

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

Softvér pre podporu monitorovania kvality ovzdušia
Dokumentácia

Tomáš Kunzo
Daniel Trizna
Michal Slávik
Frederik Kohár

Obsah

1. Úvod	4
1.1. Účel katalógu požiadaviek	4
1.2 Rozsah projektu	4
1.3 Slovník pojmov	4
1.4 Odkazy na externé stránky	5
2. Všeobecný popis	5
2.1. Perspektíva projektu	5
2.2. Funkcionalita aplikácie	5
2.3. Používateľské roly	6
2.4. Všeobecné obmedzenia	7
3. Špecifické požiadavky	7
3.1 Funkcionálne požiadavky	7
3.1.1 Vizualizácia dát pomocou čiarového grafu	7
3.1.2 Vkladanie vlastných údajov do tabuľky predpovede	8
3.1.3 Mapa	9
3.1.4 Tabuľka hodnôt	10
3.1.5 Vietor	10
3.2 Ostatné požiadavky	10
3.2.1 Dostupnosť	11
4. Implementácia	11
5. Návrh	11
5.1 Špecifikácia vonkajších interfejsov	11
5.2 Konfigurácia aplikácie	11
5.3 Použité technológie	11
5.2 Používateľské rozhranie	16
5.3 Komponent diagram	18
5.3.1 Popis komponentov	18
5.3.1.1 Frontend	18
5.3.1.1.1 HomeView	18
5.3.1.1.1.1 Mapa	18
5.3.1.1.1.2 Graf	18
5.3.1.1.1.3 Tabuľka predpovede	18
5.3.1.1.1.4 Tabuľka staníc	19
5.3.1.1.2 VzduchView	19
5.3.1.2 Backend	19
5.3.1.2.1 Chart	19
5.3.1.2.2 AvgTable	19

5.3.1.2.3 StationTable	19
5.3.1.2.4 Station.....	19
5.3.1.2.5 Pollutant.....	20
5.3.1.2.6 Wind	20
6. Testovacie scenáre.....	24
6.1 Úvod.....	24
6.2 Znečistenie ovzdušia	25
6.2.1 Spustenie aplikácie	25
6.2.2 Zvolenie časového údaju	25
6.2.3 Mapa.....	25
6.2.3.1 Zobrazenie údajov	25
6.2.3.2 Načítanie údajov pre stanicu	25
6.2.4 Graf.....	26
6.2.4.1 Zobrazenie údajov	26
6.2.4.2 Zobrazenie znečisťujúcich látok na grafe.....	26
6.2.4.3 Načítanie údajov pre stanicu	26
6.2.4.4 Zobrazenie znečisťujúcich látok na grafe za n dní	26
6.2.5 Tabuľka vlastnej predpovede.....	26
6.2.5.1 Zobrazovanie údajov	26
6.2.5.2 Vlastné údaje	26
6.2.6 Tabuľka staníc	27
6.2.6.1 Zobrazovanie údajov	27
6.3 Predpoveď vetra	27
6.3.1 Načítanie stránky	27
6.3.2 Komponent ventilačného indexu a komponent predpovede vetra.....	27
6.3.2.1 Zvolenie dátumu.....	28
6.3.2.2 Posúvanie predpovede s krokom	28

1. Úvod

1.1. Účel katalógu požiadaviek

Dokument popisuje presné požiadavky zadávateľa na projekt Softvér pre podporu monitorovania kvality ovzdušia pre Slovenský hydrometeorologický ústav. Tento projekt je vedený Mgr. Pavlom Petrovičom, PhD. a je súčasťou predmetu Tvorba informačných systémov odboru Aplikovaná informatika na Fakulte matematiky, fyziky a informatiky, Univerzity Komenského v Bratislave. Tento dokument je určený rovnako pre predstaviteľov SHMÚ aj pre náš tím. Rovnako je tento dokument pre obe strany záväzný.

1.2 Rozsah projektu

Projektom je webová aplikácia, ktorá slúži na zobrazenie nameraných hodnôt jednotlivých látok, znečisťujúcich ovzdušie, prostredníctvom grafov, tabuliek a mapy. Aplikácia automaticky spracováva dáta a pri niektorých znečisťujúcich látkach počíta kľzavý priemer.

1.3 Slovník pojmov

Django – pythonovská knižnica slúžiaca na vytváranie webových aplikácií

1.4 Odkazy na externé stránky

Farebné škály znečisťujúcich látok -

http://www.shmu.sk/sk/?page=1&id=oko_imis

Štruktúra vizualizovaných dát -

https://docs.google.com/document/d/1uo95iCkiYSIEcKqcRpAVdpCVm1FHCHaM5W0O_34xM_Y/edit?usp=sharing

2. Všeobecný popis

2.1. Perspektíva projektu

Softvér bude získavať údaje o znečistení ovzdušia z databázy, ktorá je na klientovom serveri a zobrazovať ich v prehliadači pre interné potreby pracovníkov SHMÚ. Neobsahuje žiadne exporty a nie je prepojená so žiadnymi inými aplikáciami.

2.2. Funkcionalita aplikácie

Softvér pre podporu monitorovania kvality ovzdušia je webová aplikácia, ktorá spracováva namerané hodnoty znečisťujúcich látok v ovzduší a následne ich graficky znázorňuje na webovej stránke. Aplikácia bude pozostávať z piatich základných komponentov: graf, tabuľka staníc, mapa, tabuľka vlastnej predpovede a interaktívna predpoveď vetra.

Na grafe budú znázornené hodnoty znečisťujúcich látok (ZL) PM_{10} , SO_2 , O_3 , NO_2 a $PM_{2,5}$ v hodinovom intervale pre vybraných 1- 7 dní. Pri znečisťujúcej látke PM_{10} sa vypočíta a znázorní v grafe aj 12 hodinový klzavý priemer a 24 hodinový priemer. Používateľ si bude môcť zvoliť, ktoré znečisťujúce látky sa na grafe zobrazia z ktorých staníc, s tým, že ak používateľ neurčí inak, zobrazené budú znečisťujúce látky PM_{10} , SO_2 , O_3 a NO_2 . V grafe bude znázornená hranica vysokej koncentrácie častíc vo vzduchu (červenou farbou) a hranica zvýšenej koncentrácie častíc vo vzduchu (oranžovou farbou). Nad grafom bude možnosť zvoliť počet na koľko dní dozadu sa zobrazia údaje, rovnako aj výber stanice.

Tabuľka vlastnej predpovede bude používateľom modifikovateľná tabuľka, do ktorej bude môcť vkladať svoje vlastné vymyslené hodnoty látky PM_{10} na najbližších 5 hodín (jednotlivé stĺpce budú aktuálny čas + 1h, aktuálny čas + 2h,

...). Tabuľka bude taktiež obsahovať hodnoty látky PM_{10} za posledných 12 hodín (čiže stĺpce aktuálny čas, aktuálny čas – 1h, ...), ktoré budú už vyplnené podľa údajov z grafu. Po zadaní hodnôt sa vypočíta nový 12 hodinový kĺzavý priemer pre všetky body v grafe vrátane doplnených 5 bodov a zobrazí sa na grafe aj v tabuľke. Tabuľka sa teda bude nachádzať priamo pod grafom.

Tabuľka staníc bude zobrazovať hodnoty všetkých meraných látok na staniciach. Bude sa dať vybrať medzi: „hod“, „max“, „výpadky“. Pri „hod“ bude tabuľka zobrazovať pre každú stanicu hodinové hodnoty každej látky za posledných 24 hodín. Termín (hodina merania) sa bude dať posúvať. Pri zakliknutí „max“ (nastavený bude defaultne) – zobrazí sa maximum za posledných 24 hodín. Hodinové hodnoty aj maximum budú ofarbené podľa rovnakých intervalov ako na webe SHMÚ. Po zakliknutí „výpadky“ sa zobrazí tabuľka s počtom hodín, ktoré vypadli za posledných 24 hodín pri danej stanici a látke. Výpadky dlhšie alebo rovné 2 hodinám sa ofarbia na oranžovo, 12 hodinám na červeno.

Pri mape pôjde o mapu Slovenska, na ktorej sa budú hodnoty látok (taktiež rovnakých ako aj na grafe) zobrazovať v malých krúžkoch na miestach staníc, na ktorých boli namerané, podfarbené podľa hodnoty koncentrácie ZL v závislosti od kvality ovzdušia (veľmi dobrá kvalita – zelená farba, dobrá – svetlozelená, zhoršená – žltá, zlá – oranžová, veľmi zlá - červená). Používateľ si vyberie, ktorá zo znečisťujúcich látok sa má zobraziť a následne sa na mape zobrazia hodinové hodnoty (podľa výberu dátumu času v dropdowne nad mapou) danej stanice pre vybranú ZL, vzťahnuté k poslednému dostupnému termínu alebo termínu vybranému na zobrazenie grafu.

Interaktívna predpoveď vetra bude realizovaná len pomocou obrázkov z databázy. Pôjde o dva typy obrázkov: vietor a ventilačný index. Tieto obrázky sa budú nachádzať pod sebou. Bude možné si pomocou šípok načítavať nové obrázky (oboch typov súčasne), avšak maximálne na 72 hodín dopredu.

Webová aplikácia bude naprogramovaná v programovacom jazyku python, s využitím rozhrania django. Všetky potrebné dáta bude načítavať z databázy klienta. Viac info o štruktúre dát pozri 1.4

2.3. Používateľské roly

Administrátor – jeho úlohou je nainštalovať a nakonfigurovať aplikáciu na webovom serveri

Používateľ – webová aplikácia nebude rozlišovať medzi rôznymi používateľskými rolami a každý používateľ, ktorý má URL aplikácie (dostupnej iba z intranetu SHMÚ) má dostupné všetky jej funkcie bez potreby prihlasovania sa.

2.4. Všeobecné obmedzenia

Webová aplikácia bude fungovať v poslednej verzii ľubovoľného prehliadača Chrome, Firefox, Edge na ľubovoľnom desktopovom operačnom systéme.

3. Špecifické požiadavky

3.1 Funkcionálne požiadavky

3.1.1 Vizualizácia dát pomocou čiarového grafu

1. Na grafe sa budú vykresľovať nasledovné znečisťujúce látky:

a) PM_{10}

- hodinové hodnoty
- 12 hodinový klzavý priemer
- 24 hodinový priemer

b) SO_2

- hodinové hodnoty

c) O_3

- hodinové hodnoty

d) NO_2

- hodinové hodnoty

d) $PM_{2,5}$

- hodinové hodnoty

2. Na grafe bude možné zobrazovať hodnoty pre viacero znečisťujúcich látok naraz.

3. Používateľ si bude môcť znečisťujúce látky zvoliť pomocou checkboxov, ktoré budú zobrazené nad grafom
4. Na grafe budú horizontálne čiary označujúce limitnú hodnotu a informačný/výstražný prah danej znečisťujúcej látky, ktorá bude zobrazená na grafe
5. Hodnoty vykresľované na grafe bude aplikácia čerpať z databázy poskytnutej zadávateľom
6. Graf bude obsahovať hodnoty pre užívateľom zvolenú meraciu stanicu (pozri 3.1.3.6)
7. Meraciu stanicu si bude užívateľ vyberať z dropdown menu nad grafom, alebo prostredníctvom mapy meracích staníc Slovenska (bod 3.1.3)
8. Na stránke bude zobrazený iba jeden graf
9. Y-ová os grafu bude nastavovaná v závislosti od hodnôt v grafe
10. X-ová os grafu bude zobrazovať hodnoty za posledných n dní.
11. Počet dní n bude možné si zvoliť pomocou dropdown výberu z hodnôt od 1 po 7. Taktiež si bude možné zvoliť deň, od ktorého sa týchto n dní bude zobrazovať.
12. V grafe sa nebude dať zoomovať.

3.1.2 Vkladanie vlastných údajov do tabuľky predpovede

1. Pod grafom bude tabuľka, do ktorej bude môcť užívateľ zadať hodnoty, ktoré predpokladá, že budú pravdepodobné a podľa nich sa na grafe upraví len kľzavý priemer. Tabuľka bude mať dva riadky, hodinové hodnoty a 12-hodinový kľzavý priemer, po doplnení pravdepodobných hodnôt na 5 hodín dopredu a doplnia aj predpovedané hodnoty priemeru do 2. riadku tabuľky.
2. Hodnoty doplnené užívateľom nemôžu byť záporné

3. Používateľ môže doplniť hodnoty, aby vedel simulovať vývoj 12 hodinového klzavého priemeru
4. Tabuľka nebude mať žiaden ďalší funkčný účel
5. Tabuľka nebude obsahovať žiadne predvyplnené hodnoty.
6. Tabuľka bude k dispozícii iba pre hodnoty látky PM_{10}
7. Priemer sa vypočíta aj keď zadáme iba 1 hodnotu, prepočíta sa vždy, keď zadáme ďalšiu alebo zmeníme už zadanú.

3.1.3 Mapa

1. Nad grafom bude vykreslená mapa Slovenska so všetkými stanicami
2. V mape sa bude dať zoomovať.
3. Stanica bude vykreslená ako malý krúžok, zafarbený farbou intenzity znečistenia zvolenej ZL.
4. Farby ZL budú zodpovedať tabuľke farieb podľa ich koncentrácií ktorá je dostupná na stránke zadávateľa (pozri 1.4)
5. Po kliknutí na stanicu sa na grafe vykreslí krivka pre práve zobrazovanú znečisťujúcu látku (pozri 3.1.3.4).
6. Graf bude pred kliknutím zobrazovať údaje pre poslednú zvolenú stanicu a ZL
7. V prípade, že bola stránka čerstvo otvorená a teda ešte nebola zvolená žiadna stanica tak, bude graf zobrazovať údaje pre poslednú zvolenú stanicu a ZL
8. V dropdowne bude prednastavený aktuálny deň a hodina.

3.1.4 Tabuľka hodnôt

1. Vedľa mapy bude vykreslená tabuľka rozdielna od vyššie spomínanej

2. Riadky tabuľky budú obsahovať názov stanice, stĺpce tabuľky budú zobrazovať: hodnoty zvoleného parametra pre všetky ZL vzťahnuté k zvolenému časovému údaj, pričom ako parameter sa bude dať zvoliť „hod“, „max“, „výpadky“. Pri „hod“ bude tabuľka zobrazovať pre každú stanicu hodinové hodnoty každej látky za posledných 24 hodín. Pri zakliknutí „max“ (nastavil by som ho defaultne) – sa zobrazí maximum za predchádzajúcich 24 hodín. Hodinové hodnoty aj maximum budú ofarbené podľa rovnakých intervalov ako na webe SHMÚ. Po zakliknutí „výpadky“ sa zobrazí tabuľka s počtom hodín, ktoré vypadli za posledných 24 hodín pri danej stanici a látke. Výpadky dlhšie alebo rovné 2 hodinám sa ofarbia na oranžovo, 12 hodinám na červeno. Tabuľka bude pre všetky stanice a bude zobrazovaná celá, nebude scrollovatelná.

3.1.5 Vietor

1. Na stránke bude možnosť prekliknúť sa na okno zobrazujúce obrázky predpovede vetra a ventilačného indexu

2. Tieto obrázky budú vykreslené pod sebou

3. Na prvom obrázku sa bude zobrazovať vietor

4. Na druhom obrázku sa bude zobrazovať ventilačný index

5. Oba obrázky bude možné si preklikávať súčasne pomocou šípok a zobrazovať si tak predpoveď vývoja vetra a ventilačného indexu na najbližších 72 hodín s voliteľným krokom (1 hodina, 2 hodiny, 3 hodiny, 4 hodiny, 8 hodín). Obrázky sa ako celok obsahujúci všetky informácie o predpovedi stiahnu z databázy a aplikácia ich len zobrazí používateľovi.

3.2 Ostatné požiadavky

3.2.1 Dostupnosť

Aplikácia bude dostupná iba pre užívateľov prístupujúcich k nej pomocou internej VPN ku ktorej majú prístup iba zamestnanci SHMÚ.

Aplikácia bude na serveri s lokálnym doménovým menom, ktoré nie je známe zvonku. Aplikácia nie je určená pre verejnosť iba pre zamestnancov, ktorí majú službu.

4. Implementácia

Diagramy sa budú nachádzať v priložených súboroch.

5. Návrh

5.1 Špecifikácia vonkajších interfejsov

Aplikácia komunikuje s databázovým serverom mysql pomocou django ORM modelov.

5.2 Konfigurácia aplikácie

V konfiguračnom súbore bude potrebné vyplniť nasledujúce údaje:
url databázy, database.username, database.password

5.3 Použité technológie

Python – základná technológia, ktorá bude použitá na systém

leaflet.js - javascriptová knižnica na vykresľovanie máp

Použitie leaflet.js:

Nasledujúci kód pridá na stránku mapu a krúžky pre jednotlivé stanice.

```
{% extends 'chart.html' %}

{% load static %}

{% block map %}

    <link rel="stylesheet" href="{% static 'js/leaflet/leaflet.css' %}" />

    <script src="{% static 'js/leaflet/leaflet.js' %}"></script>

    <div id="map" style="height: 360px; width: 50%;></div>

    <script>

        var map = L.map('map', {

            minZoom: 7.2,
```

```

        zoom: 7.2,
        maxZoom: 9,
        center: [48.60, 19.30],
    });

    map.setMaxBounds([[49.70, 16.00],[47.60, 22.60]])

    var cartodbAttribution = '&copy;';
    <a href="https://www.openstreetmap.org/copyright">OpenStreetMap</a> contributors, &copy;
    <a href="https://carto.com/attribution">CARTO</a>;

    var positron=L.tileLayer('https://{s}.basemaps.cartocdn.com/light_all/{z}/{x}/{y}.png'
    , {

        attribution: cartodbAttribution

    }).addTo(map);

    {% for station in stations %}

    var circle = L.circle([{{ station.lat }}, {{ station.lon }}],{ color: "{{
        station.color_name }}"
        fillColor: "{{ station.color_code }}",
        fillOpacity: 0.5,
        radius: 1000
    }).addTo(map);

    {% endfor %}

</script>
{% endblock map
%}

```

Použitie kódu sa prejaví na stránke takto:



Obr1. Mapa so stanicami

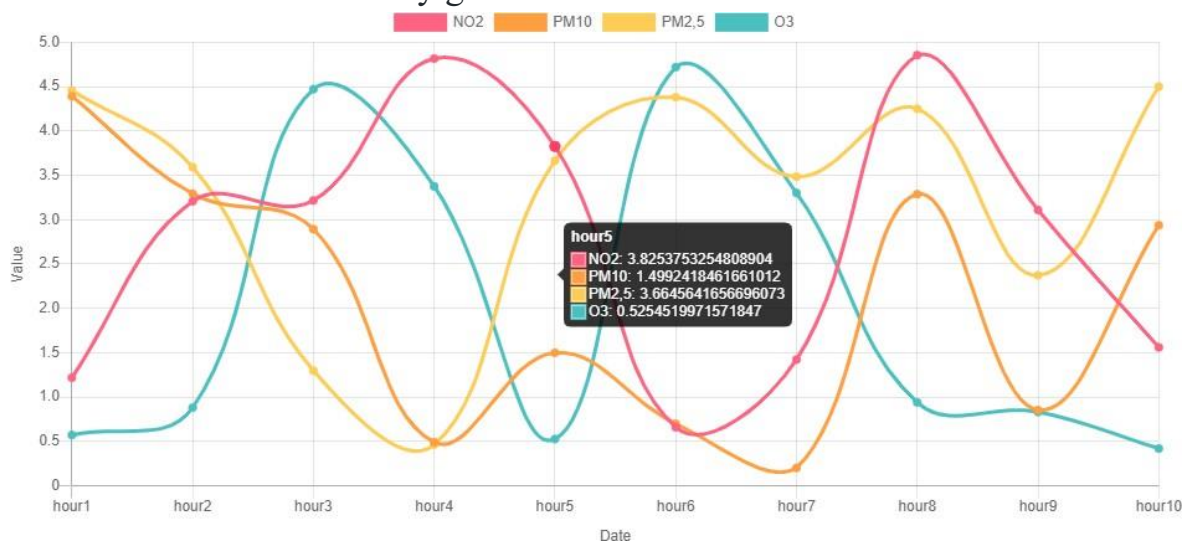
Charts.js:

Príklad Charts.js:

```
{-}
data: {-}
  datasets: (8) [-]
    0: {-}
      _meta: Object { 0: {-} }
      backgroundColor: "rgb(255, 99, 132)"
      borderColor: "rgb(255, 99, 132)"
      data: Array(30) [ "22.7", "25.9", "28.9", - ]
      fill: false
      label: "pm10"
      <prototype>: Object { - }
    1: Object { label: "pm2_5", fill: false, backgroundColor: "rgb(255, 159, 64)", - }
    2: Object { label: "so2", fill: false, backgroundColor: "rgb(255, 205, 86)", - }
    3: Object { label: "no2", fill: false, backgroundColor: "rgb(75, 192, 192)", - }
    4: Object { label: "co", fill: false, backgroundColor: "rgb(54, 162, 235)", - }
    5: Object { label: "o3", fill: false, backgroundColor: "rgb(153, 102, 255)", - }
    6: Object { label: "12-hour", fill: false, backgroundColor: "rgb(153, 255, 204)", - }
    7: Object { label: "24-hour", fill: false, backgroundColor: "rgb(0, 255, 255)", - }
      length: 8
      <prototype>: Array []
    labels: Array(30) [ "29.1.\n16:00", "29.1.\n17:00", "29.1.\n18:00", - ]
      <prototype>: Object { - }
  limits: null
  options: {-}
    animation: Object { duration: 1000, easing: "easeOutQuart", onProgress: noop(), - }
    datasets: Object { bar: {-}, horizontalBar: {-}, scatter: {-} }
    defaultColor: "rgba(0,0,0,0.1)"
    defaultFontColor: "#666"
    defaultFontFamily: "'Helvetica Neue', 'Helvetica', 'Arial', sans-serif"
    defaultFontSize: 12
    defaultFontStyle: "normal"
    defaultLineHeight: 1.2
    elements: Object { arc: {-}, line: {-}, point: {-}, - }
    events: Array(5) [ "mousemove", "mouseout", "click", - ]
    horizontalLine: Array(4) [ {-}, {-}, {-}, - ]
    hover: Object { mode: "nearest", intersect: true, animationDuration: 400, - }
    layout: Object { padding: {-} }
    legend: Object { display: true, position: "top", align: "center", - }
    legendCallback: function legendCallback(t) { }
    maintainAspectRatio: true
    onClick: null
    plugins: Object { filler: {-} }
    responsive: true
    responsiveAnimationDuration: 0
    scales: Object { xAxes: (1) [-], yAxes: (1) [-] }
    showLines: true
    spanGaps: false
    title: Object { display: true, fontStyle: "bold", fullWidth: true, - }
    tooltips: Object { enabled: true, mode: "index", position: "average", - }
    type: "line"
  <prototype>: Object { - }
```

Obr2. Chart.js

Čo nám zobrazí nasledovný graf:



Obr3. Využitie Charts.js

Django - Pomocou pythonovského kódu sme schopný vytvárať webové aplikácie a pripájať ich na databázu, na databázu sa pripája podľa nastavení ktoré sú v aplikácii v záložke vzduch/settings.

Pripojenie na databázu:

```
DATABASES = 'default': {  
    'ENGINE':  
    'django.db.backends.sqlite3',  
    'NAME': BASE_DIR /  
    'db.sqlite3',  
}
```

```
System check identified no issues (0 silenced).  
November 27, 2020 - 10:17:27  
Django version 3.1.2, using settings 'vzduch.settings.base'  
Starting development server at http://127.0.0.1:8000/  
Quit the server with CTRL-BREAK.
```

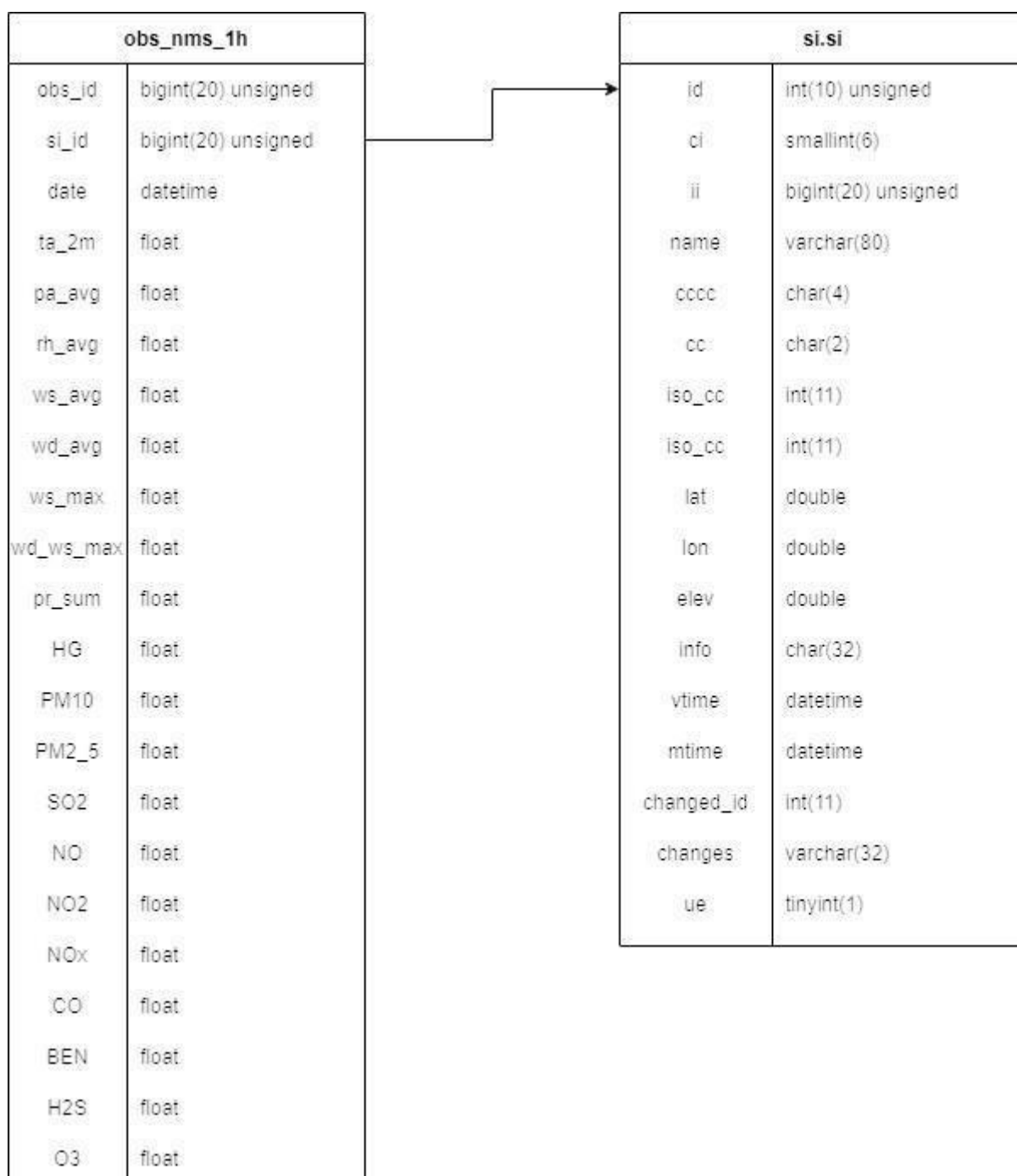
Čo nás pripojí takto:

Obr4. Pripojenie na server

Na počítači je potrebné mať nainštalovaný Python a treba mať nainštalovanú knižnicu Django.

Inštalácia Django: `pip install Django==3.1.3` 5.4.

Dátový model:



Obr5. Relačný model

Dáta o nameraných hodnotách znečisťujúcich látok sa nachádzajú v databáze **obs** v tabuľke **obs_nmsko_1h** v stĺpcoch **PM10, PM2_5, SO2, NO2, O3**.

Nachádzajú sa tu namerané hodnoty každej látky (dátový typ *float*), v ktorý deň a čas (stĺpec **date**) a ktorou stanicou (**si_id** - cudzí kľúč do tabuľky **si.si**).

Dáta o staniaciach sa nachádzajú v databáze **si** v tabuľke **si**. Nachádzajú sa

tu údaje ako názov stanice (stĺpec **name**), číslo stanice (stĺpec **ii**), súradnice stanice (stĺpce **lat** a **lon**).

Na prepojenie týchto dvoch tabuliek, teda na získanie nameraných hodnôt látok spolu aj s názvom stanice, sa použije cudzí kľúč **si_id** v tabuľke **obs_nmsko_1h** odkazujúci sa do tabuľky **si**.

5.2 Používateľské rozhranie

Webová aplikácie bude rozdelená na dve časti: Znečisťujúce látky a predpoveď vetra.

[illegible]

Obr. 6 Znečisťujúce látky

Na podstránke aplikácie s názvom Znečisťujúce látky sa bude nachádzať mapa Slovenska s farebnými guľôčkami zafarbenými podľa miery poškodenia ovzdušia. Nad mapou sa bude nachádzať tlačidlo, ktoré bude slúžiť na výber dňa a času.

Pod mapou bude graf, kde používateľ môže vidieť aké znečistenie bolo za predchádzajúcich niekoľko dní. Nad grafom budú tlačidlá, kde používateľ určí, ktoré znečisťujúce látky chce momentálne zobraziť, môže určiť počet dní ktoré chce zobraziť, alebo vybrať nejakú inú stanicu.

Pod grafom sa bude nachádzať tabuľka pre predpovede, používateľ si bude môcť doplniť svoje vlastné hodnoty a podľa toho počítat priemer.

Napravo od mapy, grafu, tabuľke predpovede sa nachádza tabuľka všetkých staníc aj s danými hodnotami. Tada tabuľkou sa bude nachádzať výberové tlačidlo s parametrami, ako maximum, výpadky, posledné hodnoty. Na posuvnej lište používateľ určí čas ktorý sa chce vrátiť.

Znečisťujúce látky

Predpoveď vetra

23.10.2020 15:00 LSEČ

Zvoliť krok ▼

Vietor

Ventilačný index

◀◀ ◀ ▶ ▶▶

Obr. 7 Predpoveď vetra

V aplikácii v druhej pod sekcii Predpoveď vetra (obr.7) sa budú nachádzať dve časti. Vietor a ventilačný index. Obe tieto časti sú obrázky, kde používateľ vidí obrázky v ten moment, ktorý si on určí. Čas a dátum si zvolí nad obrázkom vetra, rovnako si tam používateľ môže nastaviť krok, o ktorý sa môže posúvať medzi obrázkami. Toto posúvanie medzi obrázkami bude pre používateľa možné pod obrázkom ventilačný index.

5.3 Komponent diagram

Komponent diagram zobrazuje komponenty, ich prepojenia, závislosti a rozhrania, ktoré poskytujú a používajú.

5.3.1 Popis komponentov

Aplikácia je rozdelená na Frontend a Backend, pričom v sebe obsahujú potrebné komponenty.

5.3.1.1 Frontend

Tento komponent obsahuje výstup aplikácie na webe, teda to, čo sa zobrazí v prehliadači. Je zložený z komponentov HomeView a VzduchView.

5.3.1.1.1 HomeView

Tento komponent zahŕňa v sebe ďalšie komponenty – Mapa, Graf, Tabuľka predpovede, Tabuľka staníc.

5.3.1.1.1.1 Mapa

Komponent Mapa v sebe zahŕňa interaktívny prvok, kedy po prejdení myšou ponad stanicu zobrazí aktuálne hodnoty pre danú stanicu. Zahŕňa v sebe taktiež ovládací panel na nastavenie dátumu a času, kde sa podľa nastavenia aktualizujú stanice v mape.

5.3.1.1.1.2 Graf

Komponent Graf v sebe zahŕňa ovládacie prvky na nastavenie počtu dní a stanice, pre ktorú si chce zobrazit' graf. Taktiež má na výber z jednotlivých znečisťujúcich látok. Komponent obsahuje aj výber horizontálnych limitov pre danú látku. Používateľ si môže zvolit' typ grafu (bodkovaný, zaoblený, priame čiary).

5.3.1.1.1.3 Tabuľka predpovede

V komponente tabuľka predpovede sa nachádza tabuľka pre danú stanicu za posledných 12 hodín, pre každú hodinu je vypočítaný priemer. Používateľ môže do tabuľky napísať 5 nasledujúcich hodnôt pre 5 nasledujúcich hodín, pričom sa mu vždy vypočíta pod pridanou hodnotou priemer.

5.3.1.1.1.4 Tabuľka staníc

V komponente staníc sa nachádza výber medzi maximálnou hodnotou, hodinovou hodnotou a výpadkami, kedy stanica nemerala. Sú to zobrazené všetky stanice a znečisťujúce látky. V posuvnej lište sa používateľ môže posúvať v čase od aktuálneho času, po čas 24 hodín dozadu.

5.3.1.1.2 VzduchView

Tento komponent v sebe zahŕňa dva ďalšie komponenty – Vietor a Ventilačný index. Komponent Vietor zobrazuje obrázky vetra, komponent Ventilačný index zobrazuje obrázky ventilačného indexu. Pre obidva komponenty sa tu nachádzajú dva ovládacie panely. V prvom si užívateľ vyberá dátum a čas, pre ktorý chce zobraziť obrázky. Môže si tu určiť aj krok v hodinách, o ktorý si bude obrázky posúvať v druhom ovládacom paneli. V ňom sa môže posúvať dopredu alebo dozadu o určený krok.

5.3.1.2 Backend

Backend sa stará o generovanie údajov podľa požiadaviek používateľa, ktoré následne posiela na frontend.

5.3.1.2.1 Chart

Chart spracuje požiadavky na dátum a čas, dané látky a danú stanicu a posiela na prehliadač.

5.3.1.2.2 AvgTable

AvgTable počíta priemerné hodnoty látok každú hodinu, pri dopísaní ďalších hodnôt ich prepočíta na priemer.

5.3.1.2.3 StationTable

StationTable spracuje požiadavky na dátum a čas, dané látky a danú stanicu a posiela na prehliadač.

5.3.1.2.4 Station

Definuje potrebné údaje jednotlivých staníc.

5.3.1.2.5 Pollutant

Definuje potrebné údaje jednotlivých látok

5.3.1.2.6 Wind

Spracováva požiadavky a vyberá obrázok, ktorý pošle na frontend.

5.4 Komunikačný protokol medzi klientom a serverom

Naša stránka poskytuje dve rôzne funkcionality, zobrazovanie znečisťujúcich látok (viď obr.8) a zobrazovanie predpovede vetra (viď obr.9).

Pri znečisťujúcich látkach má klient(prehliadač) dve možnosti ako sa dopytovať na server.

Prvá možnosť je prostredníctvom HTTP GET requestu, ktorý nemá žiadnu nami stanovenú štruktúru, nakoľko z getu žiadne parametre nečerpáme.

Po prijatí get requestu server dopytom získa dáta z databázy pre dátum získaný z posledného post requestu. V prípade že žiadny dátum nebol zadáný, server získa dáta pre aktuálny dátum. Server získané dáta uloží do JSON štruktúry, ktorá má v prípade znečisťujúcich látok nasledovný formát:

```
{
  "ZnečisťujúcaLátka1" : {
    "stanica1" : [hodnota1,hodnota2, ... ],
    "stanica2" : [hodnota1,hodnota2, ... ],
    ...
  },
  "ZnečisťujúcaLátka2" : {
    "stanica1" : [hodnota1,hodnota2, ... ],
    "stanica2" : [hodnota1,hodnota2, ... ],
    ...
  },
  ...
}
```

kde znečisťujúca látka je jedna z látok ktoré zadávať sleduje, stanica je jedno z miest kde sa vykonávajú merania a hodnota je hladina danej znečisťujúcej látky v danej stanici za konkrétnu hodinu. Hodnoty sú usporiadané podľa času merania s tým, že posledná hodnota zoznamu predstavuje najaktuálnejšiu hodnotu.

Pre stanice je formát nasledovný:

```
{
  "Stanica1" : {
    "name" : "Stanica1",
    "lat" : hodnota,
    "lon" : hodnota
  },
  "Stanica2" : {
    "name" : "Stanica2",
    "lat" : hodnota,
    "lon" : hodnota
  },
  ...
}
```

Stanica je jedno z miest kde sa vykonávajú merania, name je názov stanice, lat a lon predstavujú geografické súradnice danej stanice.

Tieto dáta následne server zaobalí do jedného JSONu ktorý má dva kľúče a to stations, kde hodnota je JSON so stanicami a data, kde hodnota je JSON so znečisťujúcimi látkami. Takto zaobalené dáta server odošle klientovi pomocou django funkcie `django.shortcuts.render`, ktorá tieto dáta vyrenderuje do nami vytvoreného templatu. Hlavná časť templatu ktorý sa renderuje vyzerá nasledovne:

```

<div class="main-wrapper">
  <div class="app">
    <header class="header">
      <div class="header-block header-block-nav">
        <a href="/vzduch" style="padding: 6px 20px; border: 1px solid #969696; border-radius: 5px;">
          Wind prediction
        </a>
      </div>
    </header>
    <div class="mobile-menu-handle"></div>
    {% block content %}
      <article class="content buttons-page">
        <div class="section">
          <div class="row">
            <div class="col-md-8">
              <div class="card-block">
                {% block dateForm %}
                {% endblock dateForm %}
                {% block map %}
                {% endblock map %}
              </div>
              <div class="card-block">
                {% block chart %}
                {% endblock chart %}
              </div>
              <div class="card-block">
                {% block avgTable %}
                {% endblock avgTable %}
              </div>
            </div>
            <div class="col-md-4">
              {% block stationsTable %}
              {% endblock stationsTable %}
            </div>
          </div>
        </article>
      {% endblock content %}
    </div>
  </div>

```

Takto vyrenderovaný template sa pomocou HTTP protokolu odošle klientovi, ktorému sa zobrazí výsledná stránka (vid' obr.8).

Ďalšia možnosť je prostredníctvom HTTP POST requestu. Tento request by mal obsahovať jeden parameter date, ktorý predstavuje korektne vyplnený formulár pre dátum, pre ktorý sa majú dáta načítať. Tento formulár je realizovaný pomocou triedy, ktorá dedí z `django.forms.Form`. Po prijatí post requestu si post dáta zapamätáme do globálnej premennej a následne server vytvorí nový formulár, ktorý obsahuje údaje z globálnej premennej nasledovne:

```
date_form = DateForm(POST_DATA, auto_id=False)
```

Z takto vytvoreného formuláru získame používateľom zadaný dátum nasledovne:

```

date = datetime.datetime.now()
if date_form.is_valid():
    date = date_form.cleaned_data.get("date") if date_form.cleaned_data.get("date") is not None else date

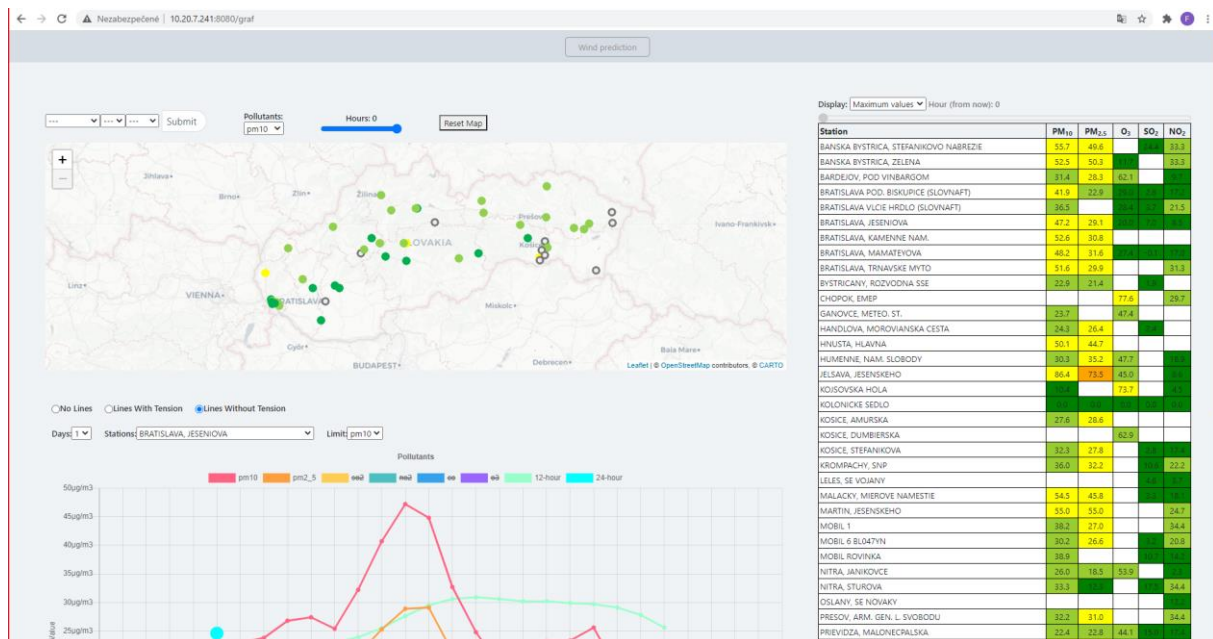
```

Tento dátum následne server využije na získanie dát z databázy. Tieto dáta server uloží do JSON formátu rovnakým spôsobom ako pri get requeste a rovnako ich aj odošle (vid' obr.8).

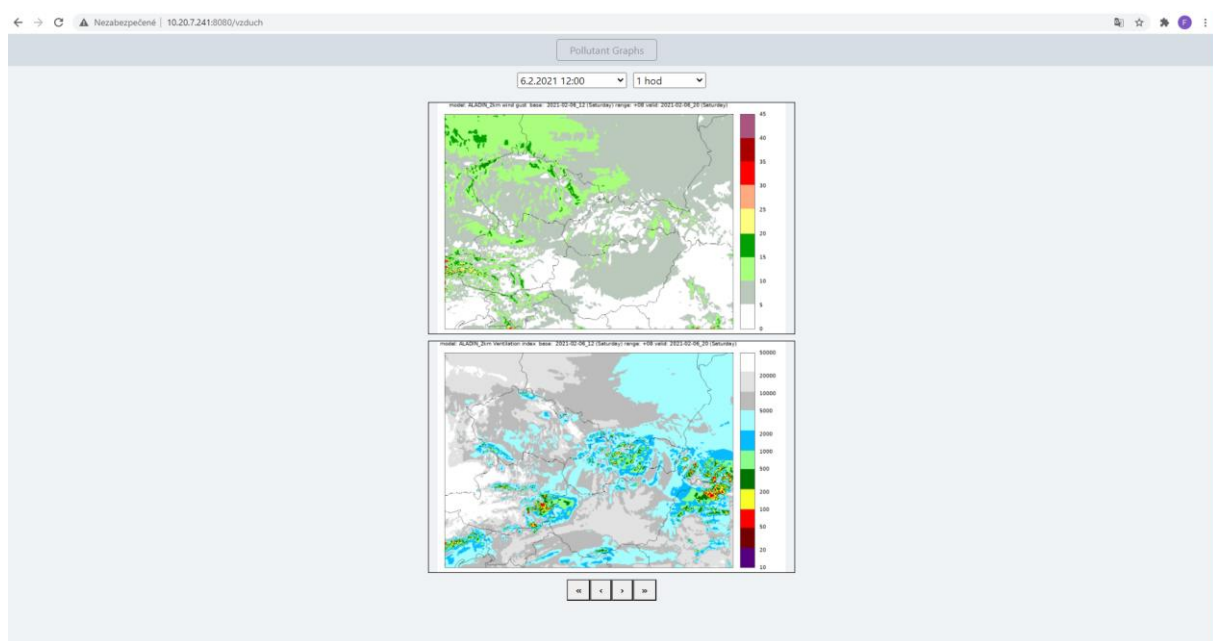
Pri vetre má klient možnosť dopytovať sa na server iba prostredníctvom HTTP GET requestu, ktorý nemá žiadnu nami stanovenú štruktúru, nakoľko z getu žiadne parametre nečerpáme. Po prijatí get requestu server načíta dáta potrebné na zobrazenie obrázkov týkajúcich sa predpovedi vetra. Tieto dáta následne uloží v JSON formáte, ktorý má nasledovnú štruktúru:

```
{
  'DIR_DATES': ["nazov1", "nazov2", ...],
  'SELECT_DATES': ["datum1", "datum2", ...],
  'PICTURES': {
    "dátum1" : {
      "GUST" : ["cesta k obr1", "cesta k obr2", ...],
      "VEID" : ["cesta k obr1", "cesta k obr2", ...],
    },
    "dátum2" : {
      "GUST" : ["cesta k obr1", "cesta k obr2", ...],
      "VEID" : ["cesta k obr1", "cesta k obr2", ...],
    },
    ...
  },
  'PATH': "cesta k obrázkom"
}
```

Pričom PATH je cesta k hlavnému priečinku, v ktorom sa nachádzajú podpriečinky s názvami nazov1, nazov2, ... Z týchto názvov sa získavajú dátumy datum1, datum2, ... a následne sa vygenerujú cesty k jednotlivým obrázkom, ktoré sú uložené pod kľúčom PICTURES. Rovnako ako pri znečisťujúcich látkach sa takto spracované údaje vyrenderujú pomocou django funkcie render a pomocou http protokolu sa odošlú klientovi (viď obr.9).



Obr. 8



Obr. 9

6. Testovacie scenáre

6.1 Úvod

Reflexia:

Pri vykonávaní každého testovacieho scenáru si prosím položte aj nasledujúce otázky:

- Funguje stránka správne?

- Prekrývajú sa niektoré komponenty?
- Reagujú všetky komponenty na váš vstup?
- Rozumiete fungovaniu jednotlivých komponentov?
- Fungujú všetky komponenty podľa vašich očakávaní?

6.2 Znečistenie ovzdušia

6.2.1 Spustenie aplikácie

Scenár: Spustenie aplikácie

Očakávaný výstup: Aplikácia načíta na úvodnej stránke komponent mapy, grafu, tabuľky vlastnej predpovede a tabuľky staníc spolu so všetkými dropdown komponentami a checkboxami. Takisto sa načíta aj tlačítko v hornej časti obrazovky na prepnutie sa do druhého okna s predpoveďou vetra.

6.2.2 Zvolenie časového údaju

Scenár: Zvoľte jeden z ponúknutých časových údajov v dropdowne nad mapou.

Očakávaný výstup: Načítanie údajov pre zvolený časový údaj pre stanice na mape, v grafe a v tabuľke vlastnej predpovede.

6.2.3 Mapa

6.2.3.1 Zobrazenie údajov

Scenár: Prečítanie údajov zo staníc na mape.

Očakávaný výstup: Mapa zobrazuje stanice ako krúžky, ktoré sú adekvátne zafarbené podľa najhoršej koncentrácie jednej z látok PM_{10} , SO_2 , O_3 , NO_2 , $PM_{2.5}$ nameranej na danej stanici za posledných 24 hodín od dátumu času zvoleného v dropdowne nad mapou. Po prejdení myšou ponad stanicu sa zobrazí názov látky s najhoršou koncentráciou.

6.2.3.2 Načítanie údajov pre stanicu

Scenár: Kliknite na jednu zo staníc na mape.

Očakávaný výstup: Na komponente grafu a tabuľky vlastnej predpovede zobrazia údaje pre túto stanicu.

6.2.4 Graf

6.2.4.1 Zobrazenie údajov

Scenár: Graf zobrazuje údaje pre správne látky.

Očakávaný výstup: Na grafe sa pre látku PM_{10} zobrazujú hodinové hodnoty, 12 hodinový kľzavý priemer a 24 hodinový priemer. Ďalej sa zobrazujú hodinové hodnoty pre látky SO_2 , O_3 , NO_2 , PM_2 .

6.2.4.2 Zobrazenie znečisťujúcich látok na grafe

Scenár: Zaškrtnite niektoré checkboxy nad grafom.

Očakávaný výstup: Na grafe sa zobrazujú údaje len pre zaškrtnuté látky.

6.2.4.3 Načítanie údajov pre stanicu

Scenár: Zvoľte stanicu v dropdowne nad grafom.

Očakávaný výstup: Na komponente grafu a tabuľky vlastnej predpovede zobrazia údaje pre túto stanicu.

6.2.4.4 Zobrazenie znečisťujúcich látok na grafe za n dní

Scenár: Zvoľte počet dní z dropdownu nad grafom.

Očakávaný výstup: Na grafe sa zobrazujú údaje len pre zaškrtnuté látky za zvolený počet dní.

6.2.5 Tabuľka vlastnej predpovede

6.2.5.1 Zobrazovanie údajov

Scenár: Prečítanie údajov z tabuľky.

Očakávaný výstup: Tabuľka zobrazuje v prvom riadku hodinové koncentrácie látky PM_{10} na-merané na zvolenej stanici za posledných 12 hodín. V druhom riadku zobrazuje kľzavý priemer hodinových koncentrácií látky PM_{10} za posledných 12 hodín.

6.2.5.2 Vlastné údaje

Scenár: Doplňte vlastné hodnoty do niektorých polí v prvom riadku.

Očakávaný výstup: Tabuľka ponúka v prvom riadku miesto na doplnenie vlastných hodinových koncentrácií na najbližších 5 hodín pre zvolenú stanicu. Po doplnení hodnoty pre ktorúkoľvek hodinu sa na príslušné miesto v spodnom riadku vypočíta kľzavý priemer zahrňajúci aj túto novú hodnotu a na grafe sa zobrazia hodnoty tohto kľzavého priemeru.

6.2.6 Tabuľka staníc

6.2.6.1 Zobrazovanie údajov

Scenár: Prečítanie údajov z tabuľky staníc.

Očakávaný výstup: Tabuľka zobrazuje v ľavom stĺpci názvy všetkých staníc, na pravo zobrazuje hodnoty pre zvolený parameter v dropdowne (predvolený je parameter *MAX*). Tieto hodnoty zobrazuje pre všetky znečisťujúce látky vzťahnuté k zvolenému časovému údaju.

6.2.6.2 Údaje pre parameter *HOD*

Scenár: Zvolenie parametra *HOD*.

Očakávaný výstup: Tabuľka pre parameter *HOD* zobrazuje hodinové hodnoty každej látky za posledných 24 hodín.

6.2.6.3 Údaje pre parameter *MAX*

Scenár: Zvolenie parametra *MAX*.

Očakávaný výstup: Tabuľka pre parameter *MAX* zobrazí najvyššiu nameranú hodnotu každej látky za posledných 24 hodín. Hodinové hodnoty aj maximum budú ofarbené podľa rovnakých intervalov ako na webe SHMÚ.

6.2.6.4 Údaje pre parameter *VÝPADKY*

Scenár: Zvolenie parametra *VÝPADKY*.

Očakávaný výstup: Tabuľka pre parameter *VÝPADKY* zobrazuje počet hodín, ktoré vypadli za posledných 24 hodín pri danej stanici a látke. Výpadky dlhšie alebo rovné 2 hodinám sa ofarbujú na oranžovo, 12 hodinám na červeno.

Tabuľka bude pre všetky stanice a bude zobrazovaná celá, nebude scrollovateľná.

6.3 Predpoveď vetra

6.3.1 Načítanie stránky

Scenár: Otvorte stránku s predpoveďou vetra a ventilačným indexom.

Očakávaný výstup: Stránka načíta dva dropdowny (jeden pre dátumčas a druhý pre veľkosť kroku) a dva komponenty s obrázkami v strede obrazovky. Pod obrázkami sa nachádzajú aj tlačidlá na posúvanie predpovede.

6.3.2 Komponent ventilačného indexu a komponent predpovede vetra

Komponenty zobrazujú predpoveď na 72 hodín dopredu od zvoleného dátumčasu v dropdowne.

6.3.2.1 Zvolenie dátumu

Scenár: Zvoľte dátum v prvom dropdowne.

Očakávaný výstup: Načítajú sa obrázky predpovede v oboch komponentoch súčasne pre zvolený dátum.

6.3.2.2 Posúvanie predpovede s krokom

Scenár: Zvoľte časový krok v druhom dropdowne a tlačidlami pod obrázkami posúvajte predpo-ved'.

Očakávaný výstup: Obrázky sa menia súčasne v oboch komponentoch podľa zvoleného kroku.

Krok môže byť 1 hodina, 2 hodiny, 3 hodiny, 4 hodiny a 8 hodín.