**Návrh**

**World Value Survey (DEKK)**

Získanie údajov, vizualizácia, výpočet korelácie,

aktualizácia dát, štatistické údaje

Zuzana Mačicová, Vladimír Ačjak, Marek Lichvár, Richard Mištík

**Obsah**

[**1. Úvod** 3](#_Toc95061838)

[**1.1 Účel dokumentu** 3](#_Toc95061839)

[**1.2 Prehľad nasledujúcich kapitol** 3](#_Toc95061840)

[**2. Špecifikácia vonkajších rozhraní** 4](#_Toc95061841)

[**3. Dátový model** 5](#_Toc95061842)

[**4. Grafické používateľské rozhranie** 6](#_Toc95061843)

[Ľavý panel 6](#_Toc95061844)

[Stredný panel 8](#_Toc95061845)

[Pravý panel 9](#_Toc95061846)

[**5. Návrh implementácie** 10](#_Toc95061847)

[**5.1 Diagram komponentov (UML Component diagram)** 10](#_Toc95061848)

[**5.2 Sekvenčný diagram (Sequence diagram)** 11](#_Toc95061849)

[**5.3 Štruktúra projektu (Namiesto class diagramu)** 11](#_Toc95061851)

[**5.5 Využité technológie** 12](#_Toc95061852)

[**5.6 Cieľové prostredie nasadenia do prevádzky** 13](#_Toc95061853)

# **1. Úvod**

## **1.1 Účel dokumentu**

Dokument slúži ako detailný návrh informačného systému. Obsahuje informácie vysvetľujúce funkcionalitu a spôsob implementácie systému. Dokument pokrýva všetky požiadavky z katalógu požiadaviek a je určený hlavne pre vývojárov.

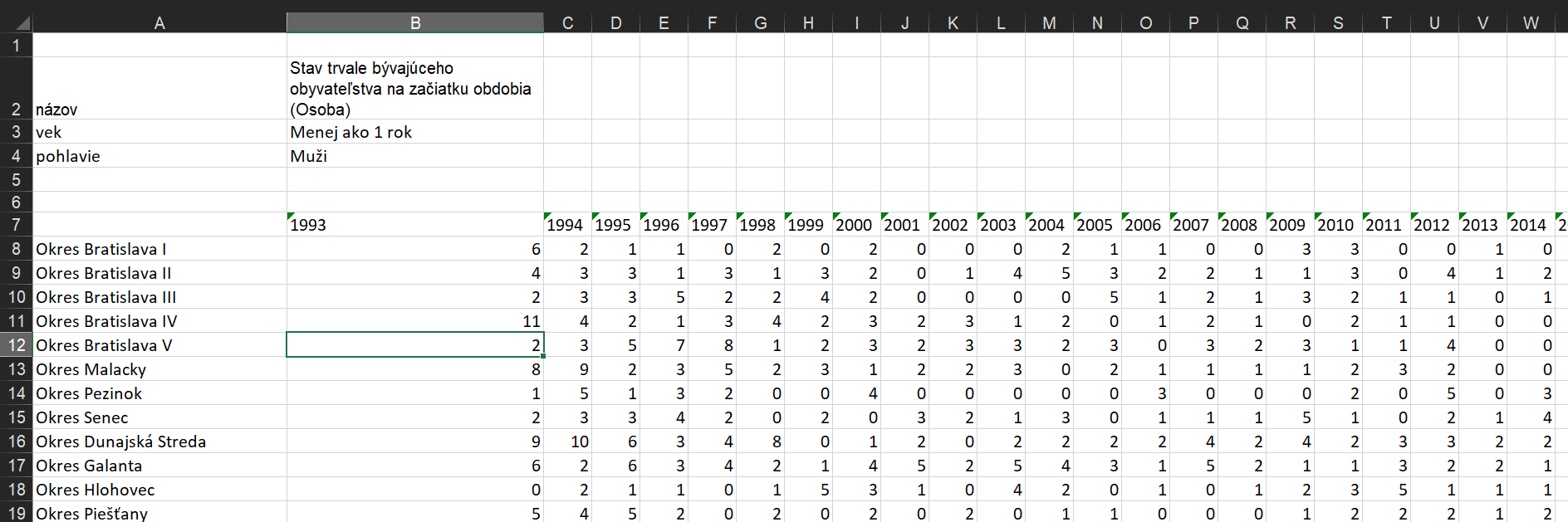
## **1.2 Prehľad nasledujúcich kapitol**

Kapitoly sa venujú špecifikácii vonkajších interfejsov, dátovému modelu, používateľskému rozhraniu, vizuálnemu zobrazeniu aplikácie a celkovému návrhu implementáci

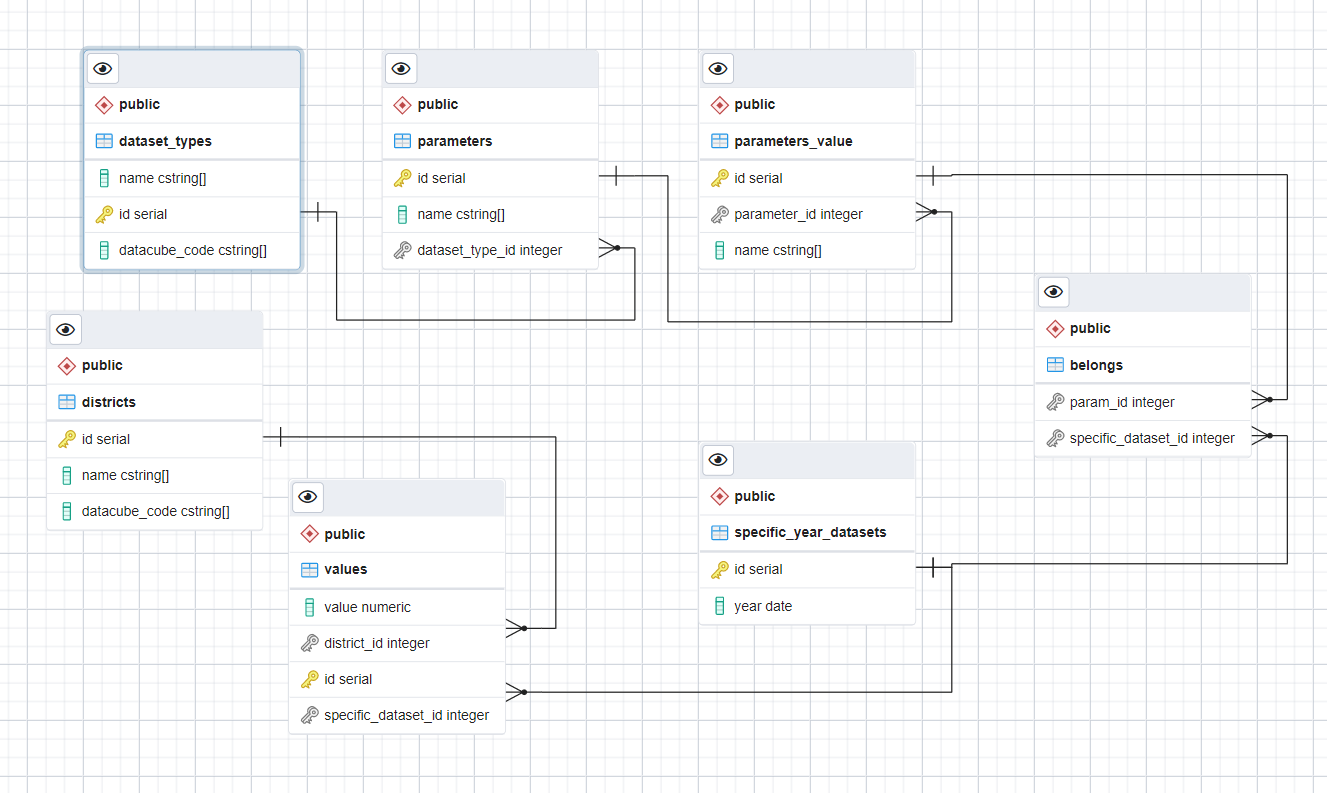
# **2. Špecifikácia vonkajších rozhraní**

* Aplikácia bude s používateľom komunikovať pomocou grafického používateľského rozhrania, ktoré bude bežať vo webovom prehliadači
* Frontend bude s backendom komunikovať pomocou HTTP request/response
* Na backende bude komunikácia s databázou MySQL zabezpečená frameworkom Laravel, ktorý bude súčasne aj spracovávať požiadavky z frontendu
* Pre výpočet korelácie sa využíva .py script, ktorý sa spúšťa z Laravel-u. Tento script najprv zbehne Shapiro-Wilk a normálny test. Ak oba testy prejdú, použije sa Personova korelácia. Inak sa použije Spearmanova korelácia.
* Zoznam okresov sa do databázy, ktoré sa pri prvom set-upe importujú do databázy sú v súbore .txt s názvom *okresy\_data.txt,* v každom riadku je jeden názov okresu
* Automatický import dát do databázy bude získavať dáta cez API zo štatistického úradu - <http://datacube.statistics.sk/> vo formáte .json: <https://slovak.statistics.sk/wps/portal/ext/Databases/Open_data/!ut/p/z0/04_Sj9CPykssy0xPLMnMz0vMAfIjo8ziw3wCLJycDB0NDEJ9LA0cg1yMDS28zIycgw31C7IdFQHW33iE/?fbclid=IwAR0CXVTJq-QUl3gutOFbSZ1MFj-9yRVhtu_CJw7NtM9MMKF8HOddnnAWbc0>
* Používateľ bude môcť dáta importovať aj priamo z vybraného .csv súboru pomocou python scriptu, formát .csv súboru bude podobný tomu zo štatistického úradu:
  + hodnoty v riadku budú oddelené čiarkou, riadky budú oddelené Enterom
  + prvý riadok popisuje názov datasetu
  + ďalšie riadky popisujú parametre datasetu
  + ďalej nasleduje aspoň jeden riadok prázdny
  + nasledujúci riadok hovorí o tom pre aký rok sú hodnoty v danom stĺpci
  + pod rokmi začína samotný dataset, v prvom stĺpci je názov okresu, ďalej nasledujú už konkrétne hodnoty pre daný okres a rok
* tento import sa bude realizovať cez terminál spustením python scriptu, ktorý si vypýta názov súboru, ktorý chceme importovať

Ten istý súbor zobrazený v Excel:



# **3. Dátový model**

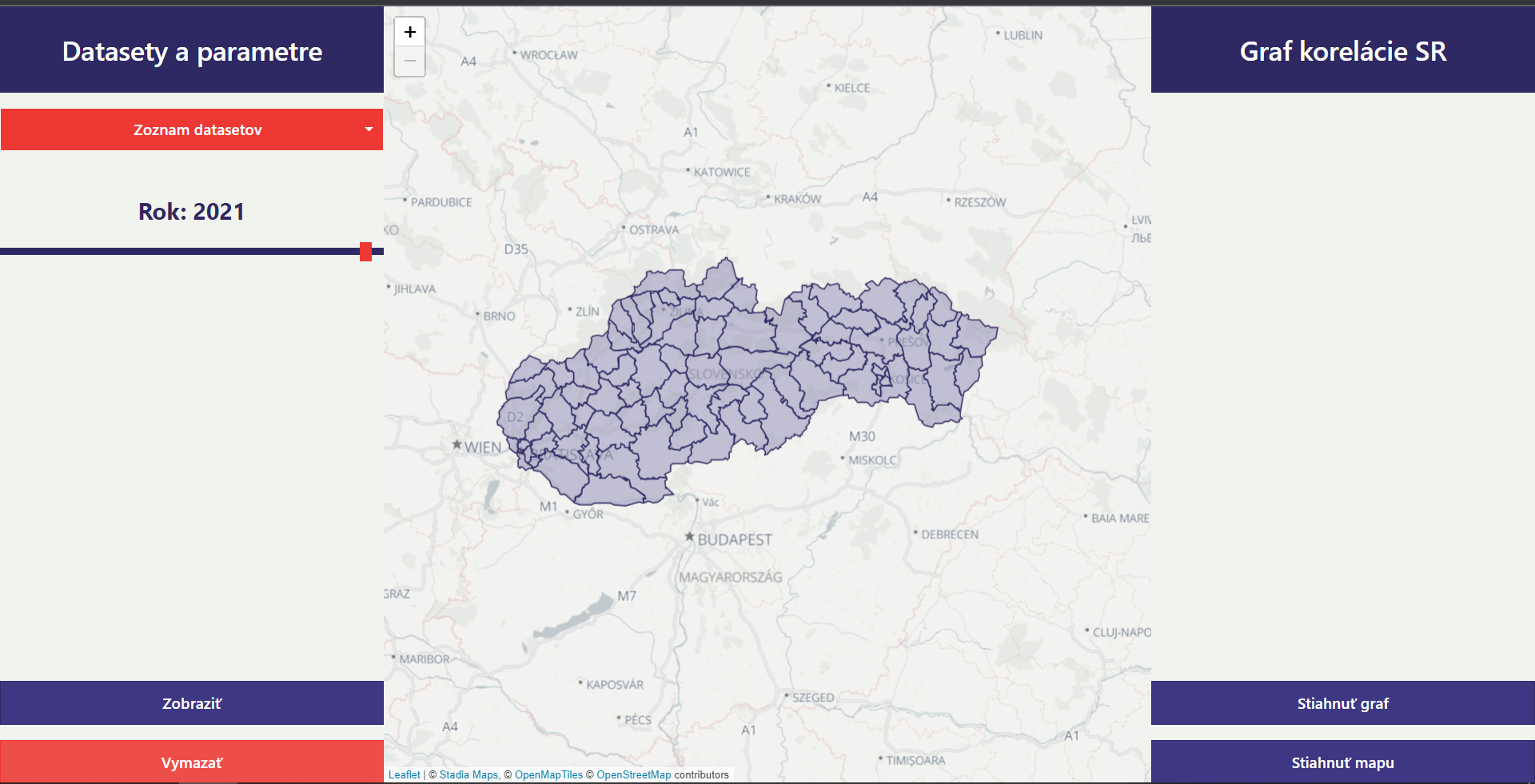


Tabuľka **dataset\_types** obsahuje aké typy datasetov máme, napr. Kriminalita, Počet obyvateľov, … . Každý typ datasetu má svoje automaticky generované **id, datacube\_code** a **name** - meno, čiže stringový reťazec, ktorý sa zobrazí aj používateľovi. **Datacube\_code**  je kód, ktorý majú len datasety z datacube. **Datacube\_code** aj **id** sú unikátne. Ku každému typu datasetu existujú nejaké parametre, napr. pohlavie a vek, o ktorých hovorí tabuľka **parameters**. Pre všetky parametre platí, že sú z nejakého datasetu, odkazujú sa naň **dataset\_type\_id,** automaticky generované **id,** **name** - meno a hodnoty, ktoré sú popísané v tabuľke **parameters\_value**. Ako príklad si zoberme parameter pohlavie, ktorý má dve hodnoty muž a žena. Každá hodnota parametra má svoje automaticky generované **id**, **name** - meno a **parameter\_id**, ktorým sa odkazuje na príslušný parameter. Každý typ datasetu má defualtný parameter “spolu” s jedinou hodnotou “spolu”. Označuje údaje bez parameta, teda dáta platné pre všetky skupiny obyvateľstva. Parameter “spolu” sa tiež musí samostatne do databázy (nevypočíta sa sčítaním čiastkových hodnôt).

V tabuľke **districts** sa nachádza zoznam všetkých okresov Slovenska, stĺpec **id** je unikátny identifikátor pre každý okres, **datacube\_code** je kód okresu podľa datacube a stĺpec **name** je názov okresu. Do tabuľky **districts** sa odkazuje tabuľka values, ktorá uchováva v stĺpci **value** konkrétne číselné hodnoty, pre daný dataset, s hodnotou určitého parametra pre vybraný rok. Špecifický dataset (teda typ parametrovanizovaný typ datasetu) je v tabuľke **specific\_dataset**. Dôležitý je atribút **year**, ktorý hovorí o roku, ku ktorému prislúcha dataset. Väzobná tabuľka **belongs** spája tabuľku **specific\_dataset**  a **parameters\_value** - hovorí o parametroch prislúchajúcich k špecifickému datasetu.

# **4. Grafické používateľské rozhranie**

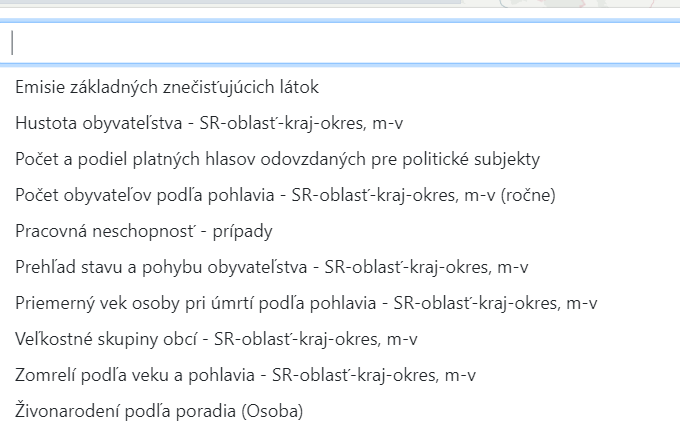
Vzhľad stránky po načítaní:

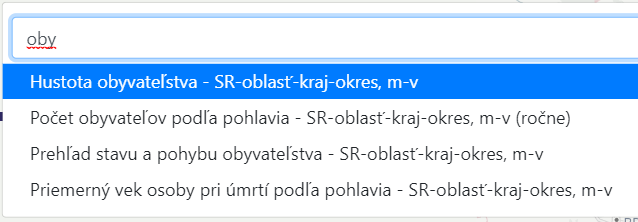


Aplikácia je rozdelená na 3 základné panely - ľavý, stredný, pravý.

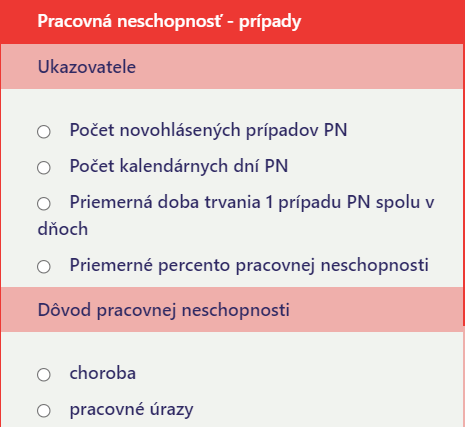
## Ľavý panel

Slúži na výber datasetov. Tlačítkom “Zoznam datasetov” si vie používateľ zobraziť všetky dostupné datasety:



Ďalej je možnosť filtrovania v nich podľa názvu:

Používateľ si dataset vyberie kliknutím naň. Vedľa vybraného datasetu sa zobrazí a v okienku nižšie sa zobrazia možné parametre zvoleného datasetu.



Ďalej je možné si vybrať za aký rok sa majú dáta zobraziť pomocou posúvania “slidera”:

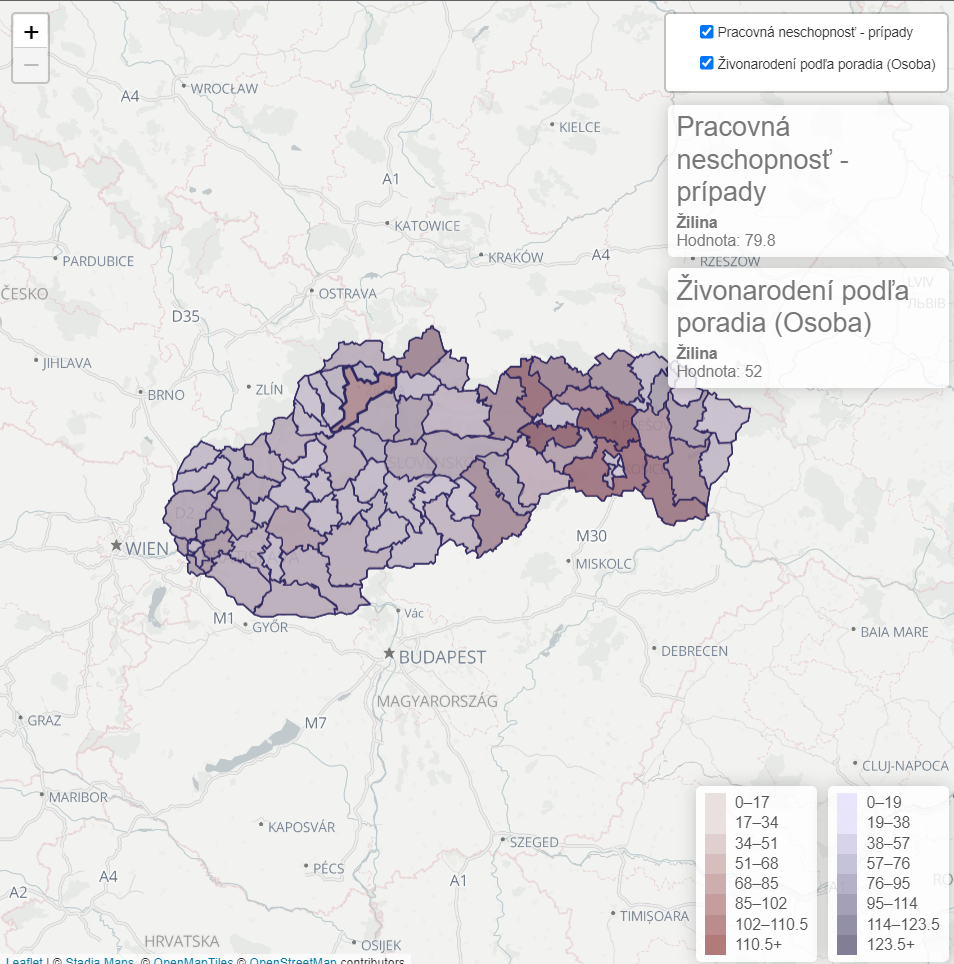


Keď má používateľ zvolený rok aj parametre, tak stlačením  sa mu vo zvyšných paneloch zobrazí mapa, graf a korelácia. Ak pre zvolené parametre a rok chýbajú dáta, používateľ je o tom notifikovaný pomocou vyskakovacieho okna v prehliadači.

Tlačítko slúži na vymazanie výberu parametrov.

## Stredný panel

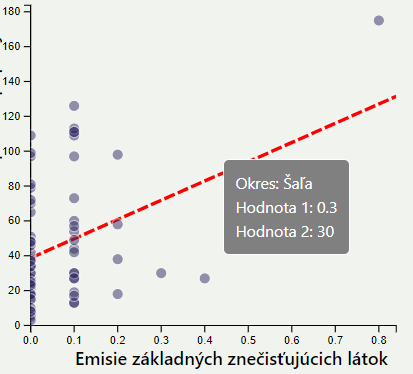
Slúži na zobrazenie mapy Slovenska, rozdelenej na okresy. Po vybraní datasetov sa zafarbí podľa číselných hodnôt pre jednotlivé okresy.

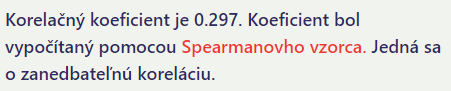


Mapu je možné si priblížiť a oddialiť tlačítkami v ľavom hornom rohu. V pravom hornom rohu sa zobrazujú vybrané datasety s checkboxom - vieme si zobraziť aj len jeden dataset na mape. Po nadídení myšou nad jednotlivé okresy sa v pravo hore zobrazí názov okresu a číselná hodnota pre jednotlivé datasety. V ľavom rohu sa nachádza legenda farieb.

## Pravý panel

Obsahuje graf pre zvolené datasety (na každej osi 1 dataset) a priamku, ktorá najlepšie aproximuje dané datasety. Bodky znázorňujú jednotlivé hodnoty a po nadídení nad ne sa zobrazia presné číselné hodnoty aj s názvom prislúchajúceho okresu.

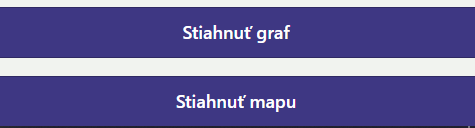


Pod grafom sa nachádza informácia o korelácií daných datasetov. Udáva korelačný koeficient, aký vzorec bol použitý pre vypočítanie danej korelácie a či je korelácia významná.

Tabuľka vyhodnotenia korelačného koeficientu r:

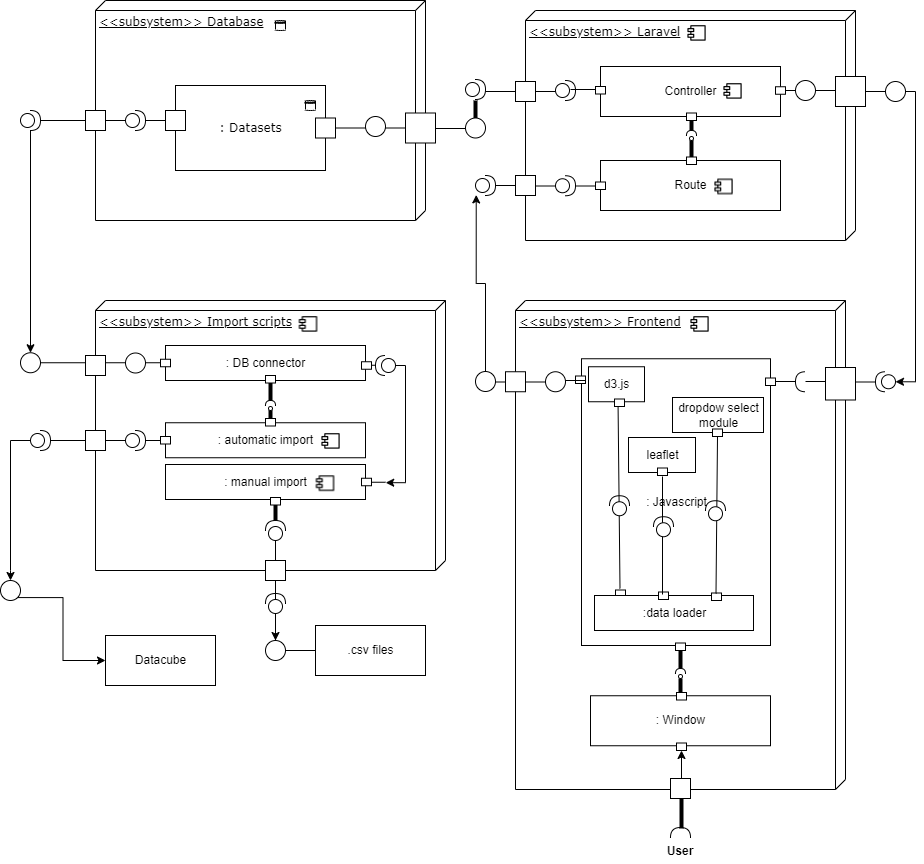
| **r** | **sila korelácie** |
| --- | --- |
| [-1, -0.7] | silná negatívna |
| (-0.7, -0.5] | priemerná negatívna |
| (-0.5, -0.3] | slabá negatívna |
| (-0.3, 0) | zanedbateľná negatívna |
| [0, 0.3) | zanedbateľná |
| [0.3, 0.5) | slabá |
| [0.5, 0.7) | priemerná |
| [0.7, 1] | silná |

Na spodku pravého panelu sa nachádzajú tlačítka pre uloženie mapy a grafu.



# **5. Návrh implementácie**

## **5.1 Diagram komponentov (UML Component diagram)**



## **5.2 Sekvenčný diagram (Sequence diagram)**

## 

## 

## 

## 

## **5.3 Štruktúra projektu**

**Routes**

* **web.php** - definuje webové routes, mapuje GET requesty na funkcie triedy HomeController

**Controllers**

* **HomeController.php** - komunikuje s MYSQL databázou, spúšťa na backende pythonovský script na výpočet korelácie

**Database**

* **Migrations** - vytvorí všetky počiatočné tabuľky v databáze
* **Seeders** - DatabaseSeeder.php - naplní databázu okresmi definovanými v ../seeders/okresy\_data.txt

**Public**

**Priečinok public obsahuje skompilované JavaScript a Cascading Style Sheets súbory.**

* **js** - frontend JavaScript, zobrazovanie mapky, grafu, dropdown menu s výberom datasetov, export mapy a grafu ako obrázok
* **css** - dizajn stránky, ako sa má HTML zobrazovať na rôznych rozlíšeniach a zariadeniach
* **correlation.py** - pythonovský script na výpočet korelácie z príkazového riadku

**Resources**

**Priečinok resources slúži na ukladanie pracovných verzií .css a .js súborov. Finálne verzie sú skompilované v priečinku public.**

* **views -** map\_leaflet.blade.php - úvodná stránka webovej aplikácie, zobrazuje mapu, graf, koreláciu medzi datasetmi a dropdown menu pre výber datasetov
* **css** - pracovná verzia .css súboru
* **js** - jednotlivé JavaScript súbory, ktoré sú skompilované v **public** priečinku. Bootstrap.js obsahuje referencie na dependency packages, ktoré sú v skompilované v public/app.js. Zvyšné súbory sú skompilované v public/main.js

**Konfigurácia**

* **.env** - Laravel konfigurácia, obsahuje prihlasovacie údaje do MYSQL databázy

**Import Scripts**

**Všetky import scripty sa spúšťajú z príkazového riadku.**

* **import\_csv.py** - načíta údaje do MYSQL databázy z csv súboru pomocou príkazového riadku príklad použitia: python3 import\_csv.py volby.csv
* **api\_final.py** - načíta údaje do MYSQL databázy podľa datacube kódu pre dataset a roku pomocou príkazového riadku.   
  Príklad použitia: python3 api\_final.py omr1234 2011
* **fetch.py** - načíta údaje do MYSQL databázy pre všetky datasety a roky z intervalov v súbore datasety\_api.txt.   
  Každý riadok v súbore musí dodržať formát “kód\_datasetu prvý\_rok posledný\_rok”. Informácie o úspechu/neúspechu pre každý import sa logujú v súbore log.txt.   
  Príklad použitia: python3 fetch.py

## **5.5 Využité technológie**

* **Python -** scriptovací programovací jazyk umožňuje vytváranie scriptov na manuálne a automatizované spracovanie a import datasetov do databázy
* **Laravel -** framework, ktorý zabezpečuje komunikáciu s databázou MySQL, backendovú funkcionalitu
* **Php -** implementačný programovací jazyk, ktorý používa Laravel
* **JavaScript** - scriptovací programovací jazyk, ktorý zabezpečuje frontendovú funkcionalitu
* **MySQL -** relačný databázový systém, slúžiaci na organizovanie dát za pomoci jazyka SQL
* **phpMyAdmin -** nástroj umožňujúci jednoduchú správu obsahu databázy MySQL prostredníctvom webového rozhrania
* **Data-driven Documents (d3.js)** - JavaScriptová knižnica na vytváranie dynamických, interaktívnych vizualizácii údajov (grafov) vo webovom prehliadači
* **leaflet.js -** JavaScriptová knižnica na vytváranie interaktívnych máp v prostredí webového prehliadača
* **bootstrap** - použité na selectpicker dropdown (bootstrap-select), na responzivitu
* **AJAX jQuery** - dropdown menu
* **Numpy, Scipy** - python-ovské knižnice používané pri počítaní korelácii
* **Amazon Lightsail -** VPS(virtual private server), na ktorom beží backend webovej aplikácie
* **Nginx -** softwarový webový server s load manažmentom a reverznou proxy s otvoreným zdrojovým kódom
* **Node.js, npm** - sťahovanie a inštalovanie JavaScript knižníc
* **Webpack (Laravel mix)** - kompilácia .js a .css súborov

## **5.6 Cieľové prostredie nasadenia do prevádzky**

Aplikácia beží na <https://aws.amazon.com/> aktuálne je dostupná pod doménou <http://slovenskovdatach.digital/> .