



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA PODNIKATELSKÁ

FACULTY OF BUSINESS AND MANAGEMENT

ÚSTAV INFORMATIKY

INSTITUTE OF INFORMATICS

NÁVRH SYSTÉMU PRO EVIDENCI POŽÁRŮ

DESIGN OF SYSTEM FOR MAINTANANCE OF FIRES

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Lucia Mešková

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Aleš Klusák, Ph.D.

BRNO 2016

ZADÁNÍ DIPLOMOVÉ PRÁCE

Mešková Lucia, Bc.

Informační management (6209T015)

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách, Studijním a zkušebním řádem VUT v Brně a Směrnicí děkana pro realizaci bakalářských a magisterských studijních programů zadává diplomovou práci s názvem:

Návrh systému pro evidenci požárů

v anglickém jazyce:

Design of System for Maintanance of Fires

Pokyny pro vypracování:

Úvod

Vymezení problému a cíle práce

Teoretická východiska práce

Analýza problému a současné situace

Vlastní návrhy řešení, přínos návrhů řešení

Závěr

Seznam použité literatury

Přílohy

Seznam odborné literatury:

DATE, C. J. An Introduction to Database Systems. Vyd. 6. Reading: Addison-Wesley, 1995. 820 s. ISBN 0-201-82458-2.

KOCH, Miloš. Datové a funkční modelování. Vyd. 1. Brno: Akademické nakladatelství CERM, 2004. 108 s. ISBN 80-214-2724-8.

MATIAŠKO, Karol. Databázové systémy: základy databázových systémů. Vyd. 2. Žilina: Žilinská univerzita v Žiline, 2008. 351 s. ISBN 978-80-8070-820-7.

RIORDAN, Rebecca M. Vytváříme relační databázové aplikace. Vyd. 1. Praha: Computer Press, 2000. 280 s. ISBN 80-7226-360-9.

VRÁNA, Jakub. 1001 tipů a triků pro PHP. Vyd. 1. Brno: Computer Press, 2010. 456 s. ISBN 978-80-251-2940-1.

Vedoucí diplomové práce: Ing. Aleš Klusák, Ph.D.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2015/2016.

L.S.

doc. RNDr. Bedřich Půža, CSc.
Ředitel ústavu

doc. Ing. et Ing. Stanislav Škapa, Ph.D.
Děkan fakulty

V Brně, dne 29.2.2016

Abstrakt

Cieľom tejto diplomovej práce je navrhnúť systém pre účely evidencie požiarov Hasičského a záchranného zboru Slovenskej republiky. Systém by mal slúžiť členom zboru na spravovanie záznamov o všetkých požiaroch v rámci Žilinského kraja a fungovať by mal prostredníctvom webového rozhrania. Pri návrhu systému bude kladený dôraz na použitie frameworkov s cieľom zjednodušiť proces tvorby systému.

Abstract

The aim of this thesis is to design a system for the purposes of registry of fires of the Fire and rescue department of the Slovak Republic. System should serve to members of the department for administration of records about all fires within the Žilina Region and it should work through the web interface. Emphasis will be placed on using frameworks with the aim to simplify the process of development of the system.

Kľúčové slová

systém, webová aplikácia, relačná databáza, framework, evidencia, HTML, PHP, CSS, požiare.

Keywords

system, web application, relational database, framework, registry, HTML, PHP, CSS, fires.

Bibliografická citácia

MEŠKOVÁ, L. *Návrh systému pro evidenci požárů*. Brno: Vysoké učení technické v Brně, Fakulta podnikatelská, 2016. 91 s. Vedoucí diplomové práce Ing. Aleš Klusák, Ph.D..

Čestné prehlásenie

Prehlasujem, že predložená diplomová práca je pôvodná a spracovala som ju samostatne. Prehlasujem, že citácie použitých prameňov sú úplné a že som vo svojej práci neporušila autorské práva (v zmysle Zákona č. 121/2000 Sb., o právu autorskom a o právach súvisiacich s právom autorským).

V Brne dňa 25. mája 2016

.....
Podpis

Pod'akovanie

Ďakujem pánovi doktorovi Alešovi Klusákovi za jeho cenné rady a odbornú pomoc pri písaní tejto práce. Ďalej ďakujem Hasičskému a záchrannému zboru v Martine za ústretovosť a poskytnutie informácií na spracovanie práce.

Obsah

Úvod.....	19
Vymedzenie problému a cieľ práce	12
1 Teoretické východiská práce.....	13
1.1 Základné pojmy.....	13
1.1.1 Databázové systémy	13
1.1.2 Dáta	14
1.1.3 Užívatelia	15
1.1.4 SRBD	15
1.2 Dátové modely	16
1.3 Relačný dátový model.....	17
1.3.1 Relácie	17
1.3.2 Entity.....	18
1.3.3 Atribúty.....	18
1.3.4 Domény.....	18
1.3.5 Vzťahy	18
1.3.6 Integrita	18
1.3.7 Modelovanie vzťahov	20
1.3.8 Normalizácia	20
1.4 Jazyk PHP	21
1.5 Framework	22
1.5.1 Výhody.....	22
1.5.2 Výber frameworku	22
1.5.3 Laravel framework.....	24
1.6 HTML5	26

1.7	CSS.....	27
1.7.1	Responzívny webdesign	29
1.7.2	CSS framework.....	30
1.7.3	Najznámejšie frameworky	32
1.7.4	Bootstrap.....	35
1.8	SWOT analýza	36
2	Analýza problému a súčasnej situácie	37
2.1	Informácie o zbore	37
2.1.1	Kompetencie a úlohy	37
2.1.2	Organizačná štruktúra	38
2.2	Informačné technológie zboru	39
2.3	Analýza súčasného systému.....	42
2.3.1	Administratíva	42
2.3.2	Bezpečnosť	45
2.3.3	Nedostatky	45
2.4	Potreba nového riešenia	46
2.4.1	Užívatelia	47
2.4.2	Objekty.....	48
2.5	SWOT analýza súčasného systému.....	52
2.6	Zhodnotenie pripravenosti na nové riešenie	53
3	Vlastný návrh riešenia.....	55
3.1	Návrh databázy	55
3.1.1	Tvorba tabuliek	55
3.1.2	Databázová schéma.....	58
3.1.3	Databázová schéma Laravelu	61

3.1.4	Import a export číselníkov	62
3.2	Laravel framework.....	63
3.3	Návrh webovej aplikácie.....	64
3.3.1	Migrácia databázy.....	64
3.3.1.1	Migrácie	64
3.3.1.2	Seeders	65
3.3.2	Routing a controllers.....	67
3.3.3	Bootstrap	68
3.4	Autentifikácia.....	69
3.4.1	Užívatelia	70
3.4.2	Užívateľské rozhranie	71
3.5	Tvorba formulárov	72
3.5.1	Základné elementy	73
3.5.2	Evidenčné číslo	75
3.5.3	Formuláre	75
3.5.4	Responzívny design	79
3.5.5	Validácia	80
3.6	Odhadovaný rozpočet	83
3.7	Prínosy navrhnutého systému	84
Záver		85
Zoznam použitých informačných zdrojov		87
Zoznam obrázkov		90
Zoznam tabuliek		91
Prílohy.....		I

Úvod

V dnešnej dobe napreduje vývoj technológií obrovskou rýchlosťou. V rámci elektroniky a elektronických médií je cieľ jasný – chceme mať dostupné všetko, vždy a všade. Túto požiadavku naplňame najmä prostredníctvom internetového pripojenia. Činnosti, ktoré by nás pár desiatok rokov dozadu vôbec nenapadlo riešiť cez internet, dnes vykonávame takýmto spôsobom na dennom poriadku. Či už je to písanie, telefonovanie, sledovanie televízie, videí, nakupovanie alebo správa bankového účtu. Všetky zmeny vedú k urýchleniu vykonania činností, čo je v tejto uponáhľanej dobe prirodzený vývoj.

Okrem vyššie uvedeného sa novším fenoménom stal aj prevod evidencie do elektronickej podoby, do ktorého sa zapojila nejedna veľká inštitúcia. Súčasťou ich bežného fungovania sa stalo aj online zálohovanie dát. Týmto spôsobom eliminujú riziko straty dát napríklad fyzickým poškodením.

K používaniu spomínaných technológií už pristúpili takmer všetky organizácie vrátane štátnych inštitúcií. To by bolo žiadúcim krokom aj pre Hasičský a záchranný zbor Slovenskej republiky, ktorý do súčasnej doby používal na evidenciu záznamov o požiaroch systém, ktorý sa inštaluje na počítač lokálne. Udržovanie elektronickej databázy by viedlo nielen ku komfortnejšiemu používaniu a k zvýšeniu efektivity práce, no zaručilo by oveľa väčšie bezpečie pre citlivejšie údaje, s akými sa tu narába.

Vymedzenie problému a cieľ práce

Cieľom tejto diplomovej práce je navrhnúť vhodný systém pre účel evidencie požiarov Hasičského a záchranného zboru Slovenskej republiky. Navrhnutý systém by mal slúžiť členom zboru na evidenciu detailných informácií o všetkých požiaroch v rámci Žilinského kraja. Prostredníctvom webového rozhrania by mal systém umožniť manipulovať s uloženými dátami a ukladať nové záznamy.

Evidencia prebiehala v zastaranom systéme vyvinutom pomocou nástroja Visual FoxPro od Microsoftu. Nový systém sa bude opierať o štruktúru pôvodného systému a importujú sa do neho zaužívané číselníky. Všetky nové údaje budú ukladané do nového systému a budú na webovom portáli zboru prístupné online.

Celý systém bude konštruovaný v jazyku PHP s využitím PHP frameworku Laravel. Laravel umožní vytvoriť potrebnú databázovú schému a navrhnutú databázu migrovať do systému. Využijú sa tu aj jazyky HTML5 a CSS. Pre vybudovanie prehľadného a uceleného webového rozhrania bude systém využívať CSS framework Bootstrap.

Prístup do systému bude rozdelený podľa toho, v akej z nasledujúcich úrovní sa užívateľ nachádza:

1. Ministerstvo vnútra
2. Krajské riaditeľstvo Hasičského a záchranného zboru v Žiline
3. Okresné riaditeľstvá Hasičského a záchranného zboru v rámci Žilinského kraja

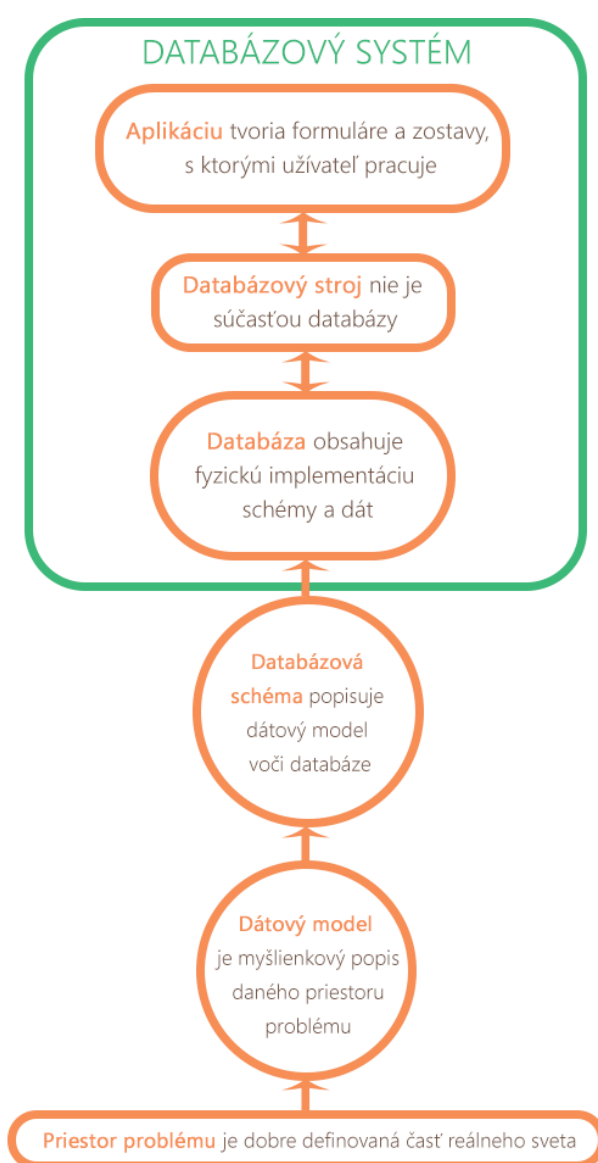
Okrem spomenutých metód bude pri analýze súčasného stavu využitá SWOT analýza súčasného stavu systému.

1 Teoretické východiská práce

1.1 Základné pojmy

1.1.1 Databázové systémy

V dnešnej dobe sú databázové systémy súčasťou väčšiny informačných systémov. Či už sa jedná o systém komerčný, technický alebo technologický, o webovú aplikáciu alebo správu vedeckých dát v rôznych odboroch. Rozvoj informačných systémov viedol k potrebe vyladiť prístup k informáciám v počítačových systémoch tak, aby bol čo najjednoduchší a prehľadný, ale zároveň čo najpresnejší (1).



Obr. 1: Terminológia relačných databáz (Vlastné spracovanie podľa (2))

Slovom „databáza“ môžeme označiť prakticky čokoľvek – môže sa jednať o jednoduchý zoznam, ale aj zložitú množinu rôznych nástrojov, akou je napríklad SQL server (2).

Databázový systém môžeme chápať ako množinu navzájom súvisiacich dát spolu s programom, ktorý umožňuje k daným dátam prístup. Tiež môže byť chápaný ako počítačový systém správy uložených záznamov (1).

Pojem databáza nezahŕňa samotnú aplikáciu, tá sa skladá z formulárov a zostáv, s ktorými užívateľ pracuje (2).

1.1.2 Dáta

„Dáta sú údaje (technické, administratívne, ekonomické, finančné, štatistické, ...), správy alebo informácie, ktoré je možné spracovávať s pomocou technických prostriedkov.“ (1, s. 9)

Dáta sú ukladané ako množina záznamov jednotlivých súborov. V záznamoch sú reprezentované podľa určených typov, ohľad sa berie na najefektívnejší spôsob uloženia. Môžu to byť numerické dáta, znakové reťazce, číselníky alebo jednotky (1).

- **Numerické dáta** môžeme uložiť napríklad binárne, v semilogaritmickom tvare či znakovo. Veľmi často je potrebné definovať aj počet potrebných bajtov pre ukladanie dát zvoleného typu, čím zabezpečíme ich presnosť (1).
- **Znakové reťazce** sa môžu líšiť znakovými sadami, v ktorých sú uložené – napríklad ASCII, UNICODE či EBDIC (1).
- **Číselníky** je vhodné využiť v prípadoch, keď sa dáta dajú reprezentovať pomocou číselných alebo znakových kódov namiesto reťazcov popisujúcich význam nejakej hodnoty danej položky. Dobrým príkladom je systém hodnotenia v škole. Hodnotenie skúšky slovom „výborný“ kódujeme ako „1“, „veľmi dobrý“ ako číslo 2 atď. Na vysokej škole sa väčšinou miesto číselného kódu využívajú kódy znakové (A až F) (1).
- **Jednotky** predstavujú dáta získané pozorovaním či meraním sú vo väčšine prípadov zapísané formou. Tu je vždy dôležité zvážiť ich dátový typ, aby boli čo

najprehľadnejšie, no zároveň dostatočne presné. Napr. pri dĺžkových mierach nemusí byť vždy vhodnou variantov vyjadrenie v milimetroch, či naopak vyjadrenie v kilometroch (1).

1.1.3 Užívatelia

V priamej interakcii so systémom sú takzvaní koncoví užívatelia, ktorí sú z hľadiska prevádzky informačného systému najdôležitejší. Bežne sú užívateľovi sú funkcie systému prístupné pomocou navigácie aplikácie, v ktorej je systém spravovaný. Pokiaľ sa jedná o skúsenejšieho užívateľa, ktorý ovláda programovacie jazyky, dáta mu môžu byť prístupné pomocou dotazov (napríklad v SQL). V ňom si môže presne sformulovať požiadavky na dáta, ktoré potrebuje (1).

Ďalšou skupinou užívateľov sú aplikační programátori, ktorí samotný program využívajúci databázu vytvorili (1).

Posledným druhom užívateľa je databázový administrátor, ktorý je zodpovedný za databázu, umiestnenie dát a spôsob údržby dát. Okrem návrhu dátového modelu je jeho funkciou definovať internú schému (akým spôsobom budú dáta reprezentované pri ukladaní do databázy, ako budú organizované a aké budú prístupové práva), definovať bezpečnostné pravidlá pre prácu s databázou, pravidlá zabezpečenia integrity databázy či monitorovať výkonnosť systému a ladiť ho na základe nových požiadaviek (1).

1.1.4 SRBD

Systém riadenia bázy dát (alebo aj anglicky DBMS – Database Management Systems) tvorí programové vybavenie databázového systému. SRBD spracúva všetky požiadavky užívateľov na prístup databázy – či už sa jedná o pridanie nového súboru do databázy, jeho vymazanie alebo prácu so samotnými dátami (1).

Tento systém vytvára prostredie pre užívateľov takým spôsobom, aby zvládli prácu s databázou bez väčšej znalosti hardvéru, ktorý je na prevádzku použitý (1).

1.2 Dátové modely

Dátové modely môžeme chápať ako konceptuálny popis priestoru problému, do ktorého zahrnieme definície entít, ich atribútov a obmedzenie entity. Zahŕňajú aj popis vzťahov medzi entitami spolu s obmedzeniami platnými pre tieto vzťahy (2).

- **Lineárny model**

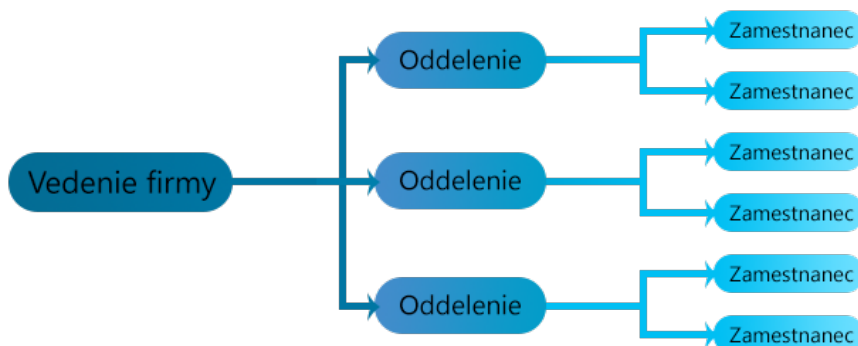
Medzi jednotlivými skupinami objektov (tabuľkami) nie je v lineárnych modeloch žiadna väzba. Z obrázku nižšie teda nemôžeme posúdiť, ktorý odberateľ odoberá ktorý výrobok, ani kto je za ich výrobu zodpovedný. Každý útvar predstavuje tabuľku databázy, medzi ktorými nie je žiadny vzťah (4).



Obr. 2: Lineárny dátový model (Vlastné spracovanie podľa (4))

- **Hierarchický model**

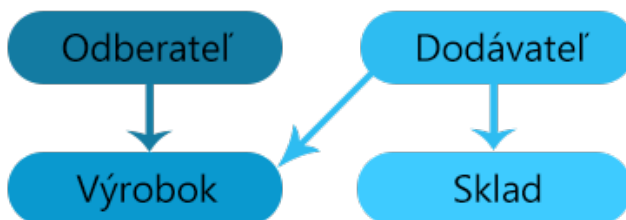
Dáta sú zorganizované do stromovej štruktúry. Je tvorená rodičovským segmentom, z ktorého vedú väzby ku jeho podriadeným segmentom - potomkom. Jedná sa o segmenty inej štruktúry aj obsahu (4).



Obr. 3: Hierarchický dátový model (Vlastné spracovanie podľa (8))

- **Sieťový model**

V tomto modeli sa už nejedná o rodičov a potomkov, ale iba o segmenty. Aj keď je obdobou hierarchického modelu, väzby sú vedené medzi segmentami nie len z rodičovských segmentov na potomkov, ale aj v rôznych iných smeroch (4).



Obr. 4: Sieťový dátový model (Vlastné spracovanie podľa (4))

- **Relačný model**

Relačný model definuje spôsob, akým sa môžu dáta reprezentovať - teda predstavuje ich štruktúru, spôsob ochrany a operácie, ktoré môžeme s nimi prevádzať. Patrí medzi najobľúbenejšie techniky tvorby databáz vďaka svojej vysokej efektívnosti a flexibilitě (2).

- **Objektový model**

Tento model je najnovším dátovým modelom a je vystavený základnému prvku – objektu. Objekt má okrem svojich atribútov definované aj metódy, ktoré určujú jeho chovanie. Volanie niektorej z metód objektu je jediný spôsob, ako s objektom pracovať. Docielime tým vysokú dátovú abstrakciu a tiež nezávislosť dát (4).

1.3 Relačný dátový model

1.3.1 Relácie

Relácie sú čisto pomyselnou a konceptuálnou konštrukciou s riadkami a stĺpcami. Po realizovaní v databáze sa z nich stávajú množiny záznamov a výsledné množiny už sú fyzickými objektami. Nazývame ich tabuľkami (2).

1.3.2 Entity

Entitou môžeme nazvať v podstate čokoľvek, o čom potrebujeme uchovávať nejaké informácie. Modeluje objekty alebo udalosti, najčastejšie v reálnom, fyzickom svete – napríklad zákazníci, výrobky, objednávky, pôžičky – všetko sú to **konkrétne entity** (2).

Abstraktné entity modelujú abstraktné pojmy (myšlienky). Najbežnejším prípadom je entita, ktorá modeluje určitý vzťah medzi inými entitami. Niekedy nám stačí namodelovať iba samotnú existenciu takéhoto vzťahu (2).

1.3.3 Atribúty

O každej entite sa zaznamenávajú určité skutočnosti – údaje, ktoré nazývame atribútmi danej entity. Samotné určovanie atribútov je takzvaný sémantický proces – rozhodujeme sa podľa významu dát a tiež podľa toho, akým spôsobom ich môžeme využiť (2).

1.3.4 Domény

Doména, alebo aj obor hodnôt, vyjadruje druh dát, ktoré atribút reprezentuje. Jedná sa o logický pojem, ktorý sa často zamieňa s dátovým typom – ten je však pojmom fyzickým. Zatiaľ čo „číslo“ je dátový typ, slovo „vek“ už vyjadruje obor hodnôt (2).

„Obor hodnôt tvorí množina všetkých prípustných platných hodnôt, ktoré môže určitý atribút obsahovať.“ (2, str. 16)

1.3.5 Vzťahy

V dátovom modeli je potrebné definovať aj vzťahy medzi entitami. Entity, ktoré sú do nejakého vzťahu zapojené, označujeme ako účastníkov, počet účastníkov značí stupeň vzťahu (2).

1.3.6 Integrita

Súčasťou dátového modelovania je aj modelovanie dátovej integrity – teda snaha zaistiť, aby skutočné fyzické dáta uložené v databázovom systéme boli správne alebo aspoň prinajmenšom dôveryhodné. Dáta teda musia vyhovovať istým obmedzujúcim podmienkam, ktoré pre nich boli definované (2).

Pravidlá pre údržbu integrity jednotlivých relácií sú definované doménovými, prechodovými a entitnými obmedzeniami (2).

- **Doménová integrita**

Doménové obmedzenie je pravidlo, ktoré definuje všetky prípustné hodnoty určitého daného atribútu. V niektorých prípadoch je nevyhnutné definovať aj viac ako jedno doménové obmedzenie. Okrem správne zvoleného logického dátového typu treba zvoliť mierku a presnosť číselného typu, resp. maximálnu dĺžku hodnoty reťazca (2).

- **Prechodová integrita**

Obmedzenia prechodovej integrity definujú stavy, ktorými môže vektor hodnôt prechádzať. Stavom môže byť napríklad stav objednávky – od počiatočného stavu, ktorý môžeme nazvať napríklad “aktívna objednávka”, cez “čaká na vyexpedovanie” až po “doručená”. Prechodová integrita môže byť považovaná za istý špeciálny typ doménovej integrity (2).

Využíva sa tu cudzí kľúč – je to atribút, ktorý nám umožňuje spolu s primárnym kľúčom vytvárať medzi reláciami spojenie, čo je vlastne hlavným účelom relačného dátového modelu (4).

- **Entitná integrita**

Tieto obmedzenia zabezpečujú integritu entít modelovaných v používanom systéme. Za takéto obmedzenie môžeme považovať existenciu primárneho kľúča. Primárny kľúč zaisťuje pravidlo, podľa ktorého musíme vedieť každú jednu entitu jednoznačne identifikovať. Entitné obmedzenie špecifikuje podmnožinu množiny definovanej doménovým obmedzením, nie však jeho nadmnožinu (2).

Môžeme sa tu stretnúť s kandidátnym kľúčom, ktorý ma rovnaké vlastnosti ako kľúč primárny, ale nie je vybraný ako primárny kľúč (4).

1.3.7 Modelovanie vzťahov

Akonáhle medzi dvoma entitami existuje určitý vzťah, je potrebné ho správne namodelovať. Integritné obmedzenie pre vzťahy obmedzuje kardinalitu vzťahu na pomery 1:1, 1:N, N:1 a N:M (4).

- **1:1**

Jedná sa o najjednoduchší typ vzťahu. Platí, že každú ľubovoľnú inštanciu entity X je možné asociovať iba s jednou inštanciou entity Y (2).

- **1:N**

Najbežnejší typ vzťahu medzi entitami je vzťah jedna k viac. Tu môžeme jednu inštanciu entity X asociovať s nulou, jednou alebo viacerými inštanciami druhej entity Y. Pri vzťahu N:1 je tomu naopak (2).

- **N:M**

Často sa stretávame aj so vzťahmi viac ku viac. Študent môže byť pre príklad zapísaný na viacero cvičení a zároveň platí, že na cvičenie môže byť zapísaných viacero študentov. Tieto typy vzťahov sa modelujú pomocou špeciálneho prostredníka – špeciálnej relácie, ktorá má s každým s oboch pôvodných účastníkov vzťah jedna k jednej. Ako prostredník sa zvyknú vytvárať spojovacie tabuľky, ktoré obvykle obsahujú iba kandidátne kľúče dvoch pôvodných účastníkov (2).

1.3.8 Normalizácia

„Normalizácia je činnosť, pri ktorej upravujeme návrhy dátových štruktúr tak, aby splňovali zvolené normalizačné formy – úrovne. Tieto normalizačné formy či pravidlá vychádzajú z požiadavku na efektívne ukladanie dát a minimalizujú redundancie zachovaní integrity a konzistencie dát.“ (4, s. 55)

Dátový model by nemal porušovať normalizačné formy. V inom prípade nie je navrhnutý optimálne. Proces normalizácie vedie ku dekompozícii relácií (tabuliek) do vhodnejšieho tvaru (4).

Normálne formy:

- **Prvá normálna forma** – každý atribút musia byť jednoduché, nie zložené alebo viachodnotové. Každý riadok by mal zároveň obsahovať jednoznačný identifikátor – primárny kľúč (4).
- **Druhá normálna forma** – relácia spĺňa prvú normálnu formu a navyše sú všetky atribúty závislé na celom kandidátom (primárnom) kľúči (4).
- **Tretia normálna forma** – relácia je v druhej normálnej forme a zároveň všetky jej neklúčové atribúty musia byť navzájom nezávislé (4).
- **Štvrtá normálna forma** – Boyce-Coddova normálna forma, ktorá je variáciou tretej normálnej formy, ak je relácia v tejto forme, je tiež v tretej normálnej forme, avšak naopak to neplatí.

1.4 Jazyk PHP

PHP je veľmi populárny skriptovací jazyk, ktorý sa využíva najmä pri vývoji webových aplikácií. Pri tomto vývoji je potrebné prevádzkovať webový server, s ktorým možno PHP prepojiť. Môže sa jednať napríklad o server Apache alebo IIS. Jadro PHP je tvorené Zend Engine, ktorý rozoberá a spúšťa zdrojový kód skriptov (5).

Architektúra PHP je rozdelená na jednu časť kódu, ktorá je špecifická pre jednotlivé prostredia (webové servery alebo príkazový riadok SAPI). Ostatný kód je rozdelený do extenzií ktoré sú zodpovedné vždy za jednu činnosť. Napríklad za prácu s poľami, reťazcami alebo s databázou. Každá takáto extenzia definuje kód, ktorý je spustený pri štarte a ukončení PHP, pri spustení a ukončení každého jedného požiadavku a definuje aj jednotlivé funkcie a triedy (5).

Inšpiráciu PHP nachádza najmä v programovacom jazyku C. Zdrojové kódy PHP sa dajú stiahnuť z php.net, aktuálnu vývojovú verziu nájdeme k dispozícii v repozitári SVN a Git a pre každú vývojovú verziu je k dispozícii samostatná vetva (tzv. branch). Kód PHP je štruktúrovaný – je veľmi dobre čitateľný a prehľadný. Nie je preto problém do neho nahliadnuť a pozrieť sa na konkrétnu implementáciu nejakej funkcie, aj keď nie je dobre zdokumentovaná (5).

1.5 Framework

Pri programovaní by jedným z hlavných cieľov malo byť uľahčiť a ušetriť si prácu. Či už sa jedná už len o použitie klávesových skratiek či o správny návrh funkcií a tried. V ešte väčšom rozmedzí si však prácu môžeme ušetriť používaním existujúcich knihoovní a nástrojov (5).

Menším či väčším sadám kódov, na ktorých sa dá postaviť celá stránka, hovoríme **frameworky**. V dnešnej dobe je veľmi jednoduché, vopred vytvorené frameworky na internete nájsť a vďaka návodom aj ľahko použiť pri svojej práci (13).

1.5.1 Výhody

Najväčšou výhodou ostáva, že aj keď vytvárame nový projekt, nie je potrebné riešiť všetky detaily a tiež problémy odznova – riešenie, ktoré je verejne dostupné a používa sa už dlhšiu dobu možno považovať za pomerne spoľahlivé. Pokiaľ sa tu vyskytla nejaká chyba alebo problém, s veľkou pravdepodobnosťou ho už v zdieľanom frameworku vyriešili a opravili. Samozrejme, vždy sa nájde časť ľudí, ktorí budú považovať svoje riešenia za lepšie. Na druhej strane dielo tímu programátorov a ľudí, ktorí využívajú tieto nástroje denne, rozhodne stojí za uváženie (13).

Medzi ďalšie výhody patrí istý štandard, akým sú na ňom aplikácie vytvorené. Vďaka nemu každý programátor, ktorý je v týchto technológiách znalý, presne vie, kde v prípade nejakého problému hľadať. Tým sa zvyšuje aj počet ľudí, ktorí sú schopní a zároveň ochotní pomôcť – na rôznych diskusných fórach je odhalenie príčiny problému otázkou niekoľkých hodín, v niektorých prípadoch dokonca minút (13).

1.5.2 Výber frameworku

Výber správneho frameworku je pre programátora veľmi dôležitý – ten ho totiž bude sprevádzať po veľmi dlhú dobu projektu. V prípade použitia nesprávneho frameworku bude mať opačný, negatívny efekt – programátor sa môže pri jeho zavedení trápiť, prípadne sa mu nedostane potrebnej pomoci a výsledne príde o viac času, ako mal pôvodne ušetriť. Pri výbere správneho frameworku by jeho inštalácia mala prebehnúť bez problémov, pri prípadných problémoch by sa mal užívateľ stretnúť so spätnou väzbou

a celkovo by jeho praktické využívanie malo šetriť čas a prinášať isté pohodlie pri tvorbe stránky (13).

Dôležitým faktorom, ktorý treba zvážiť, je **náročnosť na výkon**. Framework, ktorý nie je optimalizovaný pre výkon, obsahuje mnoho zbytočných častí, ktoré zaťažujú pamäť a zároveň spôsobujú veľké oneskorenie medzi jednotlivými požiadavkami na aplikáciu (13).

V niektorých situáciách je potrebné framework istým spôsobom rozšíriť. Práve preto by mal byť kód napísaný jednoduchým a prehľadným štýlom a mala by byť k nemu dostatočná dokumentácia. Dobře vyriešená **rozšíriteľnosť frameworku** sa určite odrazí na výslednej kvalite práce (13).

Netreba zabúdať na **bezpečnosť**. Samozrejme, že sa stane, že sa vo väčšom kóde nájde nejaká malá chyba. Tá sa časom opraví, avšak väčšia chyba v celkovom návrhu už vedie k vytvoreniu novej, častokrát nekompatibilnej verzii frameworku. Nahrádzanie frameworku novým môže byť pomerne komplikované a časovo náročné. Treba si preto ideálne pred zavedením samotného frameworku prejsť užívateľské recenzie a opierať sa o čo možno najlepší framework bez hlásenej chybovosti (13).

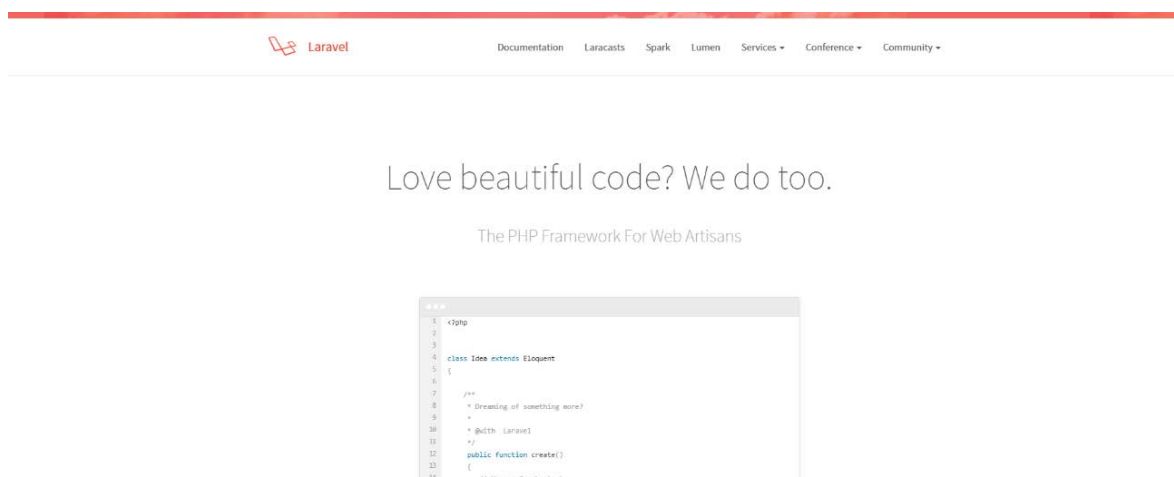
Najpoužívanjšie frameworky a ich výhody (9):

- **CakePHP** - množstvo návodov v češtine
- **CodellIgniter** – jednoduchý a intuitívny, vyvíjaný firmou
- **Fuse** – generuje celú kostru vyvíjanej aplikácie
- **Jelix** – podpora rôznych výstupných formátov (okrem XHTML) ako XUL, RSS, ATOM, XML, PDF atď.
- **Kohana** – čisto v PHP5, vývoj komunitou
- **Laravel** – výborná podpora, video tutoriály, dobre spracovaná dokumentácia, prehľadný čistý kód
- **Nette** – od českého autora, stále vo fáze vývoja
- **PRADO** – založený na komponentovom prístupe a programovaní udalostí
- **Qcodo** – veľká rýchlosť aktualizácie dotazu (zapisovanie do databáze)

- **RoR** (Ruby on Rails) – veľmi rozšírený, avšak nie je použiteľný pre PHP, ale len pre jazyk Ruby
- **Symfony2** – rozdelenie na komponenty, možnosť kombinácie s iným frameworkom
- **Zend** – populárny framework, ponúka veľa možností

1.5.3 Laravel framework

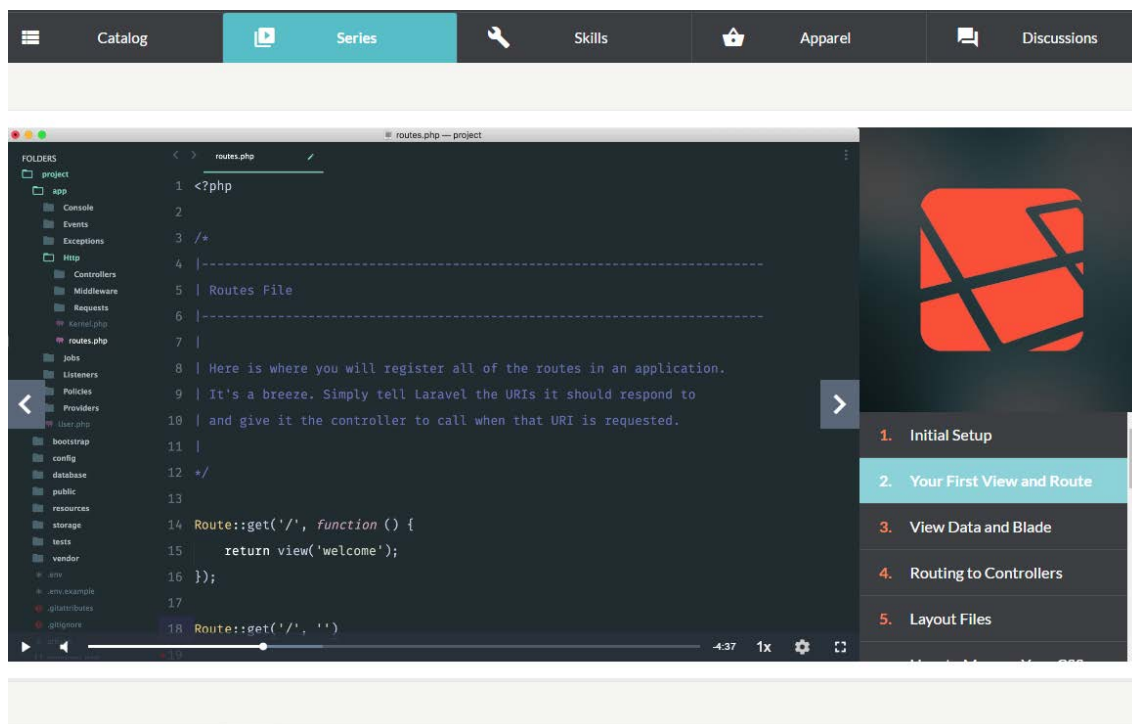
Pre prácu som si zvolila framework Laravel. Jedná sa o MVC framework, ktorý v poslednej dobe nabral na veľkej popularite a v minulom roku údajne patril medzi najpoužívanéjšie frameworky vôbec (10).



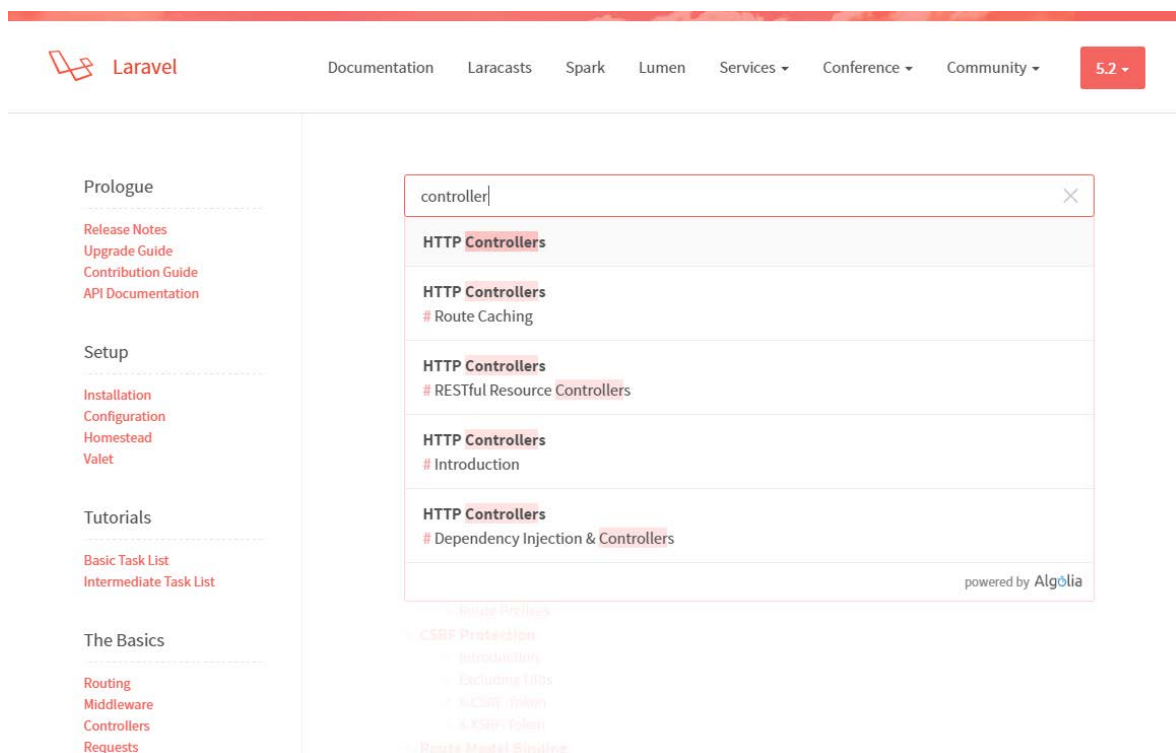
Obr. 5 Webová stránka frameworku Laravel (12)

Je známy pre svoju syntax, ktorá je veľmi prehľadná a jednoduchá. Inštalácia by mala prebiehať bez väčších problémov, všetky kroky môžeme nájsť na webe pekne popísané a vysvetlené (10).

Ako som spomínala, Laravel poskytuje v spolupráci s webovou stránkou pre vývojárov Laracasts.com nápomocné video tutoriály, ktoré sú vhodné aj pre začínajúcich programátorov a pomôžu či už pri inštalácii, tak aj pri náročnejších činnostiach. Môžeme ich nájsť stovky (12).



Obr. 6 Video tutoriál stránky Laracasts (23)



Obr. 7 Riešenie dokumentácie frameworku Laravel (12)

Veľkým plus je najmä perfektná dokumentácia. Ako vidíme na obrázku (Obr. 7), Laravel má prepracované vyhľadávanie príkazov a v podstate akýchkoľvek reťazcov dokumentácie (12).

Okrem aktualizácie dokumentácie autor neustále pracuje na vylepšení celého frameworku a často komunikuje s rôznymi blogermi o pripravovaných funkciách a novinkách ešte predtým, ako sú verejne prístupné, čím zaručí ich maximálne zviditeľnenie (10).

1.6 HTML5

Pri budovaní webovej aplikácie bude použitý aj kódovací jazyk HTML5, preto si k nemu povieme pár slov.

Tento jazyk bol vyvinutý z predchádzajúcich verzií HTML, avšak snaží sa čo najviac prispôbiť súčasným požiadavkám. Jazyk získal mnoho užitočného na základe toho, čo v predchádzajúcich verziách chýbalo. Priniesol mnoho výhod a tiež novú syntax, ktorú si vývojári webov museli pri písaní kódu osvojiť. No táto nová syntax zbaví všetkých zbytočných zápisov. Kľúčovou vlastnosťou návrhu tohoto jazyka bolo, aby čo najväčšia časť bola spätne kompatibilná (23).

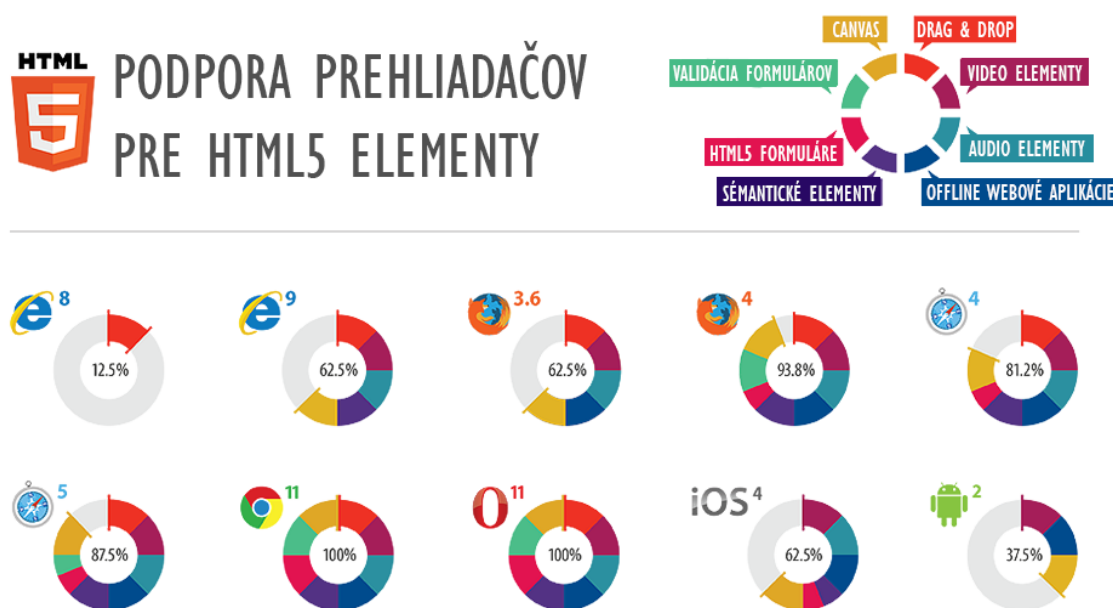
Jazyk HTML5 priniesol oproti posledným verziám veľa nových funkcií, niektoré z nich veľmi jednoduché - napríklad doplnkové elementy pre popis obsahu stránky (23):

- article – definovanie článku
- aside – “bočný“ obsah, odlišný od hlavného obsahu stránky
- header – hlavička stránky
- section – časť, sekcia dokumentu
- figure – obrázok, nezávislý obsah
- footer – päta stránky
- audio – zvukový obsah
- video

Zložitejšie funkcie pomáhajú pri tvorbe webových aplikácií. Čo však bolo veľmi dôležité, jazyk HTML5 zaviedol podporu prehrávania zvukových a video súborov, a to priamo cez

webový prehliadač (bez využitia nejakého plug-inu, ako napríklad Adobe Flash Player alebo QuickTime) (23).

HTML5 je podporovaný vo všetkých moderných prehliadačoch ako Safari, Chrome, Firefox, Opera a Internet Explorer, ktoré postupne pridávajú nové funkcie HTML5. Podpora prehliadačov pre jednotlivé elementy je rôzna, bližšie nám to môže priblížiť nasledujúca šikvná infografika od spoločnosti WebEnertia (Obr. 8).



Obr. 8 Podpora prehliadačov pre HTML5 elementy (vlastné spracovanie podľa (24))

1.7 CSS

Pod skratkou CSS rozumieme takzvané kaskádové štýly (jedná sa o skratku z anglického Cascading Style Sheets), ktoré vznikli na konci roka 1996. Aj napriek relatívne dlhej existencii sa CSS v skutočnom svete využívalo len veľmi málo, bolo obmedzené na správu písiem a farieb. Spôsobila to najmä nekonzistencia podpory v rôznych druhoch prehliadačov. Niektoré prehliadače dokonca nepracovali s kaskádovými štýlmi vôbec. Skutočný potenciál štýlov preto v začiatkoch nebolo možné využiť naplno a istotou ostával design upravený v HTML. Tento spôsob bol samozrejme veľmi nepraktický a kaskádové štýly sa postupne dostávali do povedomia užívateľov, dizajnérov

a vývojárov. Podpora kaskádových štýlov sa preto rapídne zvyšovala a nové verzie webových prehliadačov už s CSS pracovali (7).

V dnešnej dobe sa kaskádové štýly pri designe stránok využívajú naplno so všetkými cennými funkciami, ktoré ponúkajú. Ku HTML kódu priniesli veľké množstvo výhod, tu je pár príkladov (7):

- CSS umožňuje nastaviť štýly pre jeden, ale aj pre stovku dokumentov
- Pri zmene CSS kódu na jednom mieste sa zmení štruktúra všetkých dokumentov, ktoré majú k sebe tento kód pripojený, a to bez potreby do nich iným spôsobom zasahovať
- CSS kód oddelený od dokumentu má za následok to, že samotný dokument je menší a rýchlosť načítavania a zobrazenia stránky je omnoho vyššia
- CSS dokumenty sa načítavajú a ukladajú do cache, čo znamená, že sú načítané do pamäte prehliadača iba raz (pokiaľ neprebehlo premazanie cache). Prehliadač preto nemusí štýly znovu interpretovať, čo má znova za výsledok oveľa rýchlejší prechod medzi stránkami a rýchlejšie načítanie samotných stránok. Rýchla odozva technológií je dnes samozrejmosťou.
- Ponúka sa možnosť nastaviť štýly dokumentov v závislosti na médiách, s ktorými sú zobrazené, či už na monitoroch, pri tlači, projekcii, na mobilných telefónoch alebo na tabletoch.

Termín **kaskáda** alebo kaskádový štýl je vlastne hierarchia aplikácie, čo vystihuje systém aplikovania pravidiel. Existujú rôzne prostriedky aplikovania štýlov na rovnaký dokument, okrem východiskového (primárneho) štýlu prehliadača môžeme definovať štýl užívateľský, individuálny, vložený, pripojený či importovaný (7).

Jednotlivé druhy štýlov (7):

- **Štýly prehliadača** – štýly nastavené primárne.
- **Užívateľské štýly** – môžu byť užitočné pre náročných užívateľov, ktoré majú špeciálne požiadavky. Napríklad pre užívateľov s horším zrakom sa môže

vytvoriť špeciálny kontrastný štýl, ktorý zväčší písmená a nahradí vopred definované štýly.

- **Individuálne štýly** - používajú sa v jednotlivých prvkoch pomocou atribútu ! “style“ v HTML dokumente. Na časté používanie sa neodporúčajú, avšak jednorazovo poslúžia účelu.
- **Vložené štýly** – sú nastavované len pre jeden dokument a umiestnené sú vo vnútri elementu “style“.
- **Pripojené štýly** – externé súbory so štýlmi, ktoré sú v záhlaví dokumentu pripojované pomocou elementu “link“. Všetky dokumenty, ku ktorým sú tieto súbory pripojené, používajú štýly v ňom definované.
- **Importované štýly** – umožňujú importovať štýly do pripojeného externého súboru so štýlmi alebo do dokumentu, inak sa od pripojených štýlov nelíšia.

Kaskáda určuje spôsob, akým budú tieto pravidlá aplikované. Užívateľské štýly potlačia všetky ostatné, individuálne potlačia vložené, pripojené a importované štýly. Vložené majú pred pripojenými prednosť. Pripojené a importované štýly sú však rovnocenné a aplikujú sa všade tam, kde nie sú použité iné druhy štýlov. Až v prípade, že nie sú použité iné štýly, aplikujú sa východiskové štýly prehliadača. Pokiaľ by došlo ku kolízii štýlov, kaskáda interpretuje posledný pripojený štýl (7).

1.7.1 Responzívny webdesign

Nezastaviteľný vývoj informačných technológií má za následok toho, že všetko treba neustále prispôbovať novým veciam a podmienkam. Čo sa týka webových stránok, platí to dvojnásobne. Okrem zmien programovacích jazykov (a vývoju nových), zmien návrhových vzorov či zmien v obľúbenosti prehliadačov, najväčší “boom“ zažívajú zariadenia, akými sa dá k webovým stránkam pripojiť. Od zväčšujúcich či zmenšujúcich sa mobilných telefónov, tabletov, Ipadov, rôznych zariadení, ktoré sú ich kombináciou alebo od nástupu obrovských monitorov so špičkovým rozlíšením, nový prístup k návrhu webových stránok začal byť nevyhnutnou súčasťou života programátorov a webových dizajnérov. Tento prístup uprednostňuje jednotný užívateľský prežitok a viedol ku vzniku takzvaného responzívneho webdesignu (6).

Pri nástupe väčších obrazoviek sa prirodzene bolo potrebné rozhodnúť, na aké rozmery stránky bude treba zväčšiť a ako budú na danom zariadení v skutočnosti vyzerieť. Nástup menších mobilných zariadení mal podobný efekt, avšak stránky sa museli prispôbiť naopak menším rozmerom a dotykovým obrazovkám. Ďalšou komplikáciou sa stala možnosť mobilné telefóny, tablety či dokonca monitory natáčať do vodorovnej či zvislej polohy. Vznikla teda potreba webovej stránky prispôbiť akémukoľvek zariadeniu (6).

„Responzívny design je sada techník a technológií, ktorých účelom je sprístupniť aplikáciu čo najviac užívateľom na najrôznejších zariadeniach.“ (6, s. 16)

1.7.2 CSS framework

S responzívnym webdesignom priamo súvisia CSS frameworky, ktoré nám ho pomáhajú veľmi rýchlo a šikovne vytvoriť.

„Framework je definovaný ako štandardizovaný súbor pojmov, postupov a kritérií pre riešenie určitého typu problému, ktorý môže byť použitý ako referencia pre priblíženie a vyriešenie nových problémov podobného charakteru.“ (14)

Jednoduchšie povedané, framework môžeme definovať ako balíček tvorený štruktúrou súborov a priečinkov štandardizovaného kódu (HTML, CSS, JS dokumenty atď.), ktorý môže byť použitý na podporu vývoja webových stránok alebo ako základňa, z ktorej možno začať stavať web (14).

Väčšina webových stránok zdieľajú veľmi podobnú, ak nie identickú štruktúru. Cieľom frameworku je poskytnúť spoločné štruktúry, vďaka ktorým vývojári nemusia pracovať znova od nuly a kód môžu znovu voľne použiť. Týmto spôsobom nám framework umožní ušetriť veľa práce a času (14).

Rozlišujeme **dva typy frameworkov** na základe toho, či sú na prezentačnej alebo aplikačnej (logickej) vrstve (14):

- Back-end
- Front-end

PREZENČNÁ VRSTVA / FRONT-END

Užívateľské rozhranie

Jazyky: HTML, CSS, Javascript...

Framework: Bootstrap, Foundation 3, systémy mriežok...

APLIKAČNÁ VRSTVA / BACK-END

Logika a operácie webovej stránky

Jazyky: PHP, PYTHON, RUBY, JAVA...

Framework: Symfony, Django, Ruby On Rails, Spring...

DÁTOVÁ VRSTVA

Perzistentné dáta v databáze

Technológie: MySQL, PostgreSQL, NoSQL...

Obr. 9 Vrstvy architektúry (Vlastné spracovanie podľa (14))

CSS frameworky sú teda na vrstve prezenčnej a nápomocné sú najmä webovým dizajnérom. V prípade, že dizajnéri pracujú na projekte vo veľkom tíme, frameworky sú pri tímovej práci na nezaplatenie – výsledkom je ucelený vzhľad stránky. Tiež sa využívajú vtedy, ak má jedna spoločnosť viacero stránok s rozličným obsahom – podobnosť designu stránok pôsobí v tomto prípade veľmi dobre. Samozrejme, že aj jednotliví dizajnéri môžu nájsť časti frameworku, ktoré budú pre nich užitočné (20).

CSS framework má za hlavnú úlohu to, aby budovanie konečného produktu (stránky) prebiehalo čo najrýchlejšie a najjednoduchšie. Ich štruktúra väčšinou pozostáva z použiteľných blokov kódu. Tieto bloky kódu sa môžu časom upravovať tak, aby boli stabilnejšie a opakovane použiteľné naprieč projektami (21).

CSS framework sa môže vyskytnúť v rôznych variantoch. Všeobecne má však za cieľ vybudovať štruktúru zahŕňajúcu najmä nasledovné (21):

- mriežka / základný layout
- typografia
- formuláre
- základné štýly
- widgety (ako napríklad taby, drop downs, tlačidlá a ďalšie špecifické objekty a dizajnové prvky)

Niektoré frameworky vytvárajú štruktúru pre jednu z vyššie uvedených možností, niektoré sa pokúšajú zostaviť štruktúru viacerých alebo všetkých možností dohromady (21).

Framework by mal byť tvorený aspoň týmito základnými štýlmi (20):

- **reset.css** - spracúva kompletne resetovanie
- **type.css** - spracúva typografiu
- **grid.css** - spracúva rozvrhnutie mriežky
- **widgets.css** – ovláda widgety
- **base.css** - zahŕňa všetky ostatné štýly, takže nám na používanie celého frameworku stačí zavolať base.css z našich HTML dokumentov

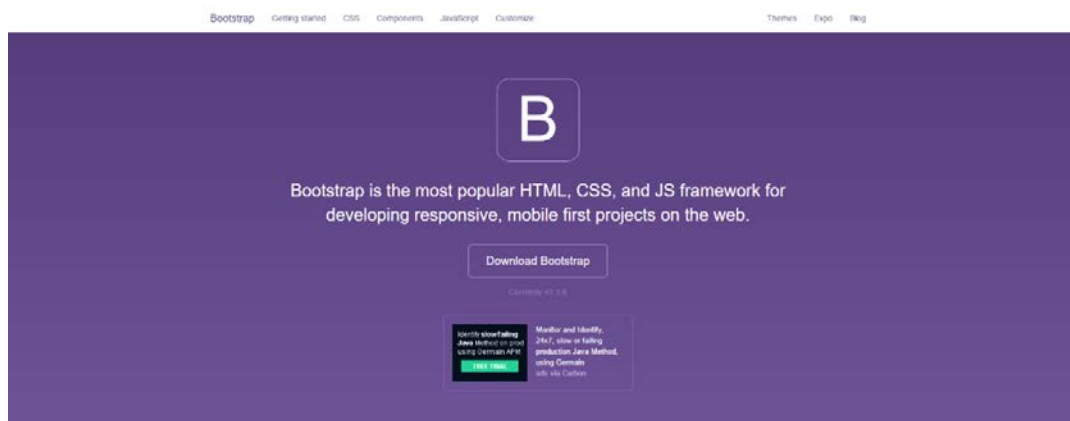
V podstate môžeme povedať, že aj väčšina CMS sa kvalifikuje ako framework. Pod skratkou CMS rozumieme systém pre správu obsahu (z anglického content management system). Tým je napríklad redakčný systém WordPress, ktorý slúži ako základ pre vybudovanie kompletnej webovej stránky. Použitie CMS je v dnešnej dobe veľmi populárne najmä preto, že pomáha vyvíjať stránky rýchlejšie a jednoduchšie. Základná verzia Wordpressu je dokonca k používaniu zdarma (21).

1.7.3 Najznámejšie frameworky

Medzi najznámejšie HTML a CSS frameworky patria:

- **Bootstrap**

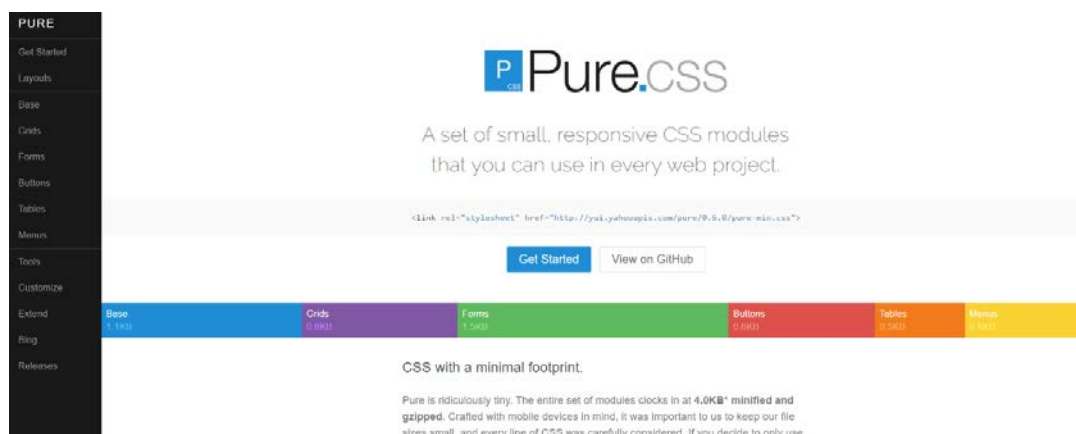
Framework Bootstrap využíva nespočetné množstvo webových dizajnérov najmä pre vytvorenie responzívnych webových stránok (15).



Obr. 10 webová stránka frameworku Bootstrap (17)

- **Pure**

Pure je veľmi efektívny framework, ktorý obsahuje sadu menších, responzívnych CSS modulov. Tie môžu byť použité na takmer akejkol'vek webovej stránke (15).



Obr. 11 Webová stránka frameworku Pure (18)

- **Semantic UI**

Semantic UI ponúka efektívne techniky pre webový design a môžeme ním vytvoriť spoločné užívateľské rozhranie. Pri vytváraní efektívnych webových stránok hrajú veľkú úlohu aj rôzne funkcie a 3D animácie, čo tento framework umožňuje (15).



Obr. 12 Webová stránka Semantic UI (19)

- **Yamli**

Yamli je výborný nástroj pre tvorbu flexibilných, dostupných a predovšetkým responzívnych webov a ponúka efektívne funkcie pre tvorbu webových stránok so všetkými dôležitými prvkami. Tento framework obsahuje tiež dobrý Ajax Builder. Ten zaisťuje kvalitné zázemie pre vývoj dizajnu v Yamli-based CSS layout (15).

- **52Frameworks**

Jedná sa o framework pre HTML a CSS3. Ponúka kompletne nový Form Framework s HTML5 validáciou skrz javascript (15).

- **Blueprint**

Tento framework ponúka kvalitný základ pre vytvorenie projektov s peknými formami a štýlmi. Má veľmi dobrý systém mriežok a prepracovanú typografiu. Je jedným z najlepších frameworkov pre tvorbu komplexných webových dizajnov (15).

- **Kube**

Kube framework je prispôsobivý, responzívny nástroj, ktorý podporuje všetko, čo potrebujete pre vytvorenie dizajnu webovej stránky (15).

- **MontageJS**

MontageJS je moderný front-end HTML5 framework. Podporuje rýchle vytvorenie single-page aplikácií a využíva rokmi preverené dizajnové vzory a softvérové princípy. Tieto princípy umožňujú vytvoriť modulárnu architektúru pre projekty a umožniť tak veľmi vysokú užívateľskú prívetivosť projektov (15).

- **Foundation**

Základom Foundation je skupina front-end frameworkov, ktoré uľahčujú navrhnuť originálne responzívne web stránky, aplikácie a e-maily. Riešenie je sémantické, čitateľné, flexibilné a prispôsobiteľné. Tvorcovia neustále pridávajú nové zdroje a kúsky kódu, vrátane praktických HTML šablón (16).

1.7.4 Bootstrap

Vzhľadom k tomu, že responzívny design je v dnešnej dobe priam nevyhnutným prvkom, zvolenie CSS frameworku nebolo veľmi náročnou úlohou. Vyžadovalo to taký framework, ktorý poskytne jednoduché skripty pre pekný, prehľadný a responzívny design stránky. Mal by mať jednoduchú inštaláciu a zameriavať sa na budovanie formulárov, ktoré budú v systéme najdôležitejšími prvkami. Bootstrap v týchto vlastnostiach spomedzi spomenuté frameworky vyčnieva. Po inštalácii už bude celý systém obsahovať štylovanie, ktoré poskytne Bootstrap. Stačí v HTML kóde zahrnúť určité triedy a štýly. Tie nájdeme na webe Bootstrapu spolu s ukážkou a príkladmi použitia (17).

Forms

Basic example

Individual form controls automatically receive some global styling. All textual `<input>`, `<textarea>`, and `<select>` elements with `.form-control` are set to `width: 100%` by default. Wrap labels and controls in `.form-group` for optimum spacing.

EXAMPLE

Email address

Password

File input

 Nie je vybratý žiadny súbor
Example block-level help text here.☐ Check me out

```
<form>
  <div class="form-group">
    <label for="exampleInputEmail">Email address</label>
    <input type="email" class="form-control" id="exampleInputEmail" placeholder="Email">
  </div>
  <div class="form-group">
    <label for="exampleInputPassword1">Password</label>
    <input type="password" class="form-control" id="exampleInputPassword1" placeholder="Password">
  </div>
  <div class="form-group">
    <label for="exampleInputFile">File input</label>
    <input type="file" id="exampleInputFile">
    <p class="help-block">Example block-level help text here.</p>
  </div>
  <div class="checkbox">
    <label>
      <input type="checkbox"> Check me out
    </label>
  </div>
  <button type="submit" class="btn btn-default">Submit</button>
</form>
```

[Copy](#)[Overview](#)[Grid system](#)[Typography](#)[Code](#)[Tables](#)[Forms](#)[Basic example](#)[Inline form](#)[Horizontal form](#)[Supported controls](#)[Static control](#)[Focus state](#)[Disabled state](#)[Readonly state](#)[Help text](#)[Validation states](#)[Control sizing](#)[Buttons](#)[Images](#)[Helper classes](#)[Responsive utilities](#)[Using Less](#)[Using Sass](#)[Back to top](#)[Preview theme](#)

Obr. 13 Formuláre s frameworkom Bootstrap (17)

1.8 SWOT analýza

Podrobnejšie zhodnotiť súčasný systém z viacerých hľadísk nám pomôže SWOT analýza. Je vlastne analýzou silných a slabých stránok, príležitostí a hrozieb. Vychádza z predpokladu, že rozvíjaním silných stránok a príležitostí a zároveň minimalizovaním slabých stránok a hrozieb dosiahne organizácia úspech (25).

Skratka SWOT pochádza zo začiatočných písmen anglických slov (25):

- Strengths - silné stránky
- Weaknesses – slabé stránky
- Opportunities -príležitosti
- Threats – hrozby

Zo záverov tejto analýzy vyplývajú nasledovné modelové stratégie (25):

- **Ofenzívna stratégia**

Volí sa vtedy, keď prevažujú sily nad slabosťami a príležitosti nad hrozbami. Vzhľadom na svoje sily je firma schopná využiť všetky naskytujúce sa príležitosti.

- **Defenzívna stratégia**

Tu sú v prevahe hrozby. Silné stránky by sa mali využiť na blokovanie nebezpečenstva alebo únik do bezpečnejšieho prostredia. Odporúča sa defenzívne konanie.

- **Stratégia spojenectva**

Pri tejto stratégii prevažujú slabosti nad silami. Odporúča sa pomoc spojenca a postupné odstránenie nedostatkov.

- **Stratégia úniku alebo likvidácie**

Prevládajú tu hrozby aj slabé stránky, riešením je únik do iného prostredia alebo likvidácia aktivít.

2 Analýza problému a súčasnej situácie

2.1 Informácie o zbore

Hasičský a záchranný zbor (ďalej už len HaZZ) ako celok bol založený 1. apríla v roku 2002 zákonom č. 315/2001 Z. z. o Hasičskom a záchrannom zbore (11).



Obr. 14 Oficiálne logo Hasičského a záchranného zboru (11)

2.1.1 Kompetencie a úlohy

Funkcie, ktoré HaZZ vykonáva (11):

- plní úlohy:
 - štátnej správy na úseku ochrany pred požiarimi
 - pri zdolávaní požiarov
 - pri poskytovaní pomoci
 - pri vykonávaní záchranných prác pri haváriách, živelných pohromách, prípadne iných mimoriadnych udalostiach a pri ochrane životného prostredia
 - na úseku materiálneho vybavenia a technického zabezpečenia súvisiace s vykonávaním činností zboru
 - v oblasti výchovy, vzdelávania a odbornej prípravy na úseku ochrany pred požiarimi a v oblasti preventívno-výchovného pôsobenia
- vykonáva štátny požiarneho dozor

- pomáha pri ohrození života a zdravia fyzických osôb, majetku právnických osôb a fyzických osôb
- vykonáva záchranné práce pri núdzovom odstraňovaní stavieb, ľadových bariér
- zabezpečuje:
 - jednotné uplatňovanie technických požiadaviek protipožiarnej bezpečnosti
 - vykonávanie dohľadu nad výrobkami
 - posudzovania zhody

HaZZ sa podieľa na (11):

- plnení úloh:
 - integrovaného záchranného systému, civilnej ochrany
 - pri poskytovaní predlekárskej pomoci
 - pri príprave na obranu štátu a úloh súvisiacich s mobilizačnými prípravami
 - vedecko-technického rozvoja na úseku ochrany pred požiarimi
- odsune ranených a chorých (pri požiaroch, haváriách, živelných pohromách a iných mimoriadnych udalostiach)
- zabezpečovaní núdzového zásobovania a núdzového ubytovania obyvateľstva
- poskytovaní humanitárnej pomoci
- likvidácii ohnisk nákaz zvierat

HaZZ pri plnení menovaných úloh spolupracuje so štátnymi orgánmi, s orgánmi samosprávy, právnickými osobami, občianskymi združeniami a s fyzickými osobami. Pri riešení otázok na úseku ochrany pred požiarimi HaZZ spolupracuje s príslušnými orgánmi iných štátov a s medzinárodnými organizáciami. HaZZ poskytuje pomoc iným štátom v rozsahu medzinárodných zmlúv, ktorými je Slovenská republika viazaná (11).

2.1.2 Organizačná štruktúra

HaZZ tvorí jedno prezídium, 8 krajských riaditeľstiev, 49 okresných riaditeľstiev, Hasičský a záchranný útvar hlavného mesta Slovenskej republiky Bratislavy, 5 zariadení a ďalšie pracoviská (11).

V práci sa sústreďíme na Žilinský kraj. V rámci kraja sa tu nachádza Krajské riaditeľstvo Hasičského a záchranného zboru v Žiline a 7 okresných riaditeľstiev v okresoch (11):

- Čadca
- Dolný Kubín
- Liptovský Mikuláš
- Martin
- Námestovo
- Ružomberok
- Žilina

2.2 Informačné technológie zboru

• Hardware

Väčšina staníc HaZZ je vybavená stolnými počítačmi - spravidla sa jedná staršie počítače s dvojjadrovými procesormi. Na operačných pracoviskách nájdeme aj modernejšie počítače s lepšími procesormi a novším operačným systémom. Firemné notebooky nie sú príliš bežné vzhľadom na to, že stolné počítače sú na administratívu pre pracovníkov dostačujúce. Môžeme však nájsť aj modernejšie notebooky Lenovo s najnovším operačným systémom a slušnou konfiguráciou – tie sú nevyhnutné v kanceláriách riaditeľov, ktorí sú v rámci vykonávania svojho povolania viac v pohybe.

• Software

Najbežnejším operačným systémom je Windows XP alebo Windows 7 – to sa týka najmä spomínaných stolných počítačov. Na operačných pracoviskách sa využíva Windows 10, čo platí aj pre firemné notebooky. Jedná sa o zakúpenú multilicenciu operačných systémov Windows. Ďalej sú zakúpené kancelárske balíčky Microsoft Office 2013, ktoré sa vo veľkom rozmedzí využívajú v administratívnej sekcii staníc.

• Informačný systém

Zbor využíva intranet Hasičského a Záchranného zboru. Iba z vnútornej siete, kde sa pracovníci dostanú po prihlásení konkrétneho užívateľa, je poskytnutý prístup do nasledujúcich systémov:

- **vlastného informačného systému zboru**
- **Infowebu** (informačný systém Ministerstva vnútra SR)
- Kvalitný ERP systém **SAP** je tu využívaný v plnej miere najmä z ekonomického a účtovného hľadiska – napríklad na dochádzku pracovníkov, mzdy, rozpočty či finančné reporty.

Všetky aktuálne výjazdy sú zaznamenávané do webovej aplikácie **EPOS**. Jedná sa o elektronickú podporu operačného strediska. Táto aplikácia sa už nachádza na verejnej sieti, kde sa pracovníci dostanú prihlásením ako konkrétny užívateľ v pevne danom zozname. Autentizácia prebieha zvolením krajského riaditeľstva, následne zvolením daného okresného riaditeľstva, výberom operačného dôstojníka, ktorý požaduje do aplikácie prístup a zadaním hesla. Webovú aplikáciu môžeme nájsť na adrese <http://www.emergency112.eu/krhazz/> .

- **Sieť a internet**

V celom zbore sa využíva lokálna počítačová sieť (LAN). Táto sieť je vysoko zabezpečená, nakoľko sa jedná o spoločnú sieť s Ministerstvom vnútra. Všetky počítače v sieti sú v doméne intra.minv.sk , každý nový počítač musí byť kvôli bezpečnosti prvý raz zaregistrovaný do domény, až potom je mu umožnený prístup do domény. Pri potenciálnom útoku použitím iného počítača by preto nestačilo pripojiť sa na LAN a následne zadať prihlasovacie údaje.

- **Server**

HaZZ má vlastný server. V rámci okresu je vždy jedno klasické PC, inštalované ako server s bežným operačným systémom Windows XP a so zdieľaním disku. Zbor k fungovaniu vlastných serverov svojho času využíval serverový operačný systém od Novell. Keď však prešli pod Intranet, prestali mať možnosť používať akékoľvek iné operačné systémy.

V rámci krajského riaditeľstva využíva zbor server so systémom Windows Server a kapacitou 2TB.

- **Záloha a archivácia dát**

Server v rámci krajského riaditeľstva sa nachádza v budove samotného krajského riaditeľstva, kde prebieha aj pravidelná záloha a archivácia dát.

- **Webový portál**

Okrem spomínanej webovej aplikácie EPOS, ktorá je však prístupná len pre konkrétnych užívateľov, má HaZZ vlastnú sekciu na webovom portáli Ministerstva vnútra SR:

<http://www.minv.sk/?hasici-zachranari>

The screenshot shows the web portal for fire and rescue services (HaZZ) on the official website of the Ministry of Interior of the Slovak Republic. The page layout includes a top navigation bar, a left sidebar with a menu, a central content area with news, and a right sidebar with additional resources.

Navigation Menu (Left Sidebar):

- Úvodná stránka MV SR
- Hasiči a záchranári
- Úvodná stránka
- Hasičský a záchranný zbor
- Legislativa
- Medzinárodná spolupráca
- Riadenie hasičských jednotiek
- Požiarna prevencia
- Vzdelávanie
- Pracovné príležitosti
- Informačný servis
- Kontakty
- Zariadenia HaZZ
- Krajské riaditeľstvá HaZZ

Main Content Area:

Na cyklickej príprave vo VC HaZZ Lešť sa zúčastnili dobrovoľní hasiči z Trenčianskeho kraja

02. 05. 2016 - Vo Výcvikovom centre Hasičského a záchranného zboru Lešť sa v dňoch 29. a 30. apríla 2016 uskutočnila cyklická príprava dobrovoľných hasičov z obcí Horná Suča, Hôrka nad Váhom, Soblahov, Pobedim, Trenčianska Teplá a Zlatníky. Dobrovoľní hasiči počas prípravy získali výcvik ...
» celá správa

Požiar áut v Žiline na sídlisku Hájk

29. 04. 2016 - Vo štvrtok pred pol štvrtou hodinou ráno boli hasiči zo Žiliny vyslaní na Baničovú...
» celá správa

V Rajeckých Tepliciach sa vo štvrtok stala vážna dopravná nehoda

28. 04. 2016 - Dnes pred pol siedmou ráno boli hasiči vyslaní k dopravnej nehode na ceste č. 1/64 v Rajeckých...
» celá správa

Dajte si do diára: 14. máj/Bratislava/DEŇ HASIČOV

27. 04. 2016 - V sobotu 14. mája 2016 sa na Tyršovom nábreží v Bratislave uskutoční 2. ročník podujatia DEŇ HASIČOV...
» celá správa

Dôležité odkazy

- Hasiči a záchranári v ČR
- CTIF - Medzinárodná asociácia hasičských a záchranných služieb

Dôležité oznamy

Čas zvýšeného nebezpečenstva

VZNIKU POŽIARU

(PDF, 137 kB)
Čas zvýšeného nebezpečenstva vzniku požiaru (PDF, 137 kB)

Kalendár podujatí HaZZ

(PDF, 214 kB)
Kalendár podujatí HaZZ (PDF, 214 kB)

facebook

Hasiči na Facebooku

Najčítanejšie

- Počet akčieschopných hasičských jednotiek sa v budúcnosti rozšíri
- Eurokomisárka Kristalina Georgieva ocenila pripravenosť Slovenska pomáhať pri katastrofách doma aj vo svete

Obr. 15 Webový portál HaZZ (11)

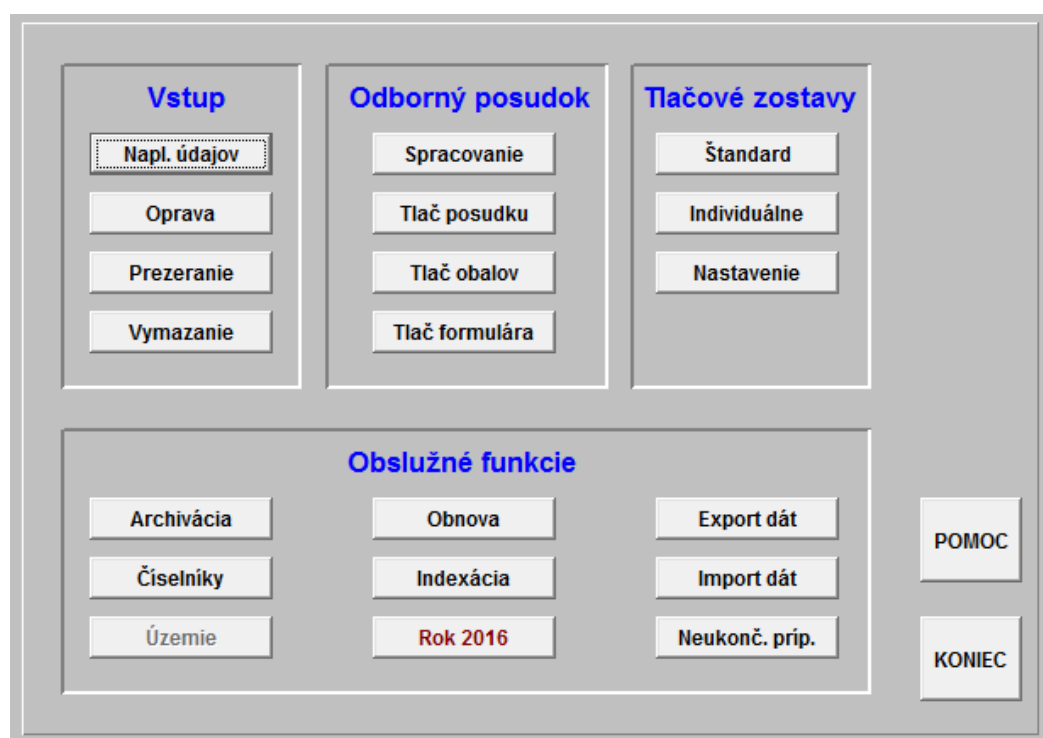
Nachádza sa množstvo užitočných informácií ako najnovšie zásahy zboru, mapy hasičských staníc, informácie o čase zvýšeného nebezpečenstva vzniku požiaru, informačný servis alebo kontakty na všetky zariadenia HaZZ.

2.3 Analýza súčasného systému

V súčasnosti prebieha evidencia požiarov v systéme, ktorý bol vytvorený pomocou nástroja od Microsoftu - Visual FoxPro. Program nazvaný “POŽIAR” využíva súbory s príponou .DBF (DataBase File). Všetky číselníky databázy sú uložené v tomto formáte. Program je využívaný dlhé roky a preto má veľmi základné grafické rozhranie, ktoré bolo používané dokonca ešte na starom 32-bitovom operačnom systéme Windows XP.

2.3.1 Administratíva

Program sa inštaluje lokálne na každý počítač, na ktorom sa ide využívať. Preto je vždy potrebné doň najskôr importovať staršie dáta. Týmto spôsobom majú jednotlivé okresy prístup len ku svojim dátam, zložitejší je však prístup v rámci kraja či celej republiky. V tomto prípade sa zvykli exportovať a zasielať jednotlivé štatistické zostavy.



Obr. 16 Program POŽIAR (Vlastné spracovanie)

Program má veľmi jednoduché užívateľské rozhranie. V obslužných funkciách si môžeme zvoliť, s ktorými dátami budeme manipulovať. Na Obr. 16 je aktuálne zvolený rok 2016.

V prípade, že už nejaké dáta z daného roku máme, je tu možnosť ich do programu importovať. Program ponúka možnosť naplňať nové údaje, rovnako tak ich aj opravovať alebo mazať. Dáta je tiež možné prezerať na základe zvoleného filtra, či už podľa dátumu, výšky škôd či počtu zranených osôb (Obr. 18).

Obr. 17 Naplňanie dát (Vlastné spracovanie)

Obr. 18 Prezeranie údajov (Vlastné spracovanie)

Program ponúka možnosť spracovať a vytlačiť odborný posudok. V ňom podľa evidenčného čísla požiaru používa uložené dáta. Pri tlači je možné zmeniť tlačové zostavy.

Odborný posudok

<u>Ev. č. požiaru</u>	506 13 187	<u>Škoda priama</u>	0
<u>Obec</u>	Kláštôr pod Znievom	<u>Dátum spoz.</u>	09.10.2015
<u>Adresa</u>	k.ú. Slovany	<u>Čas spoz.</u>	12.43
<u>Majiteľ</u>	PD Javorina Valča		
<u>Užívateľ</u>	PD Javorina Valča		
<u>Priestor</u>	pole		
<u>Miesto</u>	BLIŽŠIE NEURČENÉ MIESTO V PRÍRODNOM PROSTREDÍ		
<u>Príčina</u>	vypaľovanie trávy a suchých porastov		
<u>Iniciátor</u>	zápalka; zapaľovač		

Správa o zásahu

Navigation buttons: [Up], [Down], [Find], **Ukonč.**, [Sign], [Max], [Return]

Obr. 19 Tvorba odborného posudku (Vlastné spracovanie)

Program má napriek jednoduchému rozhraniu pomerne prepracovanú databázu plnú číselníkov. Z prvého formulára (Obr. 17) je možnosť pokračovať na detailnejšie údaje o požiari. Každá jedna položka môže byť rozkliknutá do nového okna s číselníkom. Ten v niektorých prípadoch smeruje do ďalšieho číselníku. Týmto spôsobom to funguje najmä pri obsiahlych číselníkoch, ktoré sú delené do kategórií na základe oficiálne priradeného kódu.

Obr. 20 Formulár č. 2 (Vlastné spracovanie)

Na obrázku vyššie (Obr. 20) môžeme vidieť pokračovanie formulára a otvorený číselník s iniciátormi – teda s predmetmi alebo látkami, ktoré spôsobili začiatok požiaru. Ten po zvolení možnosti “Predmety s otvoreným ohňom“ otvoril ďalší číselník, v ktorom môžeme nájsť položku “zápalka; zapaľovač” s oficiálnym kódom 5001, ktorá je už zvolená vo formulári.

2.3.2 Bezpečnosť

Bezpečnosť používaného systému možno hodnotiť z toho pohľadu, že sa nachádza iba na lokálnom počítači, za pomerne vysokú (použitie systému je na druhú stranu veľmi nepraktické). Hrozí v podstate iba napadnutie lokálneho počítača a získanie jeho súborov a dát. Túto možnosť už má doriešenú samotný zbor vlastným zabezpečením - či už prístupovými heslami alebo fyzickými opatreniami, ako je obmedzený fyzický prístup do miestnosti alebo budovy s daným počítačom.

2.3.3 Nedostatky

Systém, ktorý sa v súčasnosti využíva, má množstvo nedostatkov, ktoré sú spôsobené najmä rýchlo sa rozvíjajúcimi informačnými technológiami.

Na to, aby bol systém použiteľný v praxi v rámci celého HaZZ Slovenskej republiky by vyžadoval aspoň pravidelnú úpravu a inováciu.

Medzi najväčšie nedostatky patrí:

1. Lokálna inštalácia programu

Inštalovanie programu, pomocou ktorého prebieha evidencia na každý počítač, je veľmi nepraktickým a pomalým riešením.

2. Lokálne úložisko dát

Zbor nemá žiadny webový portál, dáta nie sú prístupné z iného počítača v akomkoľvek čase a zároveň hrozí strata dát pri fyzickom poškodení počítača.

3. Nekompatibilita s bežným systémom

Niektoré súčasti programu vyžadujú 32-bitový systém, na ktorom fungoval starý Windows XP, avšak novšie počítače sú spravidla s 64-bitovým systémom. Napríklad v prípade problémov s aplikáciou sa program snaží otvoriť pomocníka systému Windows v 32 bitovej verzii.

4. Grafické rozhranie

Grafické rozhranie muselo mať kvôli starším počítačom, na ktorých bol program spúšťaný, čo najnižšie nároky. Teraz to však nie je podmienkou a na novších počítačoch s väčším rozlíšením pôsobí neesteticky.

5. Náročnejšie užívateľské rozhranie, neprehľadnosť

Napriek jednoduchému menu je ovládanie pre bežného užívateľa pomerne náročné a neprehľadné.

6. Číselníky

Pri výbere možností z číselníkov sa treba zbytočne veľa preklikávať (Obr. 20), čo aplikácii taktiež nepridáva na prehľadnosti.

7. Práca s DBF súbormi

Väčšina novších programov súbory s touto príponou nepodporuje.

2.4 Potreba nového riešenia

Zo spomenutých nedostatkov súčasného systému jednoznačne vyplýva, že zbor potrebuje navrhnuť nové riešenie.

Toto riešenie musí rozhodne fungovať aj bez zdĺhavej inštalácie programov na osobné počítače. Zároveň musí mať vyriešenú otázku zálohovania dát.

Novo vytvorený systém bude vyžadovať najmä prepracovanie databázovej schémy. Najlepším riešením pri problémoch s neprístupnosťou dát je navrhnutie webovej aplikácie, ktorou sa k databáze bude pristupovať. Dôraz sa bude musieť klásť na

- Bezpečnosť
- Prehľadnosť
- Jednoduchosť administrácie

Požiadavkou organizácie je tiež zaviesť do systému možnosť nahrať ku záznamom aj fotografie incidentov.

2.4.1 Užívatelia

Je potrebné určiť, ako bude rozdelený prístup do systém. Malo by to záležať od toho, v akej z nasledujúcich úrovní sa užívateľ nachádza:

- **Ministerstvo vnútra**

Užívatelia prístupujúci do systému z Ministerstva vnútra Slovenskej republiky by mali mať možnosť zobrazit' zložky všetkých krajských riaditeľstiev (v našom prípade len krajského riaditeľstva v Žiline) a zároveň všetkých okresných riaditeľstiev.

- **Krajské riaditeľstvo Hasičského a záchranného zboru v Žiline**

Členovia zboru na krajskom riaditeľstve Hasičského a záchranného zboru v Žiline by mali mať prístup ku zložkám krajského riaditeľstva a zároveň všetkých okresných riaditeľstiev. V tomto prípade sú to podobné právomoci ako pre Ministerstvo vnútra, avšak v prípade rozšírenia navrhovaného systému pre iné kraje ako Žilinský by už na rozdiel od Ministerstva vnútra k zložkám krajov prístup nemali.

- **Okresné riaditeľstvá Hasičského a záchranného zboru v rámci Žilinského kraja**

Okresné riaditeľstvá by už mali mať prístup len ku zložkám daných okresov. Tu budú mať možnosť vybrať, ku ktorej obci záznam pripisujú.

Zároveň by sa nemalo zabudnúť na globálneho admina. Ten bude mať prístup k administrácii celej webovej aplikácie a tiež ku správe užívateľov.

2.4.2 Objekty

Pôvodný systém pracuje s veľkým množstvom objektov, o ktorých eviduje informácie. Pre lepšie pochopenie si jednotlivé informácie vysvetlíme. Najdôležitejšie objekty, ktoré musí nový systém zahrnúť, sú teda nasledovné:

Názov objektu: Incident

Popis: Objekt incident je ten najväčší objekt, s ktorým sa v databáze pracuje. Jedná sa vlastne o najdôležitejšie informácie, ktoré sa pri požiari stali. Tieto položky databázy budú povinné k vyplneniu.

Informácie o objekte:

- Evidenčné číslo požiaru – minimálne 6-miestne číslo, ktoré sa skladá z kódu okresu, aktuálneho roku evidencie a Id daného incidentu
- Deň ohlásenia požiaru
- Čas ohlásenia požiaru
- Deň spozorovania požiaru
- Čas spozorovania požiaru
- Obec – obec, pod ktorú zasiahnutý objekt spadá (výber z obcí k príslušnému okresu a kraju)
- Adresa – adresa objektu zasiahnutého požiarom
- Majiteľ – súčasný majiteľ objektu zasiahnutého požiarom
- Užívateľ – súčasný užívateľ objektu zasiahnutého požiarom
- Škoda
 - priama škoda – výška škody spôsobenej priamo pri požiari, vyčíslená v eurách
 - následná škoda – výška škody spôsobená následkami požiaru, teda pravdepodobný odhad sumy, ktorá mohla byť následným ziskom z činnosti, avšak pôsobením požiaru sa o tento zisk prišlo (vyčíslené v eurách, často sa neuvádza práve pre obťažnosť túto sumu odhadnúť správne)
 - uchránené hodnoty - výška škody, ktorá mohla byť spôsobená šírením požiaru, avšak požiarom zákrokom bol objekt od tejto škody uchránený, vyčíslená v eurách

- Počet usmrtených – počet ľudí, ktorý boli pri požiari usmrtení, odvíja sa od ďalšieho objektu “Zranenia“
- Počet ťažko zranených - počet ľudí, ktorý pri požiari utrpeli závažné zranenia, odvíja sa od ďalšieho objektu “Zranenia“
- Počet ľahko zranených - počet ľudí, ktorý pri požiari utrpeli ľahšie zranenia neohrozujúce život, odvíja sa od ďalšieho objektu “Zranenia“
- Poist'ovňa – príslušná poist'ovňa, v ktorej je zasiahnutý objekt poistený – možnosť nezvoliť žiadnu, jednu alebo viaceré poist'ovne
- Vlastníctvo – druh vlastníctva objektu zasiahnutého požiarom
- Určenie škody – spôsob, akým bola určená škoda (napríklad odborným odhadom, poškodeným, prípadne škoda neurčená)
- Charakter škôd (či sa jedná napríklad o budovy, časť budov, stroje, lesy, dopravné prostriedky apod.)
- Odvetvie ekonomickej činnosti – odvetvie, v ktorom sa nachádza objekt zasiahnutý požiarom

Názov objektu: Zranenia

Popis: Tabuľka s evidovanými zraneniami, ktoré pri incidente nastali. K jednému incidentu môže byť evidované jedno, viacero alebo žiadne zranenie. Vyplnenie daných informácií preto samo o sebe nie je povinné, avšak v prípade, že zranenie nastalo, je potrebné už vyplniť informácie o objekte uvedené nižšie.

Informácie o objekte:

- Evidenčné číslo požiaru – prepojené s evidenčným číslom objektu Incident
- Dôvod evidencie – druh zranenia spôsobeného požiarom (ľahké, vážne alebo smrteľné zranenie)
- Kategória – výber kategórie, do ktorej spadá zranená osoba (rozdelené podľa vekových hraníc, prípadne či sa jedná o hasiča, ktorý bol v službe pri požiarom zásahu)
- Okolnosti zranenia
- Príčiny zranenia (najčastejšími príčinami sú podľa štatistík otrava alebo priotrávenie splodinami spôsobenými horením alebo pôsobenie vysokých teplôt pri požiari)
- Poznámky k požiaru – dodatočné informácie o zranení

Názov objektu: Detaily incidentu

Popis: Jedná sa o objekt, ktorý nie je povinné vyplniť. Obsahuje veľké množstvo okolností požiaru a tvorí ho až 15 číselníkov. Každý jeden incident má práve jedny detaily incidentu.

Informácie o objekte:

- Priestor vzniku požiaru – určuje, či bol pri požiari poškodený nejaký objekt alebo prostredie
- Miesto vzniku požiaru – oficiálna kategorizácia miest vzniku požiarov
- Dopravný prostriedok – časť dopravného prostriedku alebo pracovného stroja, ktoré boli zasiahnuté požiarom
- Dopravníkové zariadenia – druh dopravníku v prípade, že bolo požiarom zasiahnuté dopravníkové zariadenie
- Príčina vzniku požiaru (či požiar vznikol úmyselne alebo nie, akým spôsobom a či vôbec bola príčina zistená)
- Manipulácia s horľavými a výbušnými látkami (ak nejaká prebehla, zodpovedá otázku kde alebo pri čom sa táto manipulácia uskutočnila)
- Iniciátor – pôvodca požiaru (napríklad rozvody elektrickej energie, spotrebiče, dopravné prostriedky apod.)
- Elektrické obvody – ak boli iniciátormi elektrické obvody, ponúknutý je výber jednotlivých častí rozvodov, ktoré napomohli vzniku požiaru
- Pôsobenie iniciátora – spôsob, akým iniciátor spôsobil požiar
- Prvé horenie – látky či materiály, ktoré začali horieť ako prvé
- Následné horenie – látky či materiály, ktoré určovali následný rozvoj požiaru
- Trieda nebezpečnosti horľavej kvapaliny – v prípade, že sa jednalo o horľavé kvapaliny, treba určiť triedu nebezpečnosti (trieda 1 až 4 alebo kvapaliny mimo triedy nebezpečnosti)
- Nedostatky v organizácii – v podstate sa jedná o bezpečnostné chyby, ktoré sa vyskytli v organizácii, v ktorej vznikol požiar. Tieto nedostatky ovplyvnili vznik daného požiaru
- Nedostatky v činnostiach – chyby, ktoré sa vyskytli pri vzniku požiaru a ovplyvnili ďalšie šírenie požiaru

- Uzatvorenie prípadu – udáva, akým subjektom bol riešený incident uzavretý a aký verdikt bol uznesený

Názov objektu: Poškodený objekt

Popis: V prípade, že sa v detailoch incidentu uvedie priestor vzniku požiaru, systém umožňuje spracovať o danom poškodenom objekte ďalšie informácie. Jedná sa o detailnejšie upresnenie objektu a spôsobených škôd. Vyskytujú sa tu najmä informácie, ktoré sú relevantné vtedy, keď poškodeným objektom bola budova.

Informácie o objekte:

- Stupeň poškodenia budov – v prípade, že sa jednalo o poškodenie budovy, určuje stupeň poškodenia
- Plocha/objem/hmotnosť – rozsah poškodeného objektu vyjadrený fyzikálnymi jednotkami ár , m^2 , m^3 alebo q (plocha v ároch alebo metroch štvorcových, objem v metroch kubických, hmotnosť v metrických centoch)
- Štátny požiarly dozor – informácia, či je nad objektom tento dozor vykonávaný
- Funkcia požiarneho úseku – o aký priestor sa jedná (celá budova, bytová jednotka, výber jednotlivých častí či iné priestory)
- Výška požiarneho úseku – podlažia budov
- Požiarla odolnosť stavebnej konštrukcie
- Horľavosť stavebných konštrukcií
- Požiarla bezpečnosť uzáverov – jedná sa o uzávery ako požiarne dvere, požiarne klapky a stupeň ich bezpečnosti
- Požiarla odolnosť uzáverov – napríklad odolnosť protipožiarlych dverí na priestup teplôt, celistvosť a radiáciu
- Dôvod rozšírenia požiaru do ďalších úsekov
- Elektrická požiarla signalizácia – či bola nejaká nainštalovaná, či bola funkčná a v činnosti
- Stabilné/polostabilné hasiace zariadenia – či boli inštalované, funkčné, v činnosti a akým spôsobom boli ovládané – ručne či automaticky
- Hasiace látky v stabilnom hasiacom zariadení – látky, ktoré sa nachádzali v použitom hasiacom zariadení

2.5 SWOT analýza súčasného systému

Tabuľka 1 SWOT analýza súčasného systému (Vlastné spracovanie)

S	
SILNÉ STRÁNKY (STRENGTHS)	
• zaužívaný systém • jednoduché používanie	

W	
SLABÉ STRÁNKY (WEAKNESSES)	
• nepraktické inštalovanie • nekompatibilita s novším OS • grafické rozhranie • bez zálohy dát	

O	
PRÍLEŽITOSTI (OPPORTUNITIES)	
• aktualizácia/vytvorenie nového systému • sprístupnenie prostredníctvom webu • pridanie funkcií • rozšírenie pre všetky zbory SR	

T	
HROZBY (THREATS)	
• strata dát • úplná strata kompatibility (nové OS)	

2.6 Zhodnotenie pripravenosti na nové riešenie

Databáza a webová aplikácia pobežia na vlastnom serveri zboru, preto nie sú na **vybavenie hardvéru** užívateľov kladené vysoké nároky. Systém sa bude ovládať prostredníctvom webovej aplikácie za použitia ľubovoľného internetového prehliadača. Ten nechýba na žiadnom počítači firmy, takže nebude problém aplikáciu rozbehnúť. Jediné, čo vyžaduje, je internetové pripojenie. Každá stanica zboru má vyriešené pripojenie prostredníctvom lokálnej počítačovej siete.

Požiadavky na software zbor spĺňa. Na obsluhu systému stačí v podstate akýkoľvek operačný systém s modernejším prehliadačom. Ten by mal zvládať HTML5, CSS a JavaScript, pomocou ktorých bude celý systém vytvorený. Prakticky každý počítač má takýto prehliadač už nainštalovaný.

Nároky na **zručnosť užívateľov v oblasti IT** rozhodne nebudú vysoké. Systém bude vyžadovať iba prihlásenie prostredníctvom webovej aplikácie, za ktorým ho bude čakať prehľadné a jednoduché užívateľské rozhranie. Po kratšej analýze zamestnancov zboru bolo zistené, že sa v zbore nachádzajú zruční pracovníci, ktorí by zvládli aj náročnejšie úlohy v oblasti informačných technológií, preto by to nemal byť v tomto smere žiadny problém.

Vzhľadom k tomu, že je potrebný niekto, kto by pridával nových užívateľov a spracúval tých súčasných, prípadne dohliadal na databázu a prevádzku celej webovej aplikácie, bude sa vyžadovať aspoň jeden pracovník so vzdelaním v oblasti IT technológií, ktorý by dostal pridelenú rolu **globálneho administrátora**. Ten by mal mať skúsenosti s administráciou v phpMyAdmin, kde by spravoval jednotlivé skupiny používateľov a pridával nových užívateľov. Zároveň by mal ovládať aspoň základy programovacích jazykov PHP alebo HTML, CSS a Javascriptu v prípade, že by sa vyskytli nejaké problémy alebo sa zmenili požiadavky vedenia zboru na nový systém. Tieto požiadavky by spĺňal zamestnanec, ktorý mal na starosti súčasný systém.

Analýza všetkých požiadaviek na zavedenie nového systému viedla k záverom, že **zbor je pripravený na implementáciu nového systému** a nič by nemalo zabrániť tomu, aby správne fungoval.

3 Vlastný návrh riešenia

Táto kapitola bude venovaná samotnému návrhu a vytvoreniu finálneho riešenia, teda systému pre evidenciu požiarov.

3.1 Návrh databázy

Prvým krokom bude práca na databáze, ktorá je najdôležitejšou časťou celého systému. Aby mohol systém naplno fungovať, musí byť databáza navrhnutá správne a logicky. Databáza sa môže meniť aj dodatočne, po vytvorení či po uvedení do prevádzky, kedy sa už naplnila aj dátami. Tieto zmeny však vyžadujú veľmi veľa práce a veľmi ľahko sa môže stať, že sa pri zmene štruktúry vyskytne nejaká chyba, ktorá môže skutočne ohroziť už naplnené dáta, v horšom prípade spôsobiť ich stratu. Pokiaľ sa dáta nezálohovali, možnosť takéhoto nebezpečenstva je ešte pravdepodobnejšia a takáto hrozba musí byť v štátnom sektore vylúčená. Databáza preto musí mať možnosť zálohy. Okrem toho vyžaduje, aby bola dostatočne zabezpečená voči neoprávnenému prístupu a manipulácii.

Zbor nevyžadoval do nového systému import starších údajov, ale požaduje úplne nové riešenie s naplnením nových údajov. Databáza by však mala obsahovať všetky číselníky, ktoré používali v starej aplikácii. To sú poväčšine najmä obsiahle tabuľky s oficiálnymi kódmi a názvami skutočností, ktoré sa môžu pri požiaroch stať.

Systém bude evidovať nasledovné objekty:

- Incident
- Zranenia
- Detail incidentu
- Poškodený objekt

3.1.1 Tvorba tabuliek

Informácie o jednotlivých objektoch je potrebné spracovať do tabuliek a tie pomenovať anglickými výrazmi. Tento krok je dôležitý najmä z dôvodu zjednodušenia práce s premennými, ktorú framework umožňuje. Premenné a ich jednotné či množné číslo prehľadne rozoznáva. Zjednoduší to aj migráciu databázy do webovej

aplikácie. Informácie o objektoch som sa snažila pomenovať anglicky čo najpresnejšie, nie vždy však existuje pre daný výraz alebo slovné spojenie správny preklad. Zároveň som sa snažila o čo najkratšie pomenovanie pre praktickejšie narábanie s premennými.

V zozname nižšie teda uvádzam, v akej tabuľke sa bude nachádzať objekt a ako budú pomenované jednotlivé záznamy o objekte. V prípade, že sa jedná o číselníky, bude nutné vytvoriť pre záznamy ďalšie tabuľky. Názvy budú nasledovné:

Názov objektu: Incident – tabuľka incidents

- Evidenčné číslo požiaru – evidence_number
- Deň a čas ohlásenia požiaru – report_date
- Deň a čas spozorovania požiaru – observe_date
- Obec – tabuľka towns
- Adresa – address
- Majiteľ – tabuľka ownerships
- Užívateľ – tabuľka property_owners
- Škoda
 - priama škoda – direct_damage_value
 - následná škoda – followup_damage_value
 - uchránené hodnoty – saved_value
- Poistovňa – tabuľka insurance_companies
- Vlastníctvo – tabuľka ownerships
- Určenie škody – tabuľka damage_specifications
- Charakter škôd – tabuľka damage_types
- Odvetvie ekonomickej činnosti – tabuľka industry_types

Názov objektu: Zranenia – tabuľka injuries

- Evidenčné číslo požiaru – evidence_number
- Dôvod evidencie – tabuľka injury_types
- Kategória – tabuľka injury_categories
- Okolnosti zranenia – tabuľka injury_circumstances

- Príčiny zranenia – tabuľka injury_causes

Názov objektu: Detaily incidentu – tabuľka incident_details

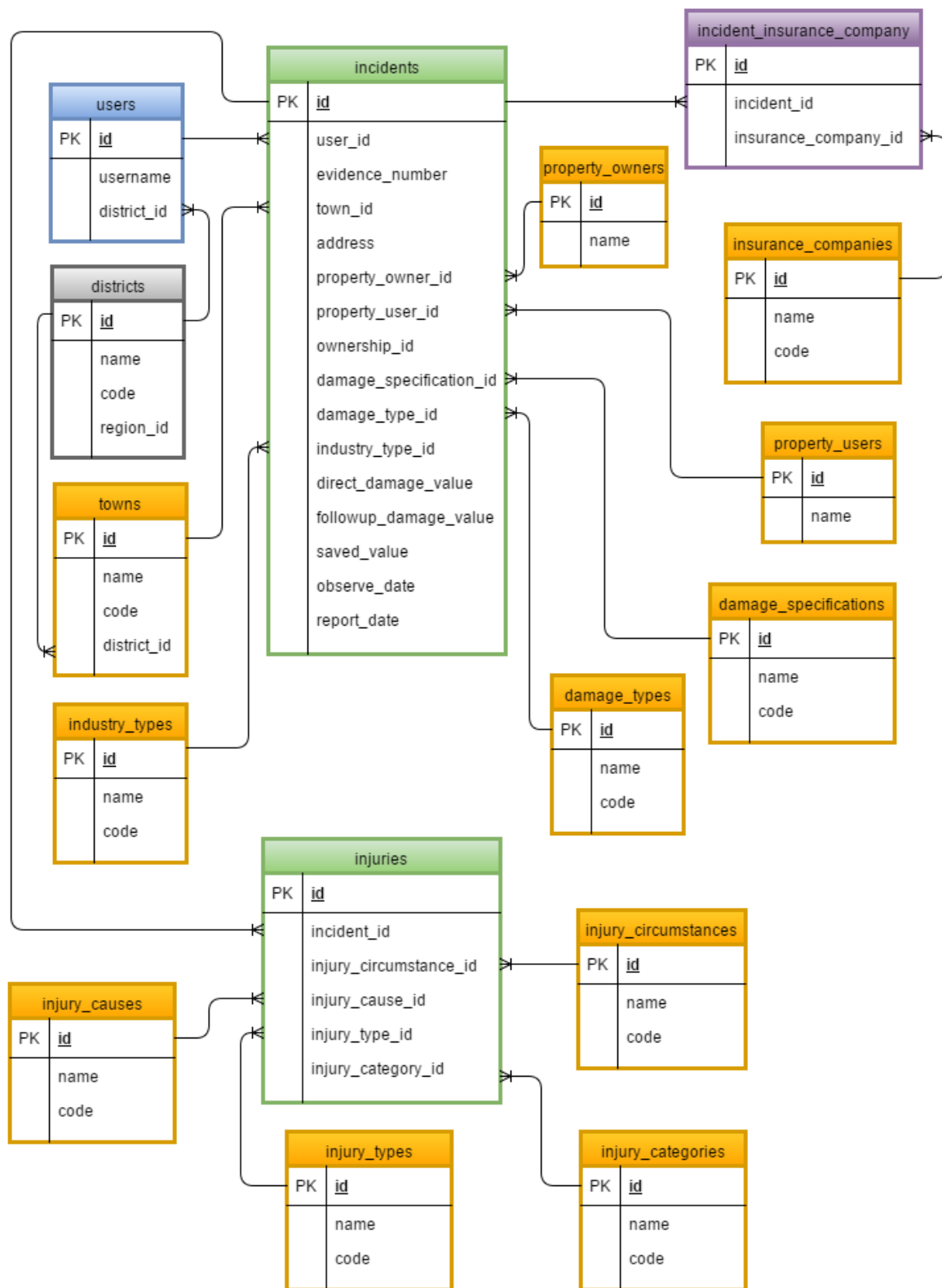
- Priestor vzniku požiaru – tabuľka areas
- Miesto vzniku požiaru – tabuľka fire_locations
- Dopravný prostriedok – tabuľka vehicle_parts
- Dopravníkové zariadenia – tabuľka conveyor equipments
- Príčina vzniku požiaru – tabuľka incident_causes
- Manipulácia s horľavými a výbušnými látkami – tabuľka flammable_instances
- Iniciátor – tabuľka initiators
- Elektrické obvody – tabuľka electrical_wirings
- Pôsobenie iniciátora – tabuľka initiators_impacts
- Prvé horenie – tabuľka burning_substances
- Následné horenie – tabuľka following_substances
- Trieda nebezpečnosti horľavej kvapaliny – tabuľka hazard_classes
- Nedostatky v organizácii – tabuľka organizations_shortcomings
- Nedostatky v činnostiach – tabuľka action_shortcomings
- Uzatvorenie prípadu – tabuľka incident_conclusions

Názov objektu: Poškodený objekt – tabuľka damaged_objects

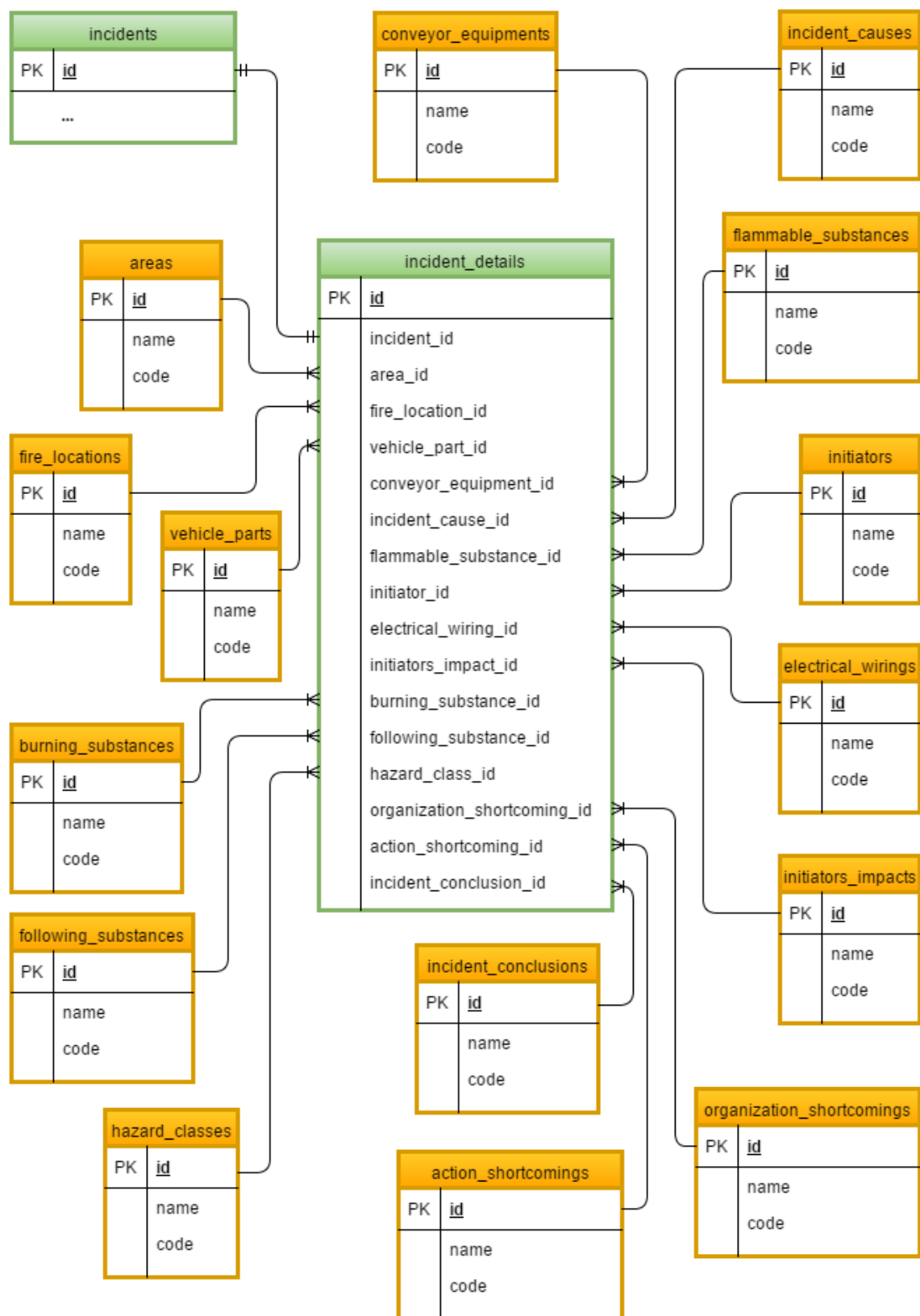
- Stupeň poškodenia budov – tabuľka damage_degrees
- Plocha/objem/hmotnosť – dimension_value
- Funkcia požiarneho úseku – tabuľka segment_functions
- Výška požiarneho úseku – tabuľka segment_altitudes
- Požiarna odolnosť stavebnej konštrukcie – tabuľka fire_resistances
- Horľavosť stavebných konštrukcií – tabuľka flammability_of_objects
- Požiarna bezpečnosť uzáverov – tabuľka fire_shutters
- Požiarna odolnosť uzáverov – tabuľka shutter_resistances
- Dôvod rozšírenia požiaru do ďalších úsekov – tabuľka spreading_causes
- Elektrická požiarna signalizácia – tabuľka fire_alarms
- Stabilné/polo-stabilné hasiace zariadenia – tabuľka fire_extinguishers
- Hasiace látky v stabilnom hasiacom zariadení – tabuľka chemicals

3.1.2 Databázová schéma

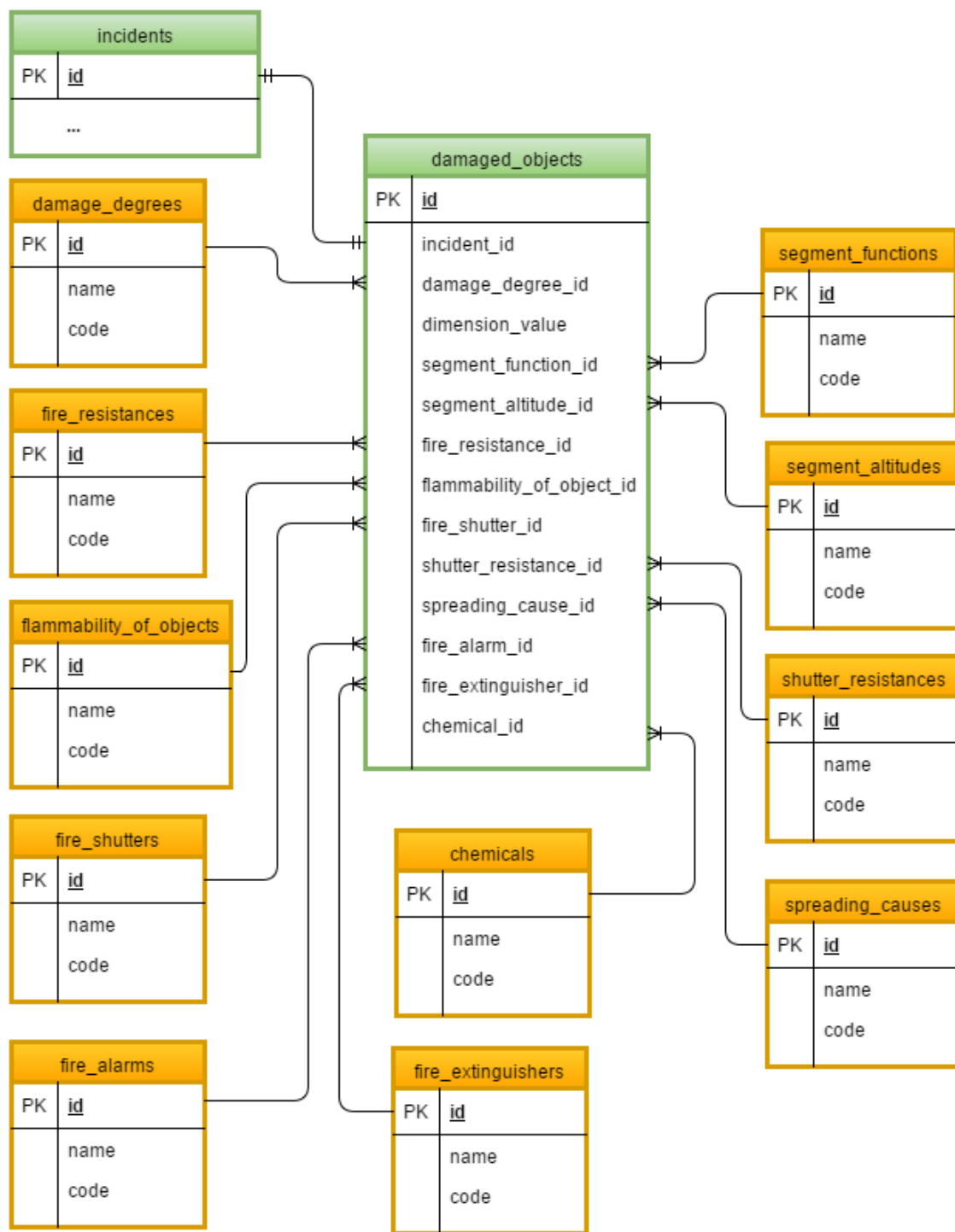
Databázová schéma po prepracovaní vyzerá nasledovne:



Obr. 21 Databázová schéma systému 1 (Vlastné spracovanie)



Obr. 22 Databázová schéma systému 2 (Vlastné spracovanie)



Obr. 23 Databázová schéma systému 3 (Vlastné spracovanie)

Obrázky sú rozdelené na 3 časti, avšak všetky zelené tabuľky s objektami sú relačne napojené na tabuľku **incidents**, ktorá je tou najdôležitejšou tabuľkou v schéme. Všetky hlavné objekty (zelené tabuľky) sú v databázovej schéme s ostatnými tabuľkami vo vzťahu 1:N.

V prípade tabuliek **incidents** a **insurance companies**, teda tabuľky o incidentoch a tabuľky s poisťovňami ide o vzťah N:M. Jeden incident totiž môže byť spätý s viacerými poisťovňami a zároveň jedna poisťovňa môže byť spätá s viacerými incidentmi. Z toho dôvodu bola vytvorená tabuľka **incident_insurance_company**, ktorá už je medzi týmito dvoma tabuľkami vo vzťahu 1:N.

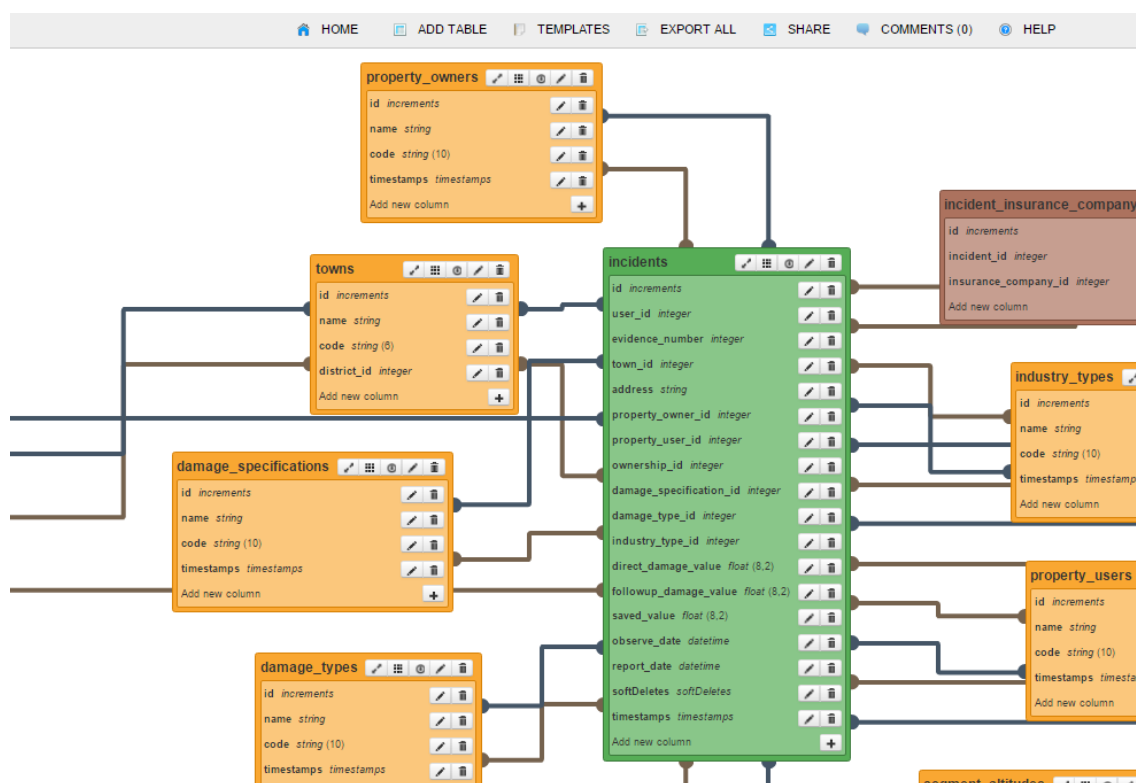
Tabuľky **incidents** a **injuries** majú taktiež definovaný vzťah 1:N, pretože jeden incident môže zahŕňať viacero zranení.

Medzi tabuľkou **incidents** a **incidents_details** je vzťah 1:1 z toho dôvodu, že jeden incident má práve jeden detail incidentu a zároveň jeden detail incidentu náleží ku jednému incidentu. To isté platí aj pre tabuľku **damaged_objects**.

Celá databázová schéma sa nachádza v prílohe.

3.1.3 Databázová schéma Laravelu

Aby bolo vytvorenie databázy čo najjednoduchšie, framework Laravel ponúka cez webové rozhranie takzvaný Laravel Schema Designer, kde sa veľmi jednoducho a rýchlo pracuje s tabuľkami a vzťahmi. Najväčšou výhodou je však možnosť exportu databázovej schémy, a to v zazipovanom súbore, ktorý obsahuje súbory jednotlivých tabuliek pripravené pre migráciu do frameworku Laravel. Pre lepšiu prehľadnosť vzťahov som celú schému spracovala v inom programe (Obr. 21 až Obr. 23 a príloha č. 1).



Obr. 24 Laravel Schema Designer (12)

3.1.4 Import a export číselníkov

Číselníky, ktoré využíval starý systém a majú sa používať aj v novom, boli poskytnuté zborom v súboroch s príponou *.DBF*. Originálne bol tento formát používaný v dBase II. Nový systém bude potrebovať tieto číselníky importovať do phpMyAdmin.

V prvom rade je potrebné súbory dostať do použiteľného formátu. Zatiaľ čo FoxPro s *.DBF* súbormi dokáže pracovať, na import potrebujeme zvoliť iný. Tým je súbor s príponou *.CSV*, ktorý pozostáva z riadkov. V tých sú jednotlivé položky oddelené určitým zvoleným znakom. Najčastejšie to býva čiarka alebo bodkočiarka, v mojom prípade som zvolila bodkočiarku, s ktorou framework nemá problém narábať.

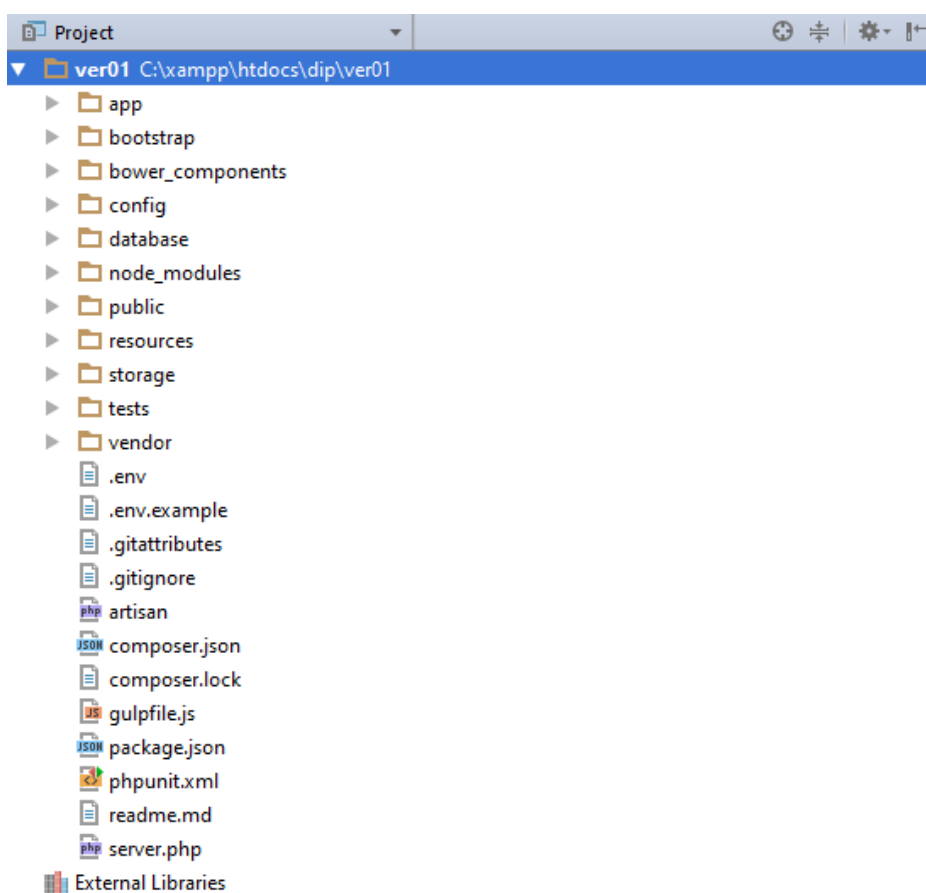
Číselníky bolo potrebné exportovať (ja som použila program DBFPlus) a rozdeliť ich na logickejšie časti, pretože obsahovali príliš veľa informácií naraz a ich filtrovanie by zbytočne pridávalo prácu. Aby sme číselníky dostali do novej databázy, budeme potrebovať nainštalovať PHP framework.

3.2 Laravel framework

Od tejto časti bude návrh systému prebiehať za pomoci PHP frameworku **Laravel**, ktorý značne uľahčí vytvorenie celej webovej aplikácie.

Návrh systému prebiehal na lokálnom počítači s použitím ovládacieho panelu XAMPP a prebiehajúcimi službami Apache a MySQL.

Na obrázku nižšie (Obr. 25) môžeme vidieť úspešne nainštalovaný projekt so základnou štruktúrou frameworku, ktorú v podstate nebudeme vôbec meniť. Bude sa len postupne naplňovať novými súbormi.



Obr. 25 Štruktúra priečinkov projektu (Vlastné spracovanie)

3.3 Návrh webovej aplikácie

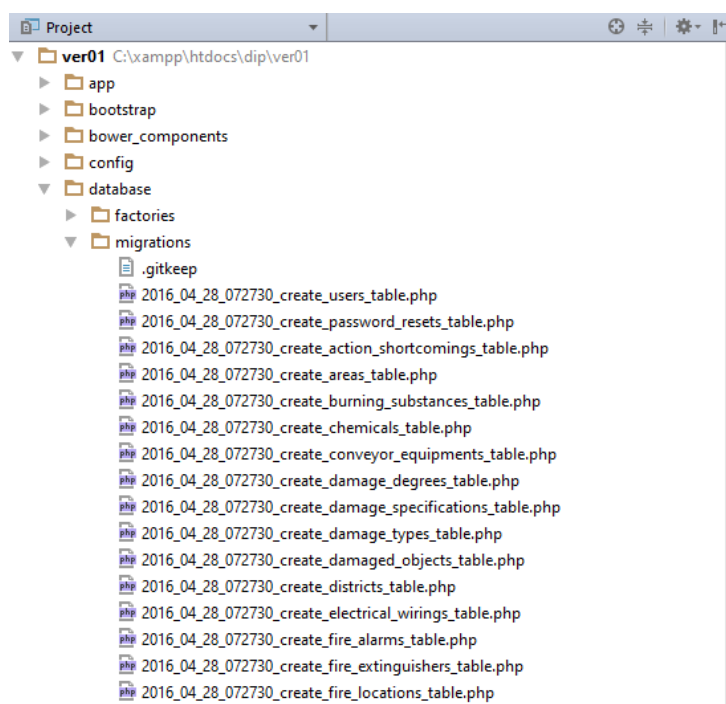
Pri návrhu webovej aplikácie treba začať najdôležitejším - databázou, ktorú sme si vopred pripravili v kapitole 3.1. Všetky tabuľky a vzťahy sú pripravené s online nástrojom Laravel Schema Designer, kde je možnosť ich spoločne vyexportovať. Pri tvorbe aplikácie budeme pracovať s konzolou **Artisan**. Jedná sa o rozhranie s príkazovým riadkom, ktoré je súčasťou frameworku Laravel. Ponúka množstvo príkazov, ktoré budú pri vytváraní aplikácie užitočné a ku každému príkazu je možné zobrazit' nápovedu k používaniu.

3.3.1 Migrácia databázy

Export databázovej schémy má za výsledok zazipovaný súbor. Ten obsahuje súbory s migráciami tabuliek. Migrácie nám umožnia veľmi rýchlo a jednoducho modifikovať našu databázu.

3.3.1.1 Migrácie

Vyexportované súbory bolo potrebné presunúť do správneho priečinku projektu (Obr. 26). V našom prípade to je pomerne veľké množstvo súborov, najmä kvôli číselníkom.



Obr. 26 Migrácie (Vlastné spracovanie)

V prípade, že sú potrebné ďalšie migrácie, s použitím **php Artisan** môžeme do príkazového riadku zadať príkaz na vytvorenie novej migrácie. Napríklad pre tabuľku s užívateľmi:

```
php artisan make:migration create_users_table
```

Vygenerované súbory migrácií obsahujú v sebe zoznam knižníc, ktoré používajú. Ďalej sú tu dve metódy - **up** and **down**. V metóde up budeme vykonávať požadované zmeny v tabuľkách databázy. V metóde down ich naopak zvrátime. Na nasledujúcom obrázku (Obr. 27) vytvárame tabuľku **Areas** – tá bude obsahovať zoznam možných priestorov, ktoré zachvátil požiar. Stĺpce tabuľky budú tvorené vygenerovaným ID priestoru, jeho názvom, kódom (ktorý bol súčasťou poskytnutých číselníkov) a časovými razítkami pre prehľad, kedy bol záznam vytvorený a modifikovaný.

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;

class CreateAreasTable extends Migration {

    public function up()
    {
        Schema::create('areas', function(Blueprint $table) {
            $table->increments('id');
            $table->string('name');
            $table->string('code', 10)->index();
            $table->timestamps();
        });
    }

    public function down()
    {
        Schema::drop('areas');
    }

}
```

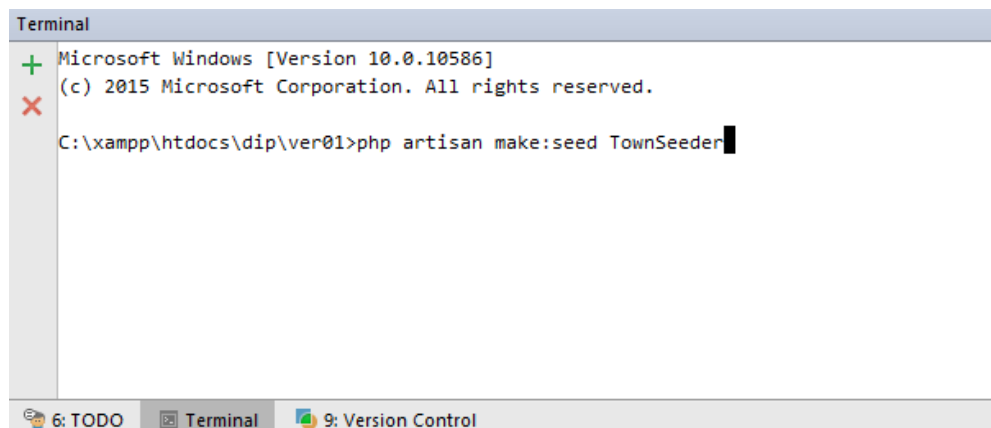
Obr. 27 Súbor migrácie pre tabuľku Areas (Vlastné spracovanie)

3.3.1.2 Seeders

Laravel ponúka jednoduchú metódu pre seedovanie databázy s použitím seedovacích tried. Exportovaním z pôvodnej databázy sme dostali CSV súbory, s ktorými budeme ďalej pracovať v takzvaných **seederoch**. Jedná sa o súbory na seedovanie databázy, ktoré

vytvoríme prostredníctvom konzoly Artisan. Vďaka týmto súborom budeme môcť databázové tabuľky kedykoľvek premazať a zase naplniť.

Postupne vytvoríme seedery pre všetky tabuľky nasledujúc príklad, kde vytvárame seeder pre tabuľku s obcami:



```
Terminal
+ Microsoft Windows [Version 10.0.10586]
(c) 2015 Microsoft Corporation. All rights reserved.
C:\xampp\htdocs\dip\ver01>php artisan make:seed TownSeeder
```

Obr. 28 Vytváranie seederu pre tabuľku s obcami (Vlastné spracovanie)

Na obrázku nižšie (Obr. 29) máme možnosť vidieť seeder pre tabuľku obcí. Vytvoríme tu metódu, ktorá spustí seedovanie databázy. Do premennej **\$csv** uložíme hodnotu metódy **array_map()**, ktorá obsahuje dva parametre:

- Funkciu **str_getcsv** - parsuje reťazec CSV do poľa
- Cestu pre CSV súbor s obcami (**Towns.csv**)

Daná metóda zavolá funkciu na každý element poľa CSV súboru. Nasleduje jednoduchý cyklus **foreach**, ktorý pre každý riadok tabuľky zoberie riadok z csv súboru a postupne vyplní údaje: meno obce, kód obce a ID okresu.

```

<?php
use ...

class TownSeeder extends Seeder
{
    /**
     * Run the database seeds.
     *
     * @return void
     */
    public function run()
    {
        $csv = array_map('str_getcsv', file(database_path('seeds/csvFiles/Towns.csv')));

        foreach ($csv as $row) {
            Town::create([
                'name' => $row[0],
                'code' => $row[1],
                'district_id' => $row[2]
            ]);
        }
    }
}

```

Obr. 29 Seeder pre tabuľku s obcami (Vlastné spracovanie)

Týmto spôsobom vytvoríme všetky seeder triedy. Aby sme potom konečne spustili proces seedovania pre našu databázu, skrz konzolu Artisan použijeme príkaz:

php artisan db:seed

Pokiaľ chcem špecifikovať, ktorú triedu seedovať, použijem nasledovný príkaz:

php artisan db:seed --class =“TownSeeder“

Tento krok som vykonala v neskoršej fázi projektu, keď som dodatočne pridávala tabuľky a nebolo potrebné seedovať všetko.

3.3.2 Routing a controllers

Aby všetky formuláre zasielali svoje údaje na správne miesto, potrebujeme im definovať cesty (routes). Na väčšinu použijeme súbor **routes.php**. Aby však súbor nebol príliš dlhý a neobsahoval toho moc, vytvoríme pre jednotlivé objekty triedy **Controllers** a tu zorganizujeme všetky požiadavky. Základná štruktúra vyzerá takto:

```

\App\Http\Controllers\IncidentDetailController
<?php

namespace App\Http\Controllers;

use ...

class IncidentDetailController extends Controller
{
    public function index()
    {
        //
    }

    public function create()
    {
    }

    public function store(Request $request)
    {
        //
    }

    public function show($id)
    {
        //
    }

    public function edit($id)
    {
        //
    }

    public function update(Request $request, $id)
    {
        //
    }

    public function destroy($id)
    {
        //
    }
}

```

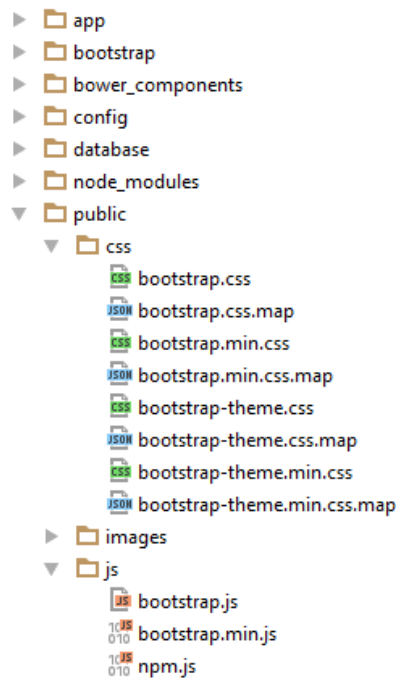
Obr. 30 Základná štruktúra controlleru (Vlastné spracovanie)

Zobrazené metódy budú slúžiť pri práci s formulármi, *create* a *store* pri vytváraní nového záznamu. *Show* pri jeho zobrazení, *edit* a *update* pri úprave záznamu a *destroy* pri jeho mazaní.

3.3.3 Bootstrap

Inštaláciou CSS frameworku Bootstrap nám pribudnú do public priečinku projektu (tak ako na Obr. 31):

- CSS súbory
- Minimalizované CSS súbory
- CSS mapy
- Javascript súbor
- Minimalizovaný Javascript súbor



Obr. 31 Štruktúra projektu po inštalácii Bootstrap (Vlastné spracovanie)

Vďaka kódu z dokumentácie Bootstrap frameworku si vytvoríme základnú HTML šablónu. Stránka frameworku ponúka CSS a Javascript kódy s príkladmi použitia, všetko veľmi prehľadne prezentované. Tieto kódy stačí jednoducho vkladať do HTML kódu.

3.4 Autentifikácia

V rámci zabezpečenia prebehne hneď pri otvorení stránky autentifikácia. Užívatelia budú musieť pre prístup do systému vyplniť prihlasovacie údaje. Tými bude e-mail a heslo. Spravovať tieto údaje bude globálny administrátor prostredníctvom databázy.

Laravel framework robí z celého procesu prihlasovania veľmi jednoduchú vec. Jeho súčasťou je konfiguračný súbor **auth.php**, kde stačí upraviť pár nastavení a o zvyšok sa framework postará.

Dôležitým krokom je použitie middleware, ktorý overí, či je aktuálne užívateľ prihlásený do systému a to ešte predtým, ako mu dovolí prístup na požadovanú stránku.

```

Route::group([], function(){
    Route::get('/', [
        'as' => 'dashboard',
        'middleware' => 'auth',
        'uses' => 'DashboardController@index'
    ]);

    Route::group([
        'as' => 'incident.',
        'middleware' => 'auth',
        'prefix' => 'incident',

    ], function(){

        Route::get('/create', [
            'as' => 'create',
            'uses' => 'IncidentController@create'
        ]);
        Route::post('/create', 'IncidentController@store');
    });
}

```

Obr. 32 Použitie middleware na autentifikáciu (Vlastné spracovanie)

Na obrázku vyššie (Obr. 32) máme možnosť vidieť použitie middleware pre autentifikáciu **auth**. Dashboard bude predstavovať hlavnú stránku webovej aplikácie, incident bude formulár so základnými údajmi o požiari. Tu sa využívajú aj spomínané controllers, v ktorých sú bližšie definované cesty pre hlavnú stránku či budúci formulár pre vytvorenie záznamu o incidente. Autentifikáciu v prípade tvorenia incidentu už netreba špecifikovať, pretože je priamo naviazaná na samotný incident, pri ktorom overenie prebehne.

3.4.1 Užívatelia

Ako som spomínala, užívateľom bude globálny admin a ďalej tri typy užívateľov podľa toho, z akej úrovne pristupujú do systému:

1. Ministerstvo vnútra
2. Krajské riaditeľstvo Hasičského a záchranného zboru v Žiline
3. Okresné riaditeľstvá Hasičského a záchranného zboru v rámci Žilinského kraja

Pre užívateľov bola vytvorená špeciálna tabuľka s názvom **user_groups**, pomocou ktorej užívateľom priradíme skupinu a rozdelíme právomoci.

Každá skupina užívateľov má svoje ID nasledovne:

- 1 - Administrátor**
- 2 - Krajské riaditeľstvo**
- 3 - Okresné riaditeľstvo**
- 4 - Ministerstvo vnútra**

Podľa toho, aké ID bude priradené užívateľovi sa bude odvíjať jeho prístup k dátam. Administrátor bude mať sprístupnené všetko vrátane administrátorského rozhrania, užívateľ v skupine “Krajské riaditeľstvo“ už nie. Ten však bude môcť vidieť a editovať všetky záznamy. Skupina “Okresné riaditeľstvo“ bude môcť prehliadať a editovať už len záznamy v rámci daného okresu užívateľa. Ministerstvo vnútra bude mať oprávnenie zobrazit’ všetky záznamy.

3.4.2 Užívateľské rozhranie

Pri pokuse o otvorenie webovej aplikácie bude teda neprihlásený užívateľ vždy presmerovaný na stránku s autentifikáciou. Po upravení HTML kódu, najmä po lokalizovaní jednotlivých elementov do slovenského jazyka, vyzerá užívateľské rozhranie pri prihlásení a registrácii nasledovne:

The screenshot shows a web application titled "EVIDENCIA POŽIAROV" (Fire Incident Reporting System) for the "Hasičský a záchranný zbor Slovenskej republiky" (Slovak Fire and Rescue Service). The page has a light blue background. At the top, there are links for "Hlavná stránka" (Main page), "Prihlásenie" (Login), and "Registrácia" (Registration). The "Registrácia" link is active, and a registration form is displayed in the center. The form has a title "Registrácia" and contains four input fields: "Meno" (Name), "E-mail", "Heslo" (Password), and "Potvrď heslo" (Confirm password). Below the fields is a blue button with a white user icon and the text "Zaregistrovať" (Register).

Obr. 33 Užívateľské rozhranie pri prihlasovaní (Vlastné spracovanie)



Obr. 34 Uživatelské rozhranie pri registrácii (Vlastné spracovanie)

3.5 Tvorba formulárov

Po inštalácii CSS frameworku a vyriešení autentizácie sa môžeme pustiť do formulárov. Tie budeme vytvárať za pomoci **Blade** šablón. Tu môžeme na rozdiel od kombinovania HTML a PHP používať špeciálnu syntax, ktorá je potom prekladaná do čistého PHP. Na nasledujúcom obrázku (Obr. 35) máme možnosť vidieť úryvok kódu, v ktorom sa používa špeciálna syntax pri plnení *select* elementu dátami.



Obr. 35 Navrhovanie formulára (Vlastné spracovanie)

3.5.1 Základné elementy

Pri tvorbe formulárov budeme využívať rôzne druhy elementov. Tak ako starý systém, začneme formulárom o celom incidente a jeho základných údajoch. Ukážeme si na ňom niekoľko príkladov použitých elementov.

- **Date and time**

Vďaka HTML kódu element *input*, ktorý bude typu "datetime-local". Framework má vopred definované, ako by mal taký element pre výber času a dátumu vyzeráť.

The screenshot shows a web form titled "Základné údaje o incidente". It contains several input fields: "Deň a čas spozorovania" (12.02.2016 18:21), "Deň a čas ohlásenia" (23.05.2016 18:21), "Obec" (a dropdown menu), "Adresa" (a text input field), "Poistovňa" (a dropdown menu with options like "Česká poisťovňa - Slovensko" and "D.A.S. poisťovňa právnej ochrany"), and "Vlastníctvo" (a dropdown menu). A calendar widget is open for the date "12.02.2016", showing the month of February 2016 with days of the week (po, ut, st, št, pi, so, ne) and dates (1-29).

Obr. 36 Čas a dátum formulára (Vlastné spracovanie)

- **Select**

Najviac používaným elementom v celom projekte bude práve element *select*. Dôvodom je veľký počet číselníkov, v ktorých si bude užívateľ vyberať z množstva možností. S cieľom čo najviac zjednodušiť výber možnosti, použitý bol element triedy *select2*, ktorý umožňuje vyhľadávanie podľa zadaného reťazca.

The screenshot shows the same web form "Základné údaje o incidente" as in the previous image. The "Obec" dropdown menu is open, showing a search bar with the text "Marš" and a list of suggestions: "Maršček", "Maršová", "Martin", and "Martinček". The "Adresa" field is empty. The "Poistovňa" and "Vlastníctvo" dropdown menus are also visible.

Obr. 37 Select vo formulári (Vlastné spracovanie)

Niektoré elementy select museli byť doplnené o takzvané *optgroup*. Jedná sa o množiny možností, ktoré ako také nemôžu byť zvolené, iba rozlišujú ostatné položky do skupín. Všetko muselo byť ošetrené v kóde pomocou podmienok.

Obr. 38 Položky selectu (Vlastné spracovanie)

- **Multiple select**

V prípade výberu poisťovní, ktorých môže byť ku jednému incidentu viac, bol použitý tzv. *multiple select*. Umožňuje zvoliť viacero možností pridržením klávesy control. Pre užívateľa bol doplnený pomocný text, ktorý sa zobrazí pri nabenutí kurzora na tento element.

Obr. 39 Multiselect vo formulári (Vlastné spracovanie)

- **Number, File**

Ďalším prvkom bol input typu number. Použitý bol pri zadávaní výšky škôd. Na Obr. 40 môžeme vidieť aj input typu *file*. Ten sme využili pri vkladaní fotografickej dokumentácie, teda (najmä) fotografií škôd spôsobených požiarom.

Obr. 40 Inputy a nahrávanie obrázkov (Vlastné spracovanie)

3.5.2 Evidenčné číslo

Po uložení prvého incidentu mu potrebujeme prideliť špecifické evidenčné číslo. To sa musí podľa starého systému skladať z oficiálneho kódu okresu, aktuálneho roku a prideleného ID. Pred samotným vložением incidentu do databázy potrebujeme vypočítať toto evidenčné číslo. Na to nám slúži funkcia, ktorá sa pustí vždy pred vytvorením incidentu a upraví jeho hodnoty podľa požiadaviek (Obr. 41).

```
Incident::creating(function ($incident){

    $code = Town::find($incident->town_id)->district->code;
    $year = Carbon::now()->format("y");
    $newIdObj = Incident::orderBy('id', 'desc')->first();

    $newId = 1;

    if($newIdObj)
        $newId = $newIdObj->id + 1;

    $incident->evidence_number = $code.$year.$newId;

    return $incident;
});
```

Obr. 41 Evidenčné číslo požiaru (Vlastné spracovanie)

3.5.3 Formuláre

Jednotlivé formuláre postupne získali svoju podobu. Jadro tvorí formulár so základnými údajmi o evidovanom incidente (Obr. 42), na ktorý sú napojené všetky ostatné formuláre. Všetky položky formulára je nutné vyplniť (s výnimkou toho, že hodnota škody môže zostať nulová a fotografická dokumentácia nemusí byť vždy prevedená, takže povinnosť pridať súbory nie je). Tomuto incidentu môžeme priradiť formulár Detaily incidentu

(Obr. 43), kde sa špecifikuje množstvo okolností požiaru. Položky nie sú povinné k vyplneniu.

Základné údaje o incidente

Deň a čas spozorovania	23.03.2016 17:18	Deň a čas ohlásenia	23.03.2016 17:18
Obec			
Adresa			
Poistovňa	Allianz poisťovňa, a. s. Amisico Prvá americko-slovenská poisťovňa Česká poisťovňa - Slovensko D.A.S. poisťovňa právnej ochrany Pracovník poisťovne		
Vlastníctvo			
Určenie škody			
Charakter škôd			
Odvetvie ekonomickej činnosti			
Škoda (€)	Priama: 0	Následná: 0	Uchránené hodnoty: 0
Fotografická dokumentácia	Vybrať súbory Nie je vybratý žiadny súbor		

Obr. 42 Formulár Incident (Vlastné spracovanie)

Detaily incidentu

Priestor	1302 základná škola
Miesto vzniku požiaru	1020 šatňa alebo odpočívareň
Časť dopravného prostriedku a pracovného stroja	
Dopravníkové zariadenia	
Príčina vzniku požiaru	3001 fajčenie
Manipulácia s horľavými výbušnými látkami	
Iniciátor	5005 nedopalk z cigarety
Časti elektrických rozvodov	
Pôsobenie iniciátora	202 teplota povrchu
Látky, ktoré začali horieť ako prvé	102 nečalunený nábytok
Látky, ktoré určovali rozvoj	105 záclona, záves
Trieda nebezpečnosti horľavých kvapalín	
Nedostatky v organizácii s vplyvom na vznik požiaru	05 nedostatočné alebo nevykonané školenia PO
Nedostatky v činnostiach s vplyvom na šírenie požiaru	01 neskoré spozorovanie požiaru
Uzatvorenie prípadu	603 odložením vyšetrovania

Obr. 43 Formulár Detaily Incidentu (Vlastné spracovanie)

Ku incidentu náleží aj poškodený objekt v prípade, že škoda na objekte nastala. Formulár (Obr. 44) tvoria špecifiká samotného poškodeného objektu a tiež poškodenia. Jeho vyplnenie taktiež nie je povinné. Zranenia (Obr. 45) je možné pridávať ku jednotlivým incidentom – v ponuke je vyplnenie štyroch jednoduchých selectov.

Poškodený objekt

Plocha/objem/hmotnosť		v jednotkách: ☐ ár ☐ m ² ☐ q
Stupeň poškodenia budov	01 drevostavba x ▾	
Funkcia požiarneho úseku		
Výška požiarneho úseku	03 1. až 3. podlažie (okrem jednopodlažných budov) x ▾	
Požiarna odolnosť stavebnej konštrukcie		
Horľavosť stavebných konštrukcií		
Požiarna bezpečnosť uzáverov	03 PB15 x ▾	
Požiarna odolnosť uzáverov		
Dôvod šírenia požiaru do ďalších úsekov		
Elektrická požiarňa signalizácia		
Stabilné/polostabilné hasiace zariadenia		
Hasiace látky v stabilnom hasiacom zariadení	05 stredná pena x ▾	

Obr. 44 Formulár Poškodený objekt (Vlastné spracovanie)

EVIDENCIA POŽIAROV

Hasičský a záchranný zbor Slovenskej republiky

Hlavná stránka Administrátor ▾

Zranenie vzniknuté pri incidente

Dôvod evidencie zranenia	02 zranenie závažné x ▾
Kategória zranenia	03 osoba od 15 do 60 rokov - mobilná x ▾
Okolnosti zranenia	02 spánok x ▾
Príčiny zranenia	02 otrava (pripotravenie) splodinami horenia x ▾

Obr. 45 Formulár Zranenia (Vlastné spracovanie)

Po pridaní údajov sa vrátime na celkový náhľad na incident, ktorý obsahuje všetky údaje, ktoré boli ku nemu pridané. Snaha bola čo najprehľadnejšie zobrazit' všetky možné dáta. Tie sa dajú priamo z náhľadu editovať.

[<< Naspäť](#)

INCIDENT Č. 501161

Incident		Detaily		Poškodený objekt	
Evidenčné číslo	501161	Priestor	transfúzna stanica	Stupeň poškodenia budov	drevostavba
Deň a čas spozorovania	2016-04-18 07:00:00	Miesto vzniku požiaru	-	Funkcia požiarneho úseku	-
Deň a čas ohlásenia	2016-04-18 16:05:00	Časť dopravného prostriedku a pracovného stroja	-	Výška požiarneho úseku	1. až 3. podlažie (okrem jednopodlažných budov)
Obec	Babiše	Dopravníkové zariadenia	pneumatický dopravník	Požiarňa odolnosť stavebnej konštrukcie	-
Adresa	Biela 3/19	Príčina vzniku požiaru	úmyselné zapálenie známou osobou	Horľavosť stavebných konštrukcií	-
Vlastníctvo	súkromné	Manipulácia s horľavými výbušnými látkami	-	Požiarňa bezpečnosť uzáverov	PB15
Určenie škody	škoda určená ZPP	Iniciátor	škafuľa v domovom rozvode	Požiarňa odolnosť uzáverov	-
Charakter škôd	vybavenie budov	Časti elektrických rozvodov	Iné meniče elektrického napätia	Dôvod šírenia požiaru do ďalších úsekov	-
Odvetvie ekonomickej činnosti	poľnohospodárstvo	Pôsobenie iniciátora	-	Elektrická požiarňa signalizácia	-
Priama škoda	350	Látky, ktoré začali horieť ako prvé	-	Stabilné/polostabilné hasiace zariadenia	-
Následná škoda	0	Látky, ktoré určovali rozvoj	-	Hasiace látky v stabilnom hasiacom zariadení	-
Uchránené hodnoty	0	Trieda nebezpečnosti horľavých kvapalín	horľavé kvapaliny II. triedy	Upraviť	
Poistovne	Allianz poisťovňa, a. s. Amislico Prvá americko-slovenská poisťovňa Česká poisťovňa - Slovensko	Nedostatky v organizácii s vplyvom na vznik požiaru	-		
Upraviť		Nedostatky v činnostiach s vplyvom na šírenie požiaru	-		
		Uzatvorenie prípadu	-		
		Upraviť			

Obr. 46 Prehľad incidentu (Vlastné spracovanie)

Hlavná stránka teda bola naplnená novými incidentami. Tie si zoradíme podľa toho, kedy boli pridané a vypíšeme im tie najpodstatnejšie informácie. Po pridaní väčšieho množstva záznamov bolo potrebné vyriešiť zobrazovanie incidentov. Pridalo sa stránkovanie, ktoré sa obmedzilo počtom 7 incidentov na stránku.

Nesmú chýbať tlačidlá na pridanie nových záznamov, editovanie aktuálnych záznamov a hlavička stránky, ktorá okrem presmerovania na hlavú stránku ponúka odhlásenie užívateľa.

EVIDENCIA POŽIAROV

Hasičský a záchranný zbor Slovenskej republiky

Hlavná stránka

Administrátor

+ Pridaj nový záznam

Posledné pridané incidenty

Evidenčné číslo	Obec	Adresa	Spozorovanie požiaru	Ohlásenie požiaru	
5101653	Štefanov nad Oravou	Hlavná 131	2016-03-26 17:20:00	2016-03-26 17:31:00	Zobraziť
5111652	Žilina	J. M. Hurbana 345/8	2016-02-02 11:20:00	2016-02-02 11:35:00	Zobraziť
5061651	Turčiansky Ďur	Turčiansky Ďur 11	2016-05-19 14:35:00	2016-05-19 14:45:00	Zobraziť
5011650	Malá Bytča	Malá Bytča 41	2016-01-10 00:10:00	2016-01-09 23:30:00	Zobraziť
5051649	Závažná Poruba	Hlavná 107/85	2016-05-23 15:21:00	2016-05-23 17:30:00	Zobraziť
5031648	Demkovská	Demkovská 7	2016-04-17 10:00:00	2016-04-17 11:10:00	Zobraziť
5061647	Martin	Kollárova 4330/87	2016-02-10 15:17:00	2016-02-10 18:20:00	Zobraziť

«

1

2

3

»

Obr. 47 Hlavná stránka webovej aplikácie (Vlastné spracovanie)

3.5.4 Responzívny design

Web navrhnutý prostredníctvom frameworku Bootstrap je responzívny bez potreby niečo upravovať. Stránka sa prispôsobí a poskladá jednotlivé elementy prehľadným spôsobom. Môžu sa tiež zakryť položky menu, ktoré sú dostupné po rozkliknutí pridanej ikony.

Takto sa správa hlavná stránka po zmenšení okna, ku ktorému môže dôjsť pri zmene zariadenia, na ktorom je stránka zobrazená (Obr. 48):

EVIDENCIA POŽIAROV					
Hasičský a záchranný zbor Slovenskej republiky					
Hlavná stránka					
+ Pridaj nový záznam					
Posledné pridané incidenty					
Evidenčné číslo	Obec	Adresa	Spozorovanie požiaru	Ohlásenie požiaru	
5061662	Valča	Valča 304	2016-05-11 21:08:00	2016-05-11 21:55:00	Zobraziť
5101653	Štefanov nad Oravou	Hlavná 131	2016-03-26 17:20:00	2016-03-26 17:31:00	Zobraziť
5111652	Žilina	J. M. Hurbana 345/8	2016-02-02 11:20:00	2016-02-02 11:35:00	Zobraziť
5061651	Turčiansky Ďur	Turčiansky Ďur 11	2016-05-19 14:35:00	2016-05-19 14:45:00	Zobraziť
5011650	Malá Bytča	Malá Bytča 41	2016-01-10 00:10:00	2016-01-09 23:30:00	Zobraziť
5051649	Závažná Poruba	Hlavná 107/85	2016-05-23 15:21:00	2016-05-23 17:30:00	Zobraziť
5031648	Demkovská	Demkovská 7	2016-04-17 10:00:00	2016-04-17 11:10:00	Zobraziť
« 1 2 3 »					

Obr. 48 Webová aplikácia v zmenšenom okne (Vlastné spracovanie)

3.5.5 Validácia

Laravel framework ponúka dve možnosti, ako validovať prichádzajúce dáta. Môžeme využiť vopred definované validačné pravidlá metódou **validate** alebo vytvoriť vlastný validátor použitím fasády **Validator**. Ja som sa rozhodla pre druhú možnosť.

Pre príklad si ukážeme validáciu dát formulára Incidenty, konkrétne metódy Update, ktorá prebieha pri ukladaní úprav záznamu.


```

\App\Http\Controllers\IncidentController

public function update(Request $request, $id)
{
    $incident = Incident::findOrFail($id);

    if(Gate::denies('update-incident', $incident)){
        abort(404);
    }

    $validator = Validator::make($request->all(), [
        'report_date' => 'required|date',
        'observe_date' => 'required|date',
        'address' => 'required',
        'ownership_id' => 'required',
        'damage_specification_id' => 'required',
        'damage_type_id' => 'required',
        'industry_type_id' => 'required',
        'town_id' => 'required',
        'direct_damage_value' => 'required|numeric',
        'followup_damage_value' => 'required|numeric',
        'saved_value' => 'required|numeric',
    ]);

    if ($validator->fails()) {
        return redirect()
            ->back()
            ->withErrors($validator)
            ->withInput();
    }

    $incident->update($request->all());

    $incident->insuranceCompanies()->detach();
    $incident->insuranceCompanies()->attach($request->get('insurance_companies'));

    Flash::success('Incident upravený.');
```

Obr. 49 Validátor (Vlastné spracovanie)

Metóda **make** generuje novú inštanciu validátora. Jej prvý argument sú dáta, ktoré ideme validovať, druhý sú pravidlá validácie, ktoré na dáta uplatníme. V tomto prípade vyžadujeme pri dátumoch dáta typu **date** a zároveň sú povinné, teda musia byť vyplnené, tak ako všetky ostatné položky formulára. Pri zadávaní hodnoty škôd (direct_damage_value – priama škoda, followup _damage_value – následná škoda, saved_value – uchránené hodnoty) sa bude vyžadovať číselná hodnota. Konkrétnejšie obmedzenie, napríklad to, že hodnota nemôže byť záporná, je už definované v HTML kóde v kombinácii s CSS (tag <div>, atribút min="0").

V prípade, že bude validácia neúspešná, validátor vráti chyby, ktoré na stránku aj vypíše nad formulárom.

Základné údaje o incidente

• Charakter škôd je povinné pole!

Deň a čas ohlásenia: 22.05.2016 15:27 Deň a čas spozorovania: 22.05.2016 15:27

Obec: Babin

Adresa: Biela 3/19

Poistovňa:

- Allianz poisťovňa, a. s.
- Amslico Prvá americko-slovenská poisťovňa
- Česká poisťovňa - Slovensko**
- D.A.S. poisťovňa právnej ochrany
- Domovská poisťovňa

Vlastníctvo: 02 | súkromné

Určenie škody: 01 | škoda neurčená

Charakter škôd:

Odvetvie ekonomickej činnosti: 103 | vodné hospodárstvo, rybolov, chov rýb

Škoda (€):

Priama: 0 Následná: 0 Uchránené hodnoty: 0

Fotografická dokumentácia: Vybrať súbory Nie je vybratý žiadny súbor

Naspäť Uložiť záznam

Obr. 50 Oznámenie chyby pri validácii dát (Vlastné spracovanie)

Zároveň je na Obr. 51 vidieť použitie triedy **Flash**, ktorá po úspešnej validácii a upravení dát incidentu spustí metódu *success*, Tá nám vypíše oznam o úspešnom upravení záznamu, ktorý nám po presmerovaní na hlavnú stránku zobrazí takto:

EVIDENCIA POŽIAROV

Hasičský a záchranný zbor Slovenskej republiky

Hlavná stránka Administrátor ▾

Incident upravený. ×

+ Pridaj nový záznam

Posledné pridané incidenty

Evidenčné číslo	Obec	Adresa	Spozorovanie požiaru	Ohlásenie požiaru	
5061662	Valča	Valča 304	2016-05-11 21:08:00	2016-05-11 21:55:00	Zobraziť
5101653	Štefanov nad Oravou	Hlavná 131	2016-03-26 17:20:00	2016-03-26 17:31:00	Zobraziť
5111652	Žilina	J. M. Hurbana 345/8	2016-02-02 11:20:00	2016-02-02 11:35:00	Zobraziť
5061651	Turčiansky Ďur	Turčiansky Ďur 11	2016-05-19 14:35:00	2016-05-19 14:45:00	Zobraziť
5011650	Malá Bytča	Malá Bytča 41	2016-01-10 00:10:00	2016-01-09 23:30:00	Zobraziť
5051649	Závažná Poruba	Hlavná 107/85	2016-05-23 15:21:00	2016-05-23 17:30:00	Zobraziť

Obr. 51 Úspešná aktualizácia záznamu (Vlastné spracovanie)

3.6 Odhadovaný rozpočet

Je nutné brať ohľad aj na odhadovanú cenu návrhu nového systému. Zbor musí pracovať s rozpočtom, ktorý mu bude na tento projekt poskytnutý. Požiadal preto aspoň o stručnú kalkuláciu nákladov, ktoré pri vytváraní systému a pri jeho údržbe vzniknú.

Tabuľka 2 Odhadované náklady na vytvorenie systému (Vlastné spracovanie)

Náklady na vytvorenie systému	
Činnosť	Odhadovaná cena
vývoj databázovej schémy	130.00 €
návrh užívateľského prostredia	50.00 €
testovanie	30.00 €
zhotovenie systému	180.00 €
Σ =	390.00 €

V tabuľke môžeme nájsť odhadované náklady na vytvorenie systému, ktoré boli konzultované s vedením zboru. Konečná suma činí 390 eur. Nie sú tu uvedené napríklad náklady na doménu a webhosting, pretože o tieto položky má už zbor postarané a teda nie sú súčasťou nákladov na nový systém.

Tabuľka 3 Odhadované potenciálne náklady (Vlastné spracovanie)

Potenciálne náklady	
Činnosť	Odhadovaná cena
implementácia systému	150.00 €
zaškolenie zamestnancov	20.00 €/hod
údržba systému	30.00 €/hod
rozšírenie systému	230.00 €

Zboru je tiež poskytnutá možnosť využiť ďalšie služby. Môžu sa rozhodnúť, či stoja o implementáciu systému či prípadné školenie zamestnancov, ktoré však podľa všetkého nie je potrebné. Ďalej je uvedená kalkulácia ceny prípadného rozšírenia systému pre ďalší kraj, v prípade rozšírenia pre celú krajinu by sa náklady kalkulovali dohodou.

3.7 Prínosy navrhnutého systému

Hasičský a záchranný zbor Slovenskej republiky používal na evidenciu požiarov veľmi zastaraný a nepraktický systém, ktorý už dlhšiu dobu potreboval vylepšiť, prípadne úplne nahradiť.

Navrhnutý systém by mal pre zbor znamenať významné prínosy, ktoré povedú k:

- zvýšeniu efektivity práce
 - komfortnejšiemu používaniu
 - bezpečnejšiemu udržiavaní dôležitých záznamov
 - novým možnostiam evidencie požiarov
1. Systém bude prístupný neustále a kdekoľvek prostredníctvom webovej aplikácie. Pre prístup budú potrebné len prihlasovacie údaje.
 2. Zavedením nového systému už nebude naďalej nutné inštalovať akýkoľvek program lokálne.
 3. Používanie systému bude prehľadné, nenáročné a vhodné pre všetky typy užívateľov. Vloženie novej evidencie zaberie minimum času.
 4. Záznamy budú evidované podľa zadaných pravidiel a ich štruktúra ostane zachovaná aj po editácii rôznymi užívateľmi.
 5. Najdôležitejšie údaje budú ošetrené tak, aby nehrozilo, že sa opomenie ich vyplnenie. Systém užívateľa upozorní na chýbajúci alebo nesprávne zadaný údaj.
 6. Systém poskytne rýchly náhľad záznamov, ktoré boli pridané alebo editované ako posledné.
 7. Dáta budú zálohované a nebude tak hroziť ich strata.
 8. Ministerstvo vnútra bude mať možnosť vzdialeného prístupu k údajom a to bez nutnosti kontaktovať členov zboru. Do tejto doby im boli poskytované vyexportované dáta len na požiadanie.

Veľkou výhodou systému je možnosť rozšíriť ho po území celej Slovenskej republiky v prípade, že sa tento systém osvedčí. Pracuje totiž s číselníkmi všetkých obcí, okresov a krajov a pre rozšírenie stačí pár úprav v programe.

Záver

Cieľom tejto práce bolo navrhnuť systém, ktorý by slúžil Hasičskému a záchrannému zboru Slovenskej republiky na evidenciu záznamov o požiaroch.

V teoretickej časti sú vysvetlené základné pojmy používané pri práci s databázou a priblížené teoretické východiská relačných databáz. Ďalej sú predstavené aj technológie, prostredníctvom ktorých bol systém navrhovaný. V stručnosti sú tu popísané jednotlivé programovacie jazyky a frameworky, ktoré boli dôležitou súčasťou praktickej časti práce.

V nasledujúcej časti bol predstavený Hasičský a záchranný zbor Slovenskej republiky, pre ktorý je systém navrhovaný. Ďalej prebehla analýza súčasného stavu v zbore. To zahŕňalo hodnotenie hardvérového vybavenia, stavu informačných technológií a podrobne sa tu zhodnotil najmä systém, ktorý sa doposiaľ na evidenciu požiarov používal. Boli odhalené nedostatky systému, z ktorých sa vyvodili dôvody pre realizáciu návrhu nového systému. Na záver prebehlo zhodnotenie pripravenosti organizácie na zavedenie nového systému, z ktorého vyplynulo, že je na zavedenie pripravená.

Posledná časť práce sa venuje samotnému návrhu nového systému. Podrobne nás oboznamuje s jednotlivými krokmi, ktoré viedli k vytvoreniu systému.

Prvým krokom je spracovanie databázovej schémy, ktorá je pomerne rozsiahla - jej prehľadnejšia verzia sa nachádza aj v prílohe. Nasleduje inštalácia zvoleného PHP frameworku a migrácia navrhutej databázovej schémy do systému. Ďalším krokom je riešenie bezpečnostnej stránky systému - autentifikácia. Po nainštalovaní vhodného CSS frameworku sa prechádza k návrhu formulárov a úprave designu, kde bola vynaložená snaha o čo najprehľadnejšie užívateľské prostredie. Nechýba ani riešenie validácie dát z formulárov, ktoré budú užívatelia posielat' na server. Na základe triedenia do užívateľských skupín majú prihlásení užívatelia rôzne oprávnenia na obsluhu systému.

Na záver bola prevedená ekonomická analýza, z ktorej vyplynul odhadovaný rozpočet. Ten pozostáva z nákladov rozdelených do dvoch skupín. V prvej sú náklady na návrh

nového systému. To sú náklady, s ktorými by mal zbor počítat'. Ďalšiu skupinu tvoria potenciálne náklady, ktoré môžu vzniknúť v prípade, že bude mať zbor záujem o ďalšie služby.

Systém pre evidenciu požiarov pre Žilinský kraj je teda navrhnutý. Po implementácii na server zboru bude pripravený na naplnenie novými záznamami o požiaroch – ostáva nám dúfať, že ich bude čo najmenej.

Zoznam použitých informačných zdrojov

Knižné zdroje:

- (1) MATIAŠKO, Karol. *Databázové systémy: základy databázových systémov*. Vyd. 2. Žilina: Žilinská univerzita v Žiline, 2008. 351 s. ISBN 978-80-8070-820-7.
- (2) RIORDAN, Rebecca M. *Vytváříme relační databázové aplikace*. Vyd. 1. Praha: Computer Press, 2000. 280 s. ISBN 80-7226-360-9.
- (3) DATE, C.J. *An Introduction to Database Systems*. Vyd. 6. Reading: Addison-Wesley, 1995. 820 s. ISBN 0-201-82458-2.
- (4) KOCH, Miloš. *Datové a funkční modelování*. Vyd. 1. Brno : Akademické nakladatelství CERM, 2004. 108 s. ISBN 80-214-2724-8.
- (5) VRÁNA, Jakub. *1001 tipů a triků pro PHP*. Vyd. 1. Brno: Computer Press, 2010. 456 s. ISBN 978-80-251-2940-1.
- (6) SHARKIE, Craig a Andrew FISHER. *Responzivní webdesign: okamžitě*. Brno: Computer Press, 2015. 144 s. ISBN 978-80-251-4384-1.
- (7) HOLZSCHLAG, M. E. *HTML a CSS: jdi do toho*. Praha: Grada Publishing, 2006, 263 s. ISBN 978-80-247-1454-7.

Elektronické zdroje:

- (8) FARANA, Radim. *Databázové systémy Microsoft Access 2.0. VŠB - Technická Univerzita Ostrava* [online]. © 2006 [cit. 2016-3-20]. Dostupné z: <http://books.fs.vsb.cz/dbacc20/Welcome.htm>
- (9) ROOT.CZ. *Velký test PHP frameworků (2. díl)*. *Root.cz* [online]. © 1998 - 2016 [cit. 2016-3-20]. Dostupné z: <http://www.root.cz/clanky/velky-test-php-frameworku-2-dil/>
- (10) SIDEPOINT. *The Best PHP Framework for 2015: SitePoint Survey Results*. *SitePoint Pty. Ltd* [online]. © 2000 - 2016 [cit. 2016-4-20]. Dostupné z: <http://www.sitepoint.com/best-php-framework-2015-sitepoint-survey-results/>
- (11) MINISTERSTVO VNÚTRA SR. *Hasičský a záchranný zbor*. *Ministerstvo vnútra Slovenskej republiky* [online]. © 2016 [cit. 2016-2-10]. Dostupné z: <http://www.minv.sk/?hasicky-a-zachranny-zbor>

- (12) LARAVEL.COM. *Laravel* [online]. © 2016 [cit. 2016-3-13]. Dostupné z:
<https://laravel.com/>
- (13) ŽELEZNÝ, Pavel. Od špagety k objektům (4) - Framework. *Blog.pavelzelezny.cz* [online]. © 2016 [cit. 2016-3-28]. Dostupné z:
[http://blog.pavelzelezny.cz/od-spagety-k-objektum/\(4\)-framework.html](http://blog.pavelzelezny.cz/od-spagety-k-objektum/(4)-framework.html)
- (14) AWWWARDS. What are Frameworks? 22 Best Responsive CSS Frameworks for Web Design. *Awwwards Online S.L* [online]. © 2010 - 2016 [cit. 2016-5-10]. Dostupné z: <http://www.awwwards.com/what-are-frameworks-22-best-responsive-css-frameworks-for-web-design.html>
- (15) MACHAČ, Marek. 11 nejlepších HTML a CSS frameworků pro webdesignéry. *Interval.cz* [online]. © 2016 [cit. 2016-3-10]. Dostupné z:
<https://www.interval.cz/clanky/11-nejlepsich-html-a-css-frameworku-pro-webove-designery/>
- (16) ZURB, INC. *Foundation* [online]. © 2016 [cit. 2016-3-13]. Dostupné z:
<http://foundation.zurb.com/>
- (17) BOOTSTRAP.COM. *Bootstrap* [online]. © 2016 [cit. 2016-4-12]. Dostupné z:
<http://getbootstrap.com/>
- (18) YAHOO! INC. *Pure* [online]. © 2016 [cit. 2016-4-12]. Dostupné z:
<http://purecss.io/>
- (19) SEMANTIC-UI.COM. *Semantic UI* [online]. © 2016 [cit. 2016-4-12].
Dostupné z: <http://semantic-ui.com/>
- (20) CROFT, Jeff. Frameworks for Designers. *A List Apart* [online]. © 1998 -2016 [cit. 2016-4-12]. Dostupné z:
<http://alistapart.com/article/frameworksfordesigners>
- (21) BRADLEY, Steven. The Pros And Cons Of CSS Frameworks. *Vanseo Design* [online]. © 2005 -2016 [cit. 2016-4-12]. Dostupné z:
<http://vanseodesign.com/css/css-frameworks-pros-cons/>
- (22) LARACASTS.COM. *Laracasts* [online]. © 2016 [cit. 2016-4-12]. Dostupné z:
<https://laracasts.com/>
- (23) KUBÍK, Stanislav. HTML 5 I. - Úvod do HTML 5. *Linux Software* [online]. © 2003 -2016 [cit. 2016-5-2]. Dostupné z:
http://www.linuxsoft.cz/article.php?id_article=1994

(24) WEBENERTIA.COM. *WebEnertia* [online]. © 1999 - 2016 [cit. 2016-5-1].

Dostupné z: <http://www.webenertia.com/>

(25) EUROEKONÓM.SK. SWOT analýza. *EuroEkonom.sk* [online]. © 2004 - 2016

[cit. 2016-4-15]. Dostupné z:

<http://www.euroekonom.sk/manazment/strategicka-diagnostika/swot-analyza/>

Zoznam obrázkov

Obr. 1: Terminológia relačných databáz.....	13
Obr. 2: Lineárny dátový model.....	16
Obr. 3: Hierarchický dátový model	16
Obr. 4: Sieťový dátový model	17
Obr. 5 Webová stránka frameworku Laravel.....	24
Obr. 6 Video tutoriál stránky Laracasts	25
Obr. 7 Riešenie dokumentácie frameworku Laravel	25
Obr. 8 Podpora prehliadačov pre HTML5 elementy	27
Obr. 9 Vrstvy architektúry	31
Obr. 10 webová stránka frameworku Bootstrap	33
Obr. 11 Webová stránka frameworku Pure	33
Obr. 12 Webová stránka Semantic UI	34
Obr. 13 Formuláre s frameworkom Bootstrap.....	35
Obr. 14 Oficiálne logo Hasičského a záchranného zboru.....	37
Obr. 15 Webový portál HaZZ.....	41
Obr. 16 Program POŽIAR.....	42
Obr. 17 Napĺňanie dát.....	43
Obr. 18 Prezeranie údajov	43
Obr. 19 Tvorba odborného posudku	44
Obr. 20 Formulár č. 2.....	45
Obr. 21 Databázová schéma systému 1	58
Obr. 22 Databázová schéma systému 2	59
Obr. 23 Databázová schéma systému 3	60
Obr. 24 Laravel Schema Designer.....	62
Obr. 25 Štruktúra priečinkov projektu.....	63
Obr. 26 Migrácie.....	64
Obr. 27 Súbor migrácie pre tabuľku Areas.....	65
Obr. 28 Vytváranie seederu pre tabuľku s obcami	66
Obr. 29 Seeder pre tabuľku s obcami	67
Obr. 30 Základná štruktúra controlleru.....	68

Obr. 31 Štruktúra projektu po inštalácii Bootstrap	69
Obr. 32 Použitie middleware na autentifikáciu	70
Obr. 33 Užívateľské rozhranie pri prihlasovaní	71
Obr. 34 Užívateľské rozhranie pri registrácii	72
Obr. 35 Navrhovanie formulára.....	72
Obr. 36 Čas a dátum formulára.....	73
Obr. 37 Select vo formulári	73
Obr. 38 Položky selectu	74
Obr. 39 Multiselect vo formulári	74
Obr. 40 Inputy a nahrávanie obrázkov.....	75
Obr. 41 Evidenčné číslo požiaru.....	75
Obr. 42 Formulár Incident	76
Obr. 43 Formulár Detaily Incidentu	76
Obr. 44 Formulár Poškodený objekt.....	77
Obr. 45 Formulár Zranenia	77
Obr. 46 Prehľad incidentu.....	78
Obr. 47 Hlavná stránka webovej aplikácie	79
Obr. 48 Webová aplikácia v zmenšenom okne.....	80
Obr. 49 Validátor	81
Obr. 50 Oznámenie chyby pri validácii dát	82
Obr. 51 Úspešná aktualizácia záznamu	82

Zoznam tabuliek

Tabuľka 1 SWOT analýza súčasného systému.....	52
Tabuľka 2 Odhadované náklady na vytvorenie systému	83
Tabuľka 3 Odhadované potenciálne náklady	83

Prílohy

