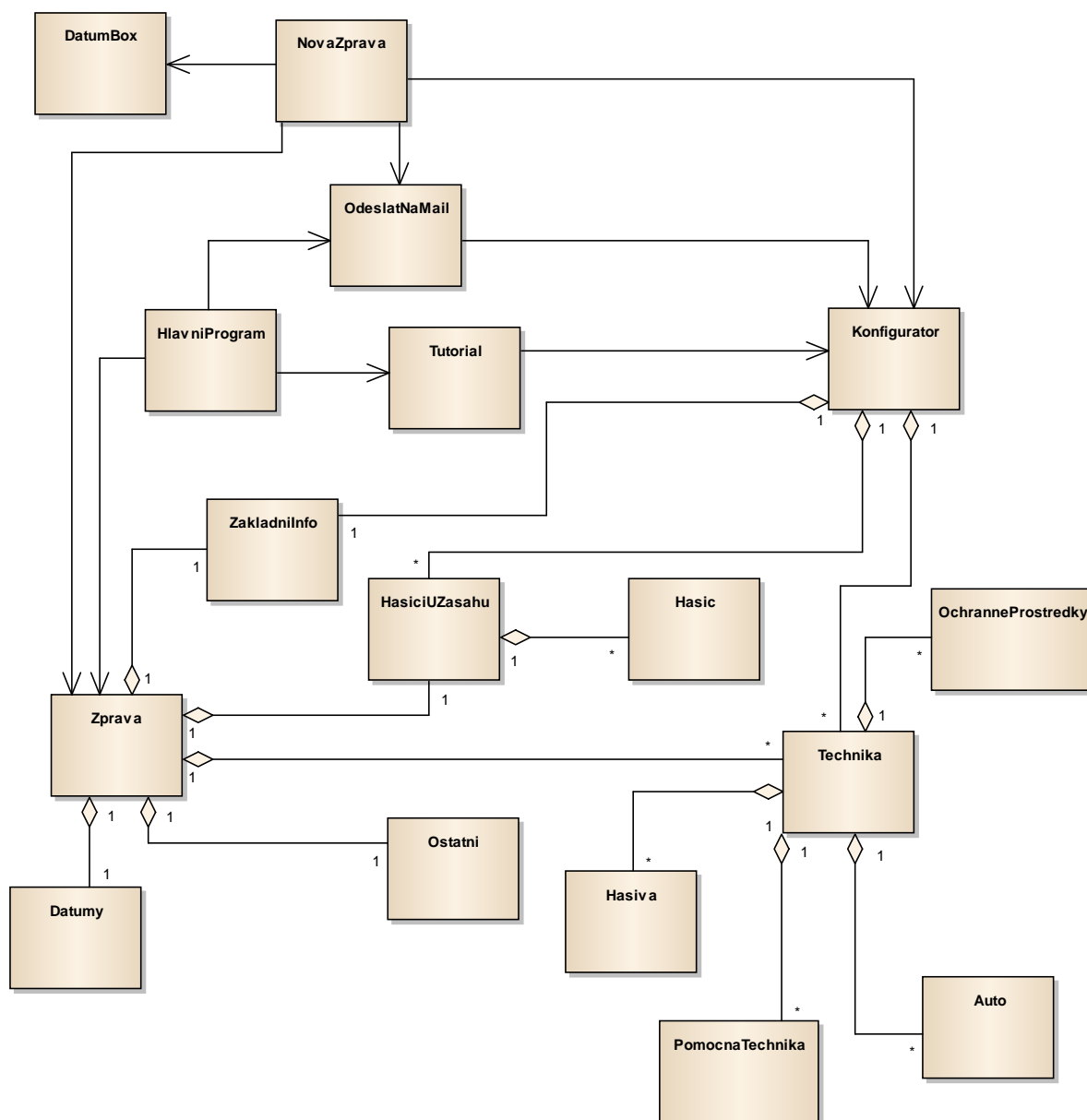
Po konzultacích a sběru informací byla vytvořena aplikace určená pro OS Windows, implementující všechny požadavky ze zadání práce, která se snaží co nejvíce vyhovět všem požadavkům a nárokům na software pro generování zpráv o zásahu. Pomocí aplikace lze vyplnit všechny informace týkající se sboru dobrovolných hasičů, které musí obsahovat zpráva o zásahu. Při psaní nové zprávy, jsou uživateli tyto informace již předem vyplněny do formuláře pro vytváření nové zprávy. Primární funkcí aplikace je vytvářet nové zprávy o zásahu tak, aby v samotném průběhu psaní zprávy bylo od uživatele vyžadováno vyplnění minimálního počtu informací, tedy těch, u kterých nebylo možné tento proces automatizovat. Všechny informace, u kterých to bylo možné, jsou zpracovány automaticky v pozadí programu a uživateli se zobrazí až v hotové zprávě o zásahu. Aplikace dále umožňuje archivovat zprávy v databázi a formátu PDF ve složce vytvořené v dokumentech, která je vytvořena aplikací při prvním startu. Dále aplikace umožňuje uživateli zpětně editovat již vytvořené zprávy, generovat zprávy ve formátu PDF a odesílat je elektronickou poštou na určené e-mailové adresy. Všechny zprávy o zásahu vygenerované aplikací se zobrazují v hlavním okně aplikace, ze kterého lze ovládat celý program. Aplikace je implementována tak, aby ji dokázal ovládat i méně zkušený uživatel PC a je zabezpečena proti zadání neplatných údajů, jako jsou špatný formát e-mailových adres, nebo zadání textu do polí, kde jsou vyžadovány číselné hodnoty.

V teoretické části byly probrány požadavky sborů dobrovolných hasičů, popsán jazyk C# spolu s knihovnou PDFSharp a objektovou databází NDatabase. V poslední části je popsána implementace aplikace a její funkčnost.

Literatura

- [1] NAGEL CH. et al. *C# 2008. Programujeme profesionálně*. Brno, 2009. ISBN 978-80-251-2407-7.
- [2] PETZOLD, Charles. *Programování Microsoft Windows v jazyce C#*. Praha : SoftPress, 2003. 1208 s. ISBN 80-86497-54-2
- [3] Standard ECMA-334: C# Language Specification. *ECMA International* [online]. 2006 [cit. 2014-09-23]. Dostupné z: <http://www.ecma-international.org/publications/standards/Ecma-334.htm>
- [4] SPÓLNIK, Jacek. NDatabase Documentation Page. *CodePlex: NDatabase, C# Lightweight Object Database* [online]. 2012. vyd. 2012-12-29 [cit. 2015-04-20]. Dostupné z: <https://ndatabase.codeplex.com/documentation>
- [5] HÖVEL, Thomas. PDFsharp and MigraDoc Wiki: MigraDoc Articles. EMPIRA SOFTWARE GMBH. *PDFSharp: PDFsharp - A .NET library for processing PDF* [online]. 2009. vyd. 2014, 2014-12-17 [cit. 2015-04-20]. Dostupné z: <http://www.pdfsharp.net/wiki/MigraDocArticles.ashx>
- [6] ADAMEC, Martin. *ZOZ formulář: Dílčí zpráva o zásahu*. 2012-003. Ostrava, 2012. Dostupné z: <http://www.hzscr.cz/soubor/formular-pro-zpracovani-zoz-verze-2012.aspx>
- [7] ADAMEC, Martin. *Návod k vypracování a použití „Zprávy o zásahu“ a „Dílčí zprávy o zásahu“*. Ostrava, 2011. Dostupné z: <http://www.hzscr.cz/soubor/zoz-navod-doc.aspx>
- [8] LOUKOTA, Roman. *Analýza motivů v kontextu činnosti dobrovolných hasičů*. Brno, 2007. Dostupné z: https://is.muni.cz/th/54029/pedf_m/Diplomova_prace.txt. Diplomová práce. Masarykova univerzita, Pedagogická fakulta. Vedoucí práce Jiří Němec.

Příloha A – UML Diagram



Příloha B – Manuál k aplikaci

Obsah

1	Úvod	45
2	První spuštění.....	45
3	Ovládání hlavního programu	46
4	Nová zpráva o zásahu	47

Seznam obrázků

Obrázek B.1 – Nastavení	45
Obrázek B.2 – Hlavní program.....	46
Obrázek B.3 – Nová zpráva.....	47
Obrázek B.4 – Data a časy	47
Obrázek B.5 – Data a časy automobilů.....	48
Obrázek B.6 – Řádek automobil.....	48
Obrázek B.7 – Odeslání na e-mail	48

1 Úvod

Cílem manuálu je naučit uživatele, jak vytvořit novou zprávu o zásahu a seznámení se základním ovládáním programu.

2 První spuštění

Při prvním spuštění programu, se uživateli zobrazí úvodní tutoriál, ve kterém vyplní všechny informace o sboru dobrovolných hasičů. Mezi okny pro vyplnění se přepíná v horní části programu pomocí čtyř *tabů*: „Základní informace“, „Technika“, „Hasiči“ a „Prostředky“. Kromě informací o sboru, se zde vyplňují i informace o e-mailové adrese a e-mailové adrese, na které uživatel požaduje odeslání zpráv o zásahu. V případě že má uživatel e-mail od poskytovatelů google.com, nebo seznam.cz, je možné pomocí *comboboxu* umístěného pod nápisem „Nastavení e-mailu“, vybrat jeden z nich, aby nemusel dohledávat informace o nastavení SMTP serveru a portu. Tutoriál se opustí jedním z tlačítek ve spodní části, po kliknutí se spustí hlavní okno programu.

The screenshot shows a window titled "Nastavení" (Settings) with a standard Windows-style title bar. Inside, there are four tabs: "Základní informace", "Technika", "Hasiči", and "Prostředky". The "Základní informace" tab is active. It contains several input fields: "Jméno JPO", "Evidenční číslo", a dropdown menu under "Nastavení e-mailu", and fields for "E-mail", "Heslo", "SMTP", and "Port". Below these is a table with two columns, the first of which is labeled "Emaily pro odeslání ZOZ". At the bottom of the window are two buttons: "Uložit a zavřít" and "Zavřít bez uložení".

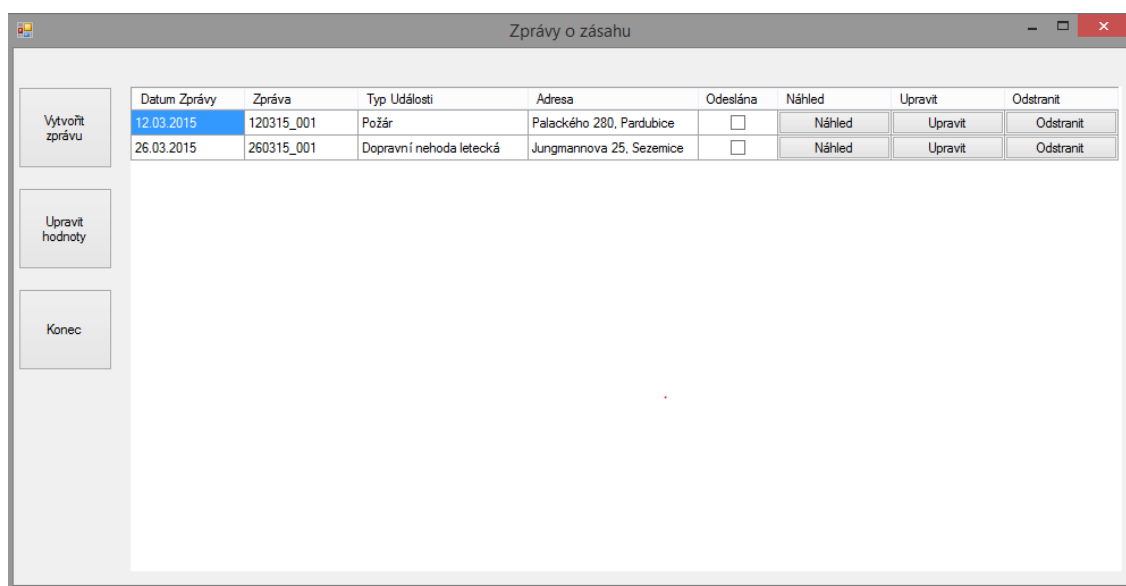
Obrázek B.13 – Nastavení

3 Ovládání hlavního programu

Pomocí hlavního okna lze ovládat celý program. Zobrazeny jsou všechny již vygenerované zprávy o zásahu s datem zprávy, jedinečným ID, typem události, adresou a informací, zda byla zpráva odeslána e-mailem. Zprávu lze ovládat pomocí ovládacích prvků na konci řádku, kde se zobrazuje zpráva:

- Sloupec „Odeslána“ – kliknutím na *checkbox* je otevřeno okno pro odeslání zprávy o zásahu na předem definované e-maily. Tento formulář je popsán níže.
- Sloupec „Náhled“ – použitím tlačítka s textem „Náhled“, vedle zprávy o zásahu, se zobrazí vygenerovaná zpráva ve formátu PDF.
- Sloupec „Upravit“ – kliknutím na tlačítko upravit, je otevřena zpráva v okně pro vytváření nových zpráv, se všemi informacemi, vyplněnými v původní zprávě o zásahu.
- Sloupec „Odstranit“ – pomocí tlačítka „Odstranit“ je zpráva na tomto řádku odstraněna z databáze a ze seznamu v hlavním okně. Pro zabezpečení proti chybnému kliknutí, musí uživatel odstranění potvrdit v otevřeném vyskakovacím okně.

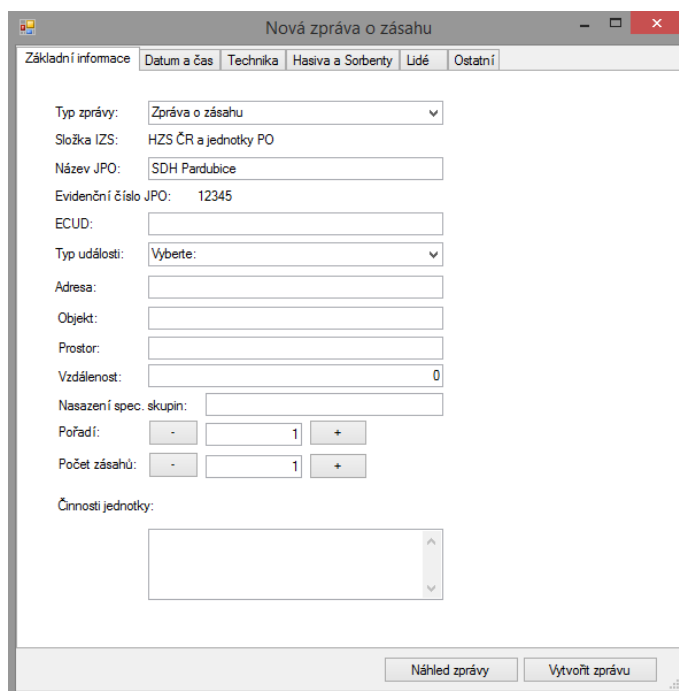
Hlavní okno slouží také jako rozcestník aplikace. Kliknutím na tlačítko „Vytvořit zprávu“ se otevře formulář pro vytváření nové zprávy o zásahu (popsáno ve 4. části), ve kterém budou vyplněny všechny hodnoty, které byly vyplněny v úvodním tutoriálu. Tyto hodnoty lze změnit kliknutím na prostřední tlačítko v levé části hlavního okna „Upravit hodnoty“. Poslední tlačítko slouží pro ukončení programu.



Obrázek B.14 – Hlavní program

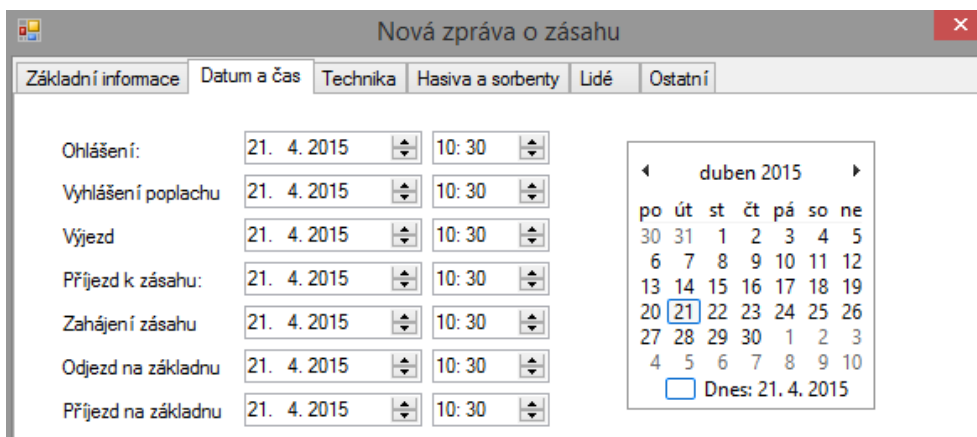
4 Nová zpráva o zásahu

Po kliknutí na tlačítko v hlavním okně „Vytvořit zprávu“ se zobrazí formulář pro vyplnění nové zprávy o zásahu. V okně jsou již vyplněny hodnoty z nastavení aplikace. Ovládání okna pro vytvoření nové zprávy, se provádí pomocí šesti *tabů* v horní části okna, pojmenovaných podle kategorií. Povinné položky pro úspěšné uložení zprávy jsou „ECUD“ a „Typ Události“. Všechny ostatní položky nemusí být vyplněny. U položek „Pořadí“ a „Počet zásahů“ lze měnit hodnotu pomocí tlačítek „+“ a „-“.



Obrázek B.15 – Nová zpráva

U *tabu* „Datum a čas“, je možnost vybrat v pravé části datum události. Kliknutím na datum se změní všechna data v levé části. Při nastavování času se všechny následující časy zvyšují s měněným časem.



Obrázek B.16 – Data a časy

Pro vyplnění dat a časů povolání, příjezdu a návratu u každého automobilu, se po kliknutí na jeden z dat v řádku zobrazí následující formulář, pro vyplnění všech dat a časů.

Obrázek B.17 – Data a časy automobilů

Po vyplnění formuláře se datum a čas zobrazí v příslušném řádku vedle automobilu. Ostatní položky se vyplňují vepsáním hodnoty do textového pole.

Účast	Automobil	Motohodiny	Ujeté Km	Povolána	Příjezd	Návrat
<input checked="" type="checkbox"/>	Tatra	55	27	12-04-15 12:00	12-04-15 13:00	12-04-15 17:00

Obrázek B.18 – Řádek automobil

Kliknutí na tlačítko pro vytvoření nové zprávy, proběhne kontrola zda jsou vyplněna všechna povinná pole, v případě že ano, je uživateli nabídnuta možnost odeslání zprávy jako přílohu pomocí elektronické pošty. Zpráva se odešle na všechny e-maily, vedle kterých je zaškrtnut *checkbox* V případě že se nepodaří zprávu odeslat zobrazí se seznam e-mailů, na které zpráva nebyla odeslána.

Obrázek B.19 – Odeslání na e-mail

Po odeslání vytvoření je zpráva zobrazena v hlavním programu.