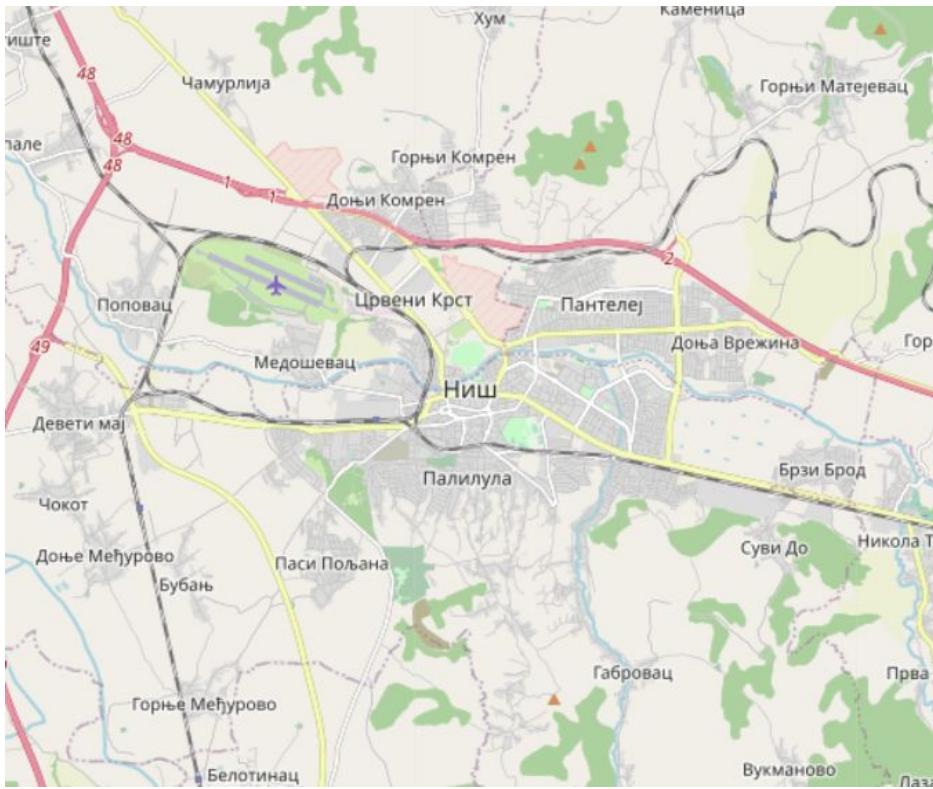


Geografski Informacioni Sistemi

- Projekat 1
OSM, PostGIS, i QGIS
- Projekat 2
GeoServer, Leaflet, App
- Projekat 3
GIS

Podaci



OSM - Elements

Elementi:

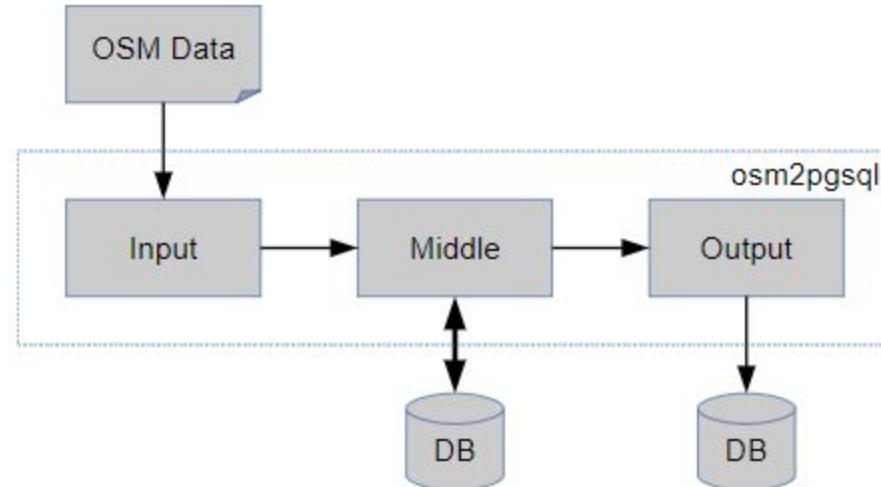
- Node 
- Way 
- Relation 
- Tag 

```
buildings:  
    types:  
        - polygons  
    select:  
        - name  
        - building  
    where: building IS NOT NULL  
waterways:  
    types:  
        - lines  
        - polygons  
    select:  
        - name  
        - waterway  
    where: natural IN ('waterway')  
hospitals:  
    select:  
        - name  
        - amenity  
    where: amenity = "hospital"
```

Osm

- Preuzeta oblast za projekat je "Nis"
- OSM podaci su skinuti putem sajta HotExport u osm formatu (osm.pbf)
- b9f24ab2-5694-4c55-9894-6607263adb80 - HotExport ID
- Podaci konvertovani i importovani u PostGIS koriscenjem osm2pgsql alata

- **osm2pgsql -c -d osm -U postgres -H localhost path_to_os.pbf**
 - -c = create
 - -d = database
 - -U = username
 - -H = host



PostGIS

- PostGIS je prostorno proširenje PostgreSQL baze podataka
- Pored prezentacija sa predavanja korištena je "Introduction to PostGIS" radionica kao referenca i tutorijal - <https://postgis.net/workshops/postgis-intro/index.html>
- Podržava geometrijske tipove podataka u skladu sa OGC Simple Features specifikacijom
- Obezbedjuje funkcije koje su dostupne u okviru SQL u skladu sa OGC Simple Features specifikacijom
- PostGIS koristi Quadtree i R tree kao indeksne strukture

Slojevi

Preuzeti podaci su organizovani u 3 tabele:

- planet_osm_point
- planet_osm_line
- planet_osm_polygon

Nakon importovanja u bazu, podaci su reorganizovani u slojeve, nad kojima su konstruisani prostorni indexi, koji su kasnije korišteni prilikom razvoja projekata

- | | |
|--|---|
| 1. Landuse polygon - zemljište | 10. Highway line - putevi |
| 2. Natural polygon - prirodne površine | 11. Power line - električna mreža |
| 3. Power polygon - transformatori,... | 12. Fortress boundary - tvrđava |
| 4. Leasure polygon - odmarališta | 13. Railway line - pruge |
| 5. Highway polygon - površine za kretanje | 14. Power point - dalekovodi,... |
| 6. Amenity polygon - razno | 15. Historic point - istoriske tačke |
| 7. Building polygon - zgrade, kuće,.. | 16. Amenity point - razno |
| 8. Historic polygon - istorijske površine | 17. Shop point - prodavnice |
| 9. Railway polygon - objekti železnice | 18. Public transport point - stanice |

Slojevi

Preuzeti su i vremenski podaci u vidu tab-delimited tekstualnih tabela

- AVL_DataPoints.txt
- OBD_DataPoints.txt

Ovi podaci su importovani u PostGIS:

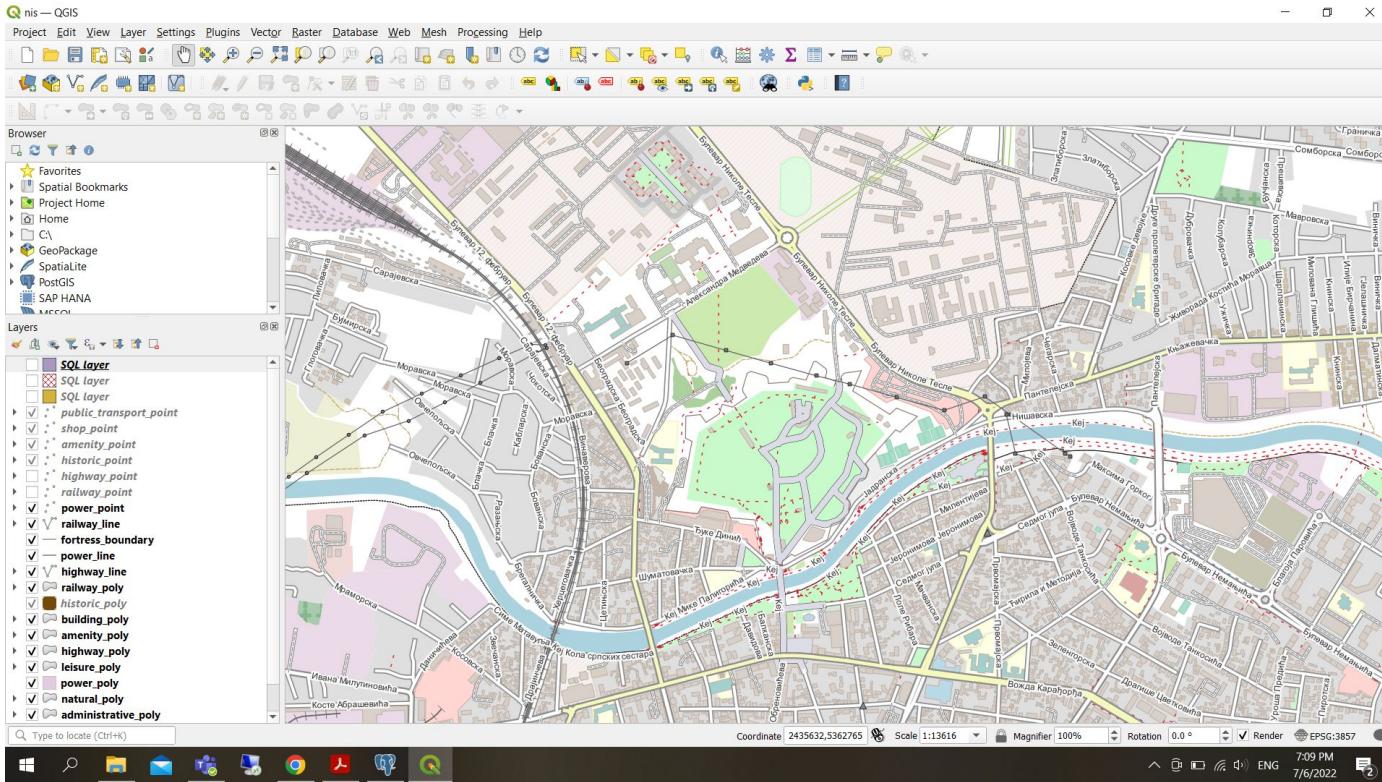
1. TDV \Rightarrow CSV
2. Rucno kreirana tabela
3. Kopiranje podataka iz fajla u tabelu

```
\copy new_table(columns) FROM '/local/path/to/Individual_Landmarks.csv'  
DELIMITERS ',' CSV HEADER;
```

4. Kreiranje geometrije i prilagodjavanje projekcije: 4326 \Rightarrow 3857

```
UPDATE landmarks  
SET the_geom = ST_Transform(ST_GeomFromText('POINT(' || longitude || ' ' ||  
latitude || ')'), 4326), 3857);
```

QGIS



Upiti - Towers

```
SELECT count(*) as num_power, p2.osm_id, p2.building,  
p2.way  
FROM planet_osm_point as p1  
JOIN planet_osm_polygon as p2  
    ON ST_DWithin(p1.way,p2.way, 50)  
WHERE p1.power='tower' AND not p2.building is null  
GROUP BY p2.osm_id, p2.building, p2.way  
ORDER BY num_power DESC;
```

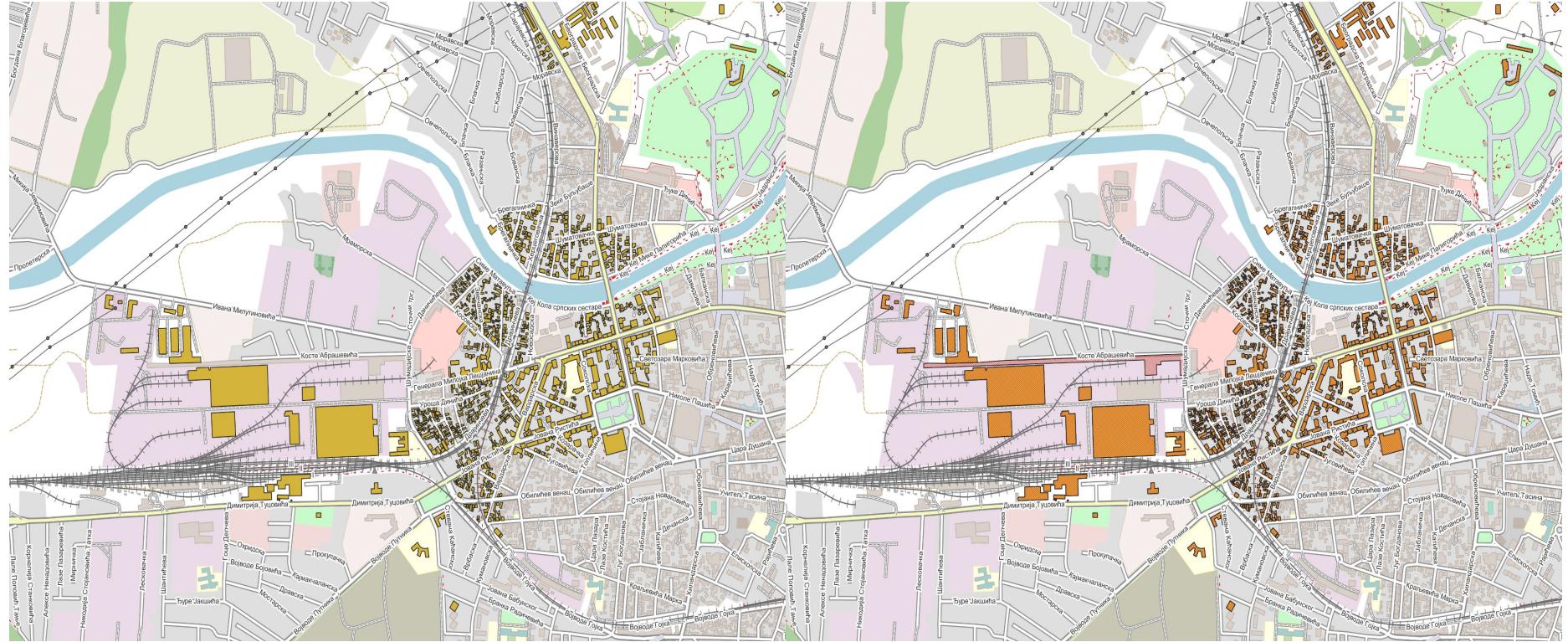


Upiti - Military

```
SELECT
    DISTINCT(pb.osm_id),
    pb.way
FROM planet_osm_polygon as pb
JOIN planet_osm_polygon as pm
    ON ST_DWithin(pm.way, pb.way, 400)
WHERE
pb.osm_id not in
    (SELECT pb.osm_id
    FROM planet_osm_polygon as pb
    JOIN planet_osm_polygon as pm
        ON ST_Intersects(pb.way, pm.way)
    WHERE not pb.building is null
        AND pm.landuse='military')
AND not pb.building is null
AND pm.landuse='military';
```

```
SELECT
    DISTINCT(pb.osm_id),
    pb.way
FROM planet_osm_polygon as pb
JOIN planet_osm_polygon as pm
    ON ST_DWithin(pm.way, pb.way, 400)
WHERE
pb.osm_id not in
    (SELECT pb.osm_id
    FROM planet_osm_polygon as pb
    JOIN planet_osm_polygon as pm
        ON ST_Contains(pb.way, pm.way)
    WHERE not pb.building is null
        AND pm.landuse='military')
AND not pb.building is null
AND pm.landuse='military';
```

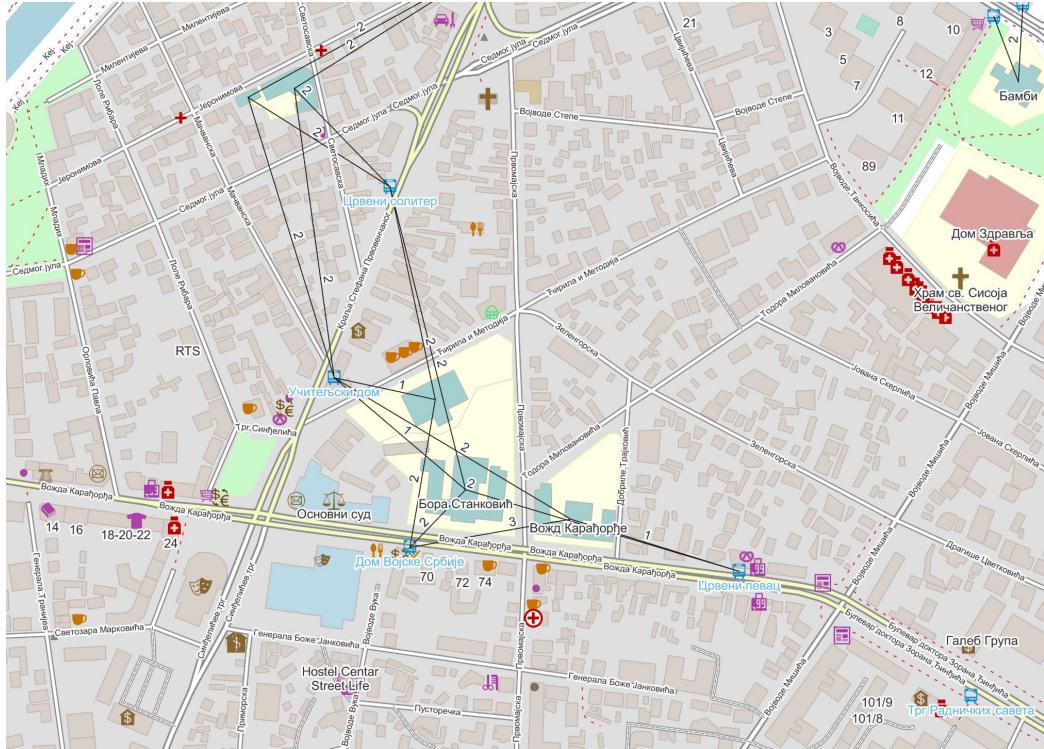
Upiti - Military



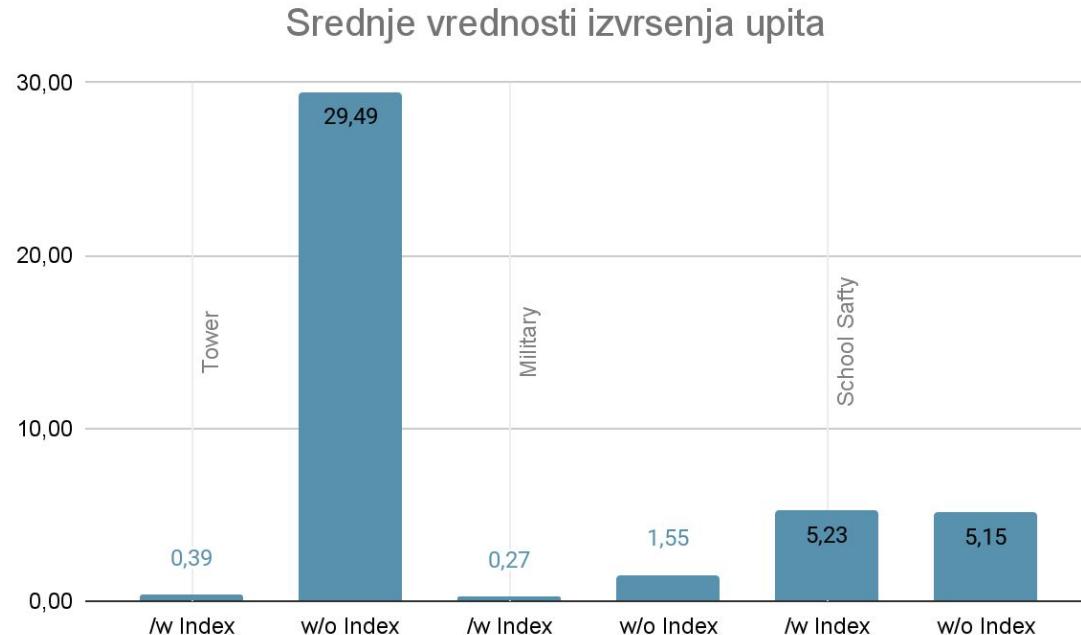
Upiti - School Safty

```
SELECT
    ROW_NUMBER() OVER (ORDER BY 1) AS id,
    paths.platform_id,
    paths.platform_name,
    paths.school_id,
    paths.school_name,
    COUNT(CASE WHEN ST_Crosses(l.way, paths.bird_pth) THEN 1 END),
    paths.bird_pth as way
FROM
    (SELECT
        p1.osm_id as platform_id,
        p1.name as platform_name,
        p2.osm_id as school_id,
        p2.name as school_name,
        ST_MakeLine(ST_Centroid(p2.way),p1.way) as bird_pth
    FROM planet_osm_point as p1
    JOIN planet_osm_polygon as p2
        ON ST_DWithin(ST_Centroid(p2.way),p1.way, 400)
    WHERE p1.public_transport='platform' AND p2.building='school') as paths
CROSS JOIN planet_osm_line as l
WHERE l.highway in ('residential', 'secondary', 'secondary_link', 'tertiary', 'tertiary_link', 'unclassified')
GROUP BY platform_id, platform_name, school_id, school_name, bird_pth;
```

Upiti - School Safty



Prostorni index i izvrsenje upita



Projekat 2 - UI

Page Title

File | C:/Users/Sreten/Documents/Master%20Studije/gis/msc-gis/project-2/main.html

YouTube Akira Kurosawa Bullshitary Workout Music Suspicious Recepti Courses X Само Ме Обриши... Have I been pwned... Object-Oriented Pr... Educational websites Bayesian Thinking L...

Attribute Value

osm_id	689098717
addr:housenumber	14
amenity	college
building	yes
name	Електронски факултет
operator	Универзитет у Нишу
z_order	0
way_area	8489.6
id	17336

landuse_poly

- grass
- industrial
- meadow
- military
- residential
- leisure_poly
- amenity_poly
- college
- university
- building_poly
- college
- YES
- restaurant
- highway_line
- footway
- living_street
- path
- pedestrian
- secondary
- service
- unclassified
- amenity_point
- post_office
- public_transport_point
- platform

Leaflet

Electronski fakultet

Electronski fakultet

Студентски ресторант

Студентски ресторант

Ресторан Индекс

Александра Малинова

Александра Малинова

Page Title

File | C:/Users/Sreten/Documents/Master%20Studije/gis/msc-gis/project-2/main.html

YouTube Akira Kurosawa Bullshitary Workout Music Suspicious Recepti Courses X Само Ме Обриши... Have I been pwned... Object-Oriented Pr... Educational websites Bayesian Thinking L...

7:24 PM 7/6/2022

Projekat 2 - Geoserver

Layers

Manage the layers being published by GeoServer

[Add a new layer](#) [Remove selected layers](#)

<< < 1 2 > >> Results 1 to 25 (out of 39 items)

Type	Title	Name	Store	Enabled	Native SRS
<input type="checkbox"/>	<input checked="" type="checkbox"/> administrative_poly	nis:administrative_poly	nis_store		EPSG:3857
<input type="checkbox"/>	<input checked="" type="radio"/> amenity_point	nis:amenity_point	nis_store		EPSG:3857
<input type="checkbox"/>	<input checked="" type="checkbox"/> amenity_poly	nis:amenity_poly	nis_store		EPSG:3857
<input type="checkbox"/>	<input checked="" type="checkbox"/> avl_datapoints	nis:avl_datapoints	nis_store		EPSG:3857
<input type="checkbox"/>	<input checked="" type="checkbox"/> building_poly	nis:building_poly	nis_store		EPSG:3857
<input type="checkbox"/>	<input checked="" type="checkbox"/> fortress_boundary	nis:fortress_boundary	nis_store		EPSG:3857
<input type="checkbox"/>	<input checked="" type="checkbox"/> highway_line	nis:highway_line	nis_store		EPSG:3857
<input type="checkbox"/>	<input checked="" type="checkbox"/> highway_poly	nis:highway_poly	nis_store		EPSG:3857
<input type="checkbox"/>	<input checked="" type="radio"/> historic_point	nis:historic_point	nis_store		EPSG:3857
<input type="checkbox"/>	<input checked="" type="checkbox"/> historic_poly	nis:historic_poly	nis_store		EPSG:3857
<input type="checkbox"/>	<input checked="" type="checkbox"/> landuse_poly	nis:landuse_poly	nis_store		EPSG:3857
<input type="checkbox"/>	<input checked="" type="checkbox"/> leisure_poly	nis:leisure_poly	nis_store		EPSG:3857
<input type="checkbox"/>	<input checked="" type="checkbox"/> natural_poly	nis:natural_poly	nis_store		EPSG:3857
<input type="checkbox"/>	<input checked="" type="checkbox"/> power_line	nis:power_line	nis_store		EPSG:3857
<input type="checkbox"/>	<input checked="" type="radio"/> power_point	nis:power_point	nis_store		EPSG:3857
<input type="checkbox"/>	<input checked="" type="checkbox"/> power_poly	nis:power_poly	nis_store		EPSG:3857
<input type="checkbox"/>	<input checked="" type="radio"/> public_transport_point	nis:public_transport_point	nis_store		EPSG:3857
<input type="checkbox"/>	<input checked="" type="checkbox"/> railway_line	nis:railway_line	nis_store		EPSG:3857
<input type="checkbox"/>	<input checked="" type="checkbox"/> railway_poly	nis:railway_poly	nis_store		EPSG:3857
<input type="checkbox"/>	<input checked="" type="radio"/> shop_point	nis:shop_point	nis_store		EPSG:3857
<input type="checkbox"/>	<input checked="" type="checkbox"/> World rectangle	tiger:giant_polygon	nyc		EPSG:4326

Search

About & Status

- [Server Status](#)
- [GeoServer Logs](#)
- [Contact Information](#)
- [About GeoServer](#)

Data

- [Layer Preview](#)
- [Workspaces](#)
- [Stores](#)
- [Layers](#)
- [Layer Groups](#)
- [Styles](#)

Services

- [WMTS](#)
- [WCS](#)
- [WFS](#)
- [WMS](#)

Settings

- [Global](#)
- [Image Processing](#)
- [Raster Access](#)

Tile Caching

- [Tile Layers](#)
- [Caching Defaults](#)
- [Gridsets](#)
- [Disk Quota](#)
- [BlobStores](#)

Security

- [Settings](#)
- [Authentication](#)
- [Passwords](#)
- [Users Groups Roles](#)

Projekat 2 - kod, leaflet

```
var map = L.map('map').setView([43.32, 21.88], 13);
var popup = L.popup();
wms_tileLayer =
L.tileLayer.wms("http://localhost:8080/geoserver/nis/wms",{
    layers: ["nis_bato"],
    format: 'image/png',
    transparent: true
})
wms_tileLayer.addTo(map)
```

- Kreira se objekat mape i centrira se na koordinatne nisa u WGS84 projekciji koja se koristi u leafletu.
- Kreira se "TileLayer" objekat leafleta koji crta mapu sa WMS servisa

Projekat 2 - kod, leaflet

'http://localhost:8080/geoserver/nis/wms'

- service: 'wms',
- version: '1.1.1',
- request: 'GetLegendGraphic',
- format: 'image/png',
- width: '20',
- height: '20',
- layer: 'nis_bato',
- scale: `\${scale}`,
- bbox: `\${bbox_arr}`,
- srs: 'EPSG:3857',
- srcwidth: '512',
- srcheight: '512',
- legend_options: 'hideEmptyRules:true'

- legenda postoji kao request u okviru WMS servisa
- Request vraća sliku koja se embeduje unutar html-a

Projekat 2 - kod, leaflet

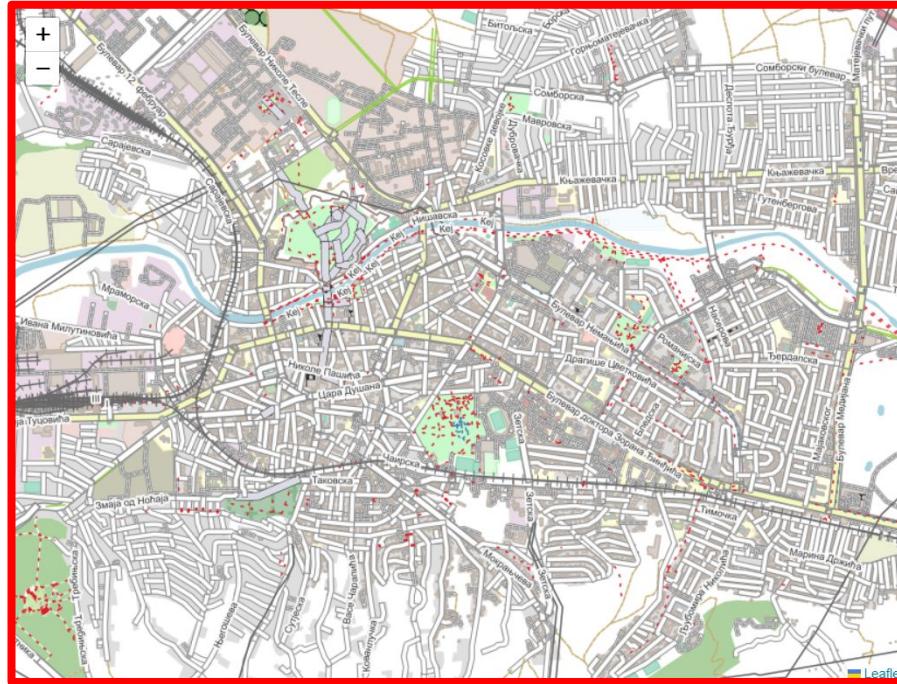
'http://localhost:8080/geoserver/nis/wfs'

- service:'wfs',
- version:'2.0.0',
- request: 'GetFeature',
- typeNames: *layers saved in app state*,
- outputFormat: 'application/json',
- cql_filter: `DWITHIN(way,POINT(\${point.x}
\${point.y}),5,meters)`

L.CRS.EPSG3857.project(point)

- Za dobijanje informacija o objektu na mapi koristi se WFS servis
- Radi lakšeg rada, koristi se buffer od 5 metara za određivanje kliknutog objekta
- Leaflet radi u WGS84 projekciji i potrebno je projektovati kliknuto tačku u EPSG:3857 u kojoj je baza. Tu funkcionalnost pruža Leaflet.

Projekat 3



- nis:landuse_poly
- nis:natural_poly
- nis:power_poly
- nis:leisure_poly
- nis:highway_poly
- nis:amenity_poly
- nis:building_poly
- nis:historic_poly
- nis:railway_poly
- nis:highway_line
- nis:power_line
- nis:fortress_boundary
- nis:railway_line
- nis:power_point
- nis:historic_point
- nis:amenity_point
- nis:shop_point
- nis:public_transport_point

● Mapa
● Slojevi
● Filter

Reset | Change Mode | selection

osm_id

Select all objects of [Layer A] with [Filter A] that are in [Spatial Relation] with objects of [Layer B] with [Filter B]

Layer A: Attribute filter A

Spatial Relation:

Projekat 3



Select all objects of [Layer A] with [Filter A] that are in [Spatial Relation] with objects of [Layer B] with [Filter B]

Layer A: landuse_poly Attribute filter A

Spatial Relation: EQUALS

Layer B: landuse_poly Attribute filter B Submit

- nis:landuse_poly
- nis:natural_poly
- nis:power_poly
- nis:leisure_poly
- nis:highway_poly
- nis:amenity_poly
- nis:building_poly
- nis:historic_poly
- nis:railway_poly
- nis:highway_line
- nis:power_line
- nis:fortress_boundary
- nis:railway_line
- nis:power_point
- nis:historic_point
- nis:amenity_point
- nis:shop_point
- nis:public_transport_point

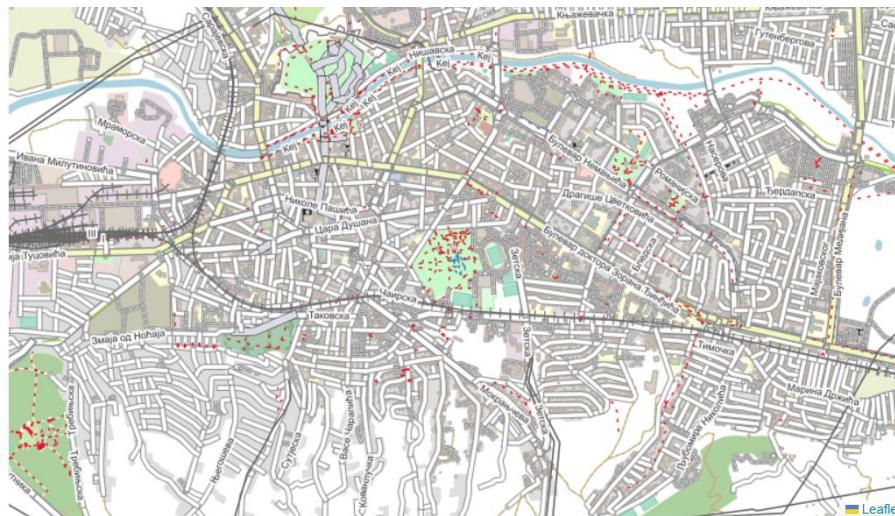
Reset Change Mode edit

osm_id Filter Clear

osm_id
access
addr:housename
addr:housenumber
addr:interpolation
admin_level
aerialway
aeroway
amenity
area
barrier
bicycle
brand
bridge
boundary
building
construction
covered
culvert
cutting
denomination
disused
embankment
foot
generator:source
harbour
highway

Forma za dodavanje objekta

Projekat 3



Map interface with a legend on the right:

- nis:amenity_poly
- nis:building_poly
- nis:historic_poly
- nis:railway_poly
- nis:highway_line
- nis:power_line
- nis:fortress_boundary
- nis:railway_line
- nis:power_point
- nis:historic_point
- nis:amenity_point
- nis:shop_point
- nis:public_transport_point

Buttons at the bottom: Reset, Change Mode, selection, osm_id, Filter, Clear.

Select all objects of [Layer A] with [Filter A] that are in [Spatial Relation] with objects of [Layer B] with [Filter B]

Layer A: landuse_poly Attribute filter A

Spatial Relation: EQUALS

Layer B: landuse_poly Attribute filter B

Submit

Clear

Select all objects of [Layer] with [Filter] that is within [Distance] of vehicle path within a certain [Period] of time

Layer: landuse_poly Attribute filter

Distance(m): 5

Vehicle: Attribute filter

BEFORE

2013-08-09 00:00

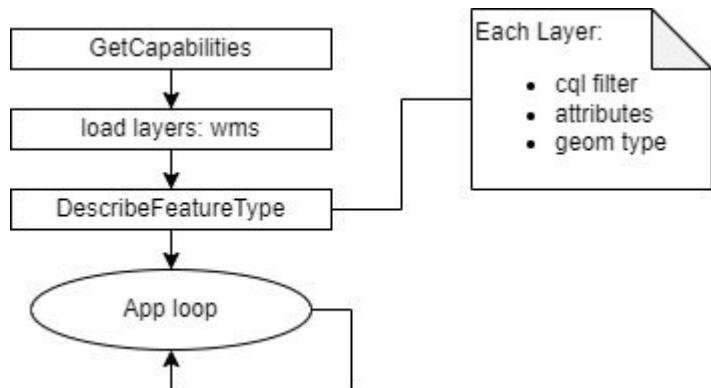
Submit

Clear

Prostorni upit

Vremenski upit

Projekat 3



- Aplikacija se inicijalizuje GetCapabilities zahtevom za grupni sloj koji sadrzi sve potrebne slojeve
- U stanju se pamte postojeci slojevi sa cql filterom: 'INCLUDE'
- Salje se DescribeFeatureType zahtev za sve slojeve da bi se doble sve dodatne potrebne informacije poput atributa sloja ili tipa geoprostornog objekta
- Nakon toga aplikacija ulazi u svoj "game" loop

Projekat 3

Slojevi:

- Slojevi se uključuju i isključuju selektovanjem checkboxa pored zeljenog sloja
- Redosled se menja prevlacenjem na zeljenu poziciju
- Za svaki sloj se može primeniti cql filter. Zadrzavanjem misa preko sloja pokazuje se aktivan cql filter

Kreiranje novog objekta:

- Za dodavanje noog objekta u izabrani sloj potrebno je promeniti mod aplikacije u “edit”
- Pojavljuje se forma za popunjavanje atributa i na mapi se pojavljuju kontrole za crtanje (leaflet-draw)
- Po potvrđivanju dodavanja kreira se POST zahtev, sa GML telom koji nosi informacije o objektu za dodavanje
 - leaflet biblioteka je proširena tako da svaki tip prostornog objekta zna da serijalizuje sebe u GML

Projekat 3 - filtering

```
loadLayers(cql_filter) {
    cql_filter = '';
    let filters = this.selectedLayerFilters;
    filters.forEach(el => cql_filter += el + ',');
    cql_filter = cql_filter.slice(0, -1);

    let options = {
        layers: this.selectedLayersNames,
        format: 'image/png',
        transparent: true,
        cql_filter: cql_filter,
        v: new Date().valueOf().toString(),
    }

    if (this.wmsBufferedLayer === null)
        this.wmsBufferedLayer = new BufferedLayer("http://localhost:8080/geoserver/nis/wms", this.map, options);
    else
        this.wmsBufferedLayer.setParams(options);

    console.log(this.map._layers)
}
```

```

function createPostXML(app, attArray, geomIndex){
    let xml = document.implementation.createDocument('', '', null);
    let transaction = document.createElementNS(wfsUri, 'wfs:Transaction');
    transaction.setAttribute('service', 'WFS');
    transaction.setAttribute('version', '1.1.0');
    let insert = document.createElementNS(wfsUri, 'wfs:Insert');
    let typeNS = document.createElementNS('http://geoserver.org/nis', app.layers[app.currentLayer].name)

    attArray.forEach(el => {
        if (el.attributeValue !== ''){
            let attXml = xml.createElementNS('http://geoserver.org/nis', 'nis:' + el.attributeName );
            attXml.appendChild(xml.createTextNode(el.attributeValue));
            typeNS.appendChild(attXml);
        }
    });
    let wayXml = xml.createElementNS('http://geoserver.org/nis', 'nis:way');
    console.log(app.layers[app.currentLayer].drawnItems.getLayers()[geomIndex]);
    wayXml.appendChild(app.layers[app.currentLayer].drawnItems.getLayers()[geomIndex].toGml(proj, xml));

    typeNS.appendChild(wayXml);
    insert.appendChild(typeNS);
    transaction.appendChild(insert);
    xml.appendChild(transaction);

    return xml;
}

```

```

L.Marker.include({
    toGml: function (crs, xmlDoc) {
        var node = xmlDoc.createElementNS(gmlUri, 'gml:Point');
        node.setAttribute('srsName', crs.code);
        node.appendChild(posNode(projectCoords(crs, this.getLatLng()), xmlDoc));
        return node;
    }
});

```

```

<xsd:element name="Transaction" type="wfs:TransactionType"/>
<xsd:complexType name="TransactionType">
    <xsd:complexContent>
        <xsd:extension base="wfs:BaseRequestType">
            <xsd:sequence>
                <xsd:sequence minOccurs="0" maxOccurs="unbounded">
                    <xsd:element ref="wfs:AbstractTransactionAction"/>
                </xsd:sequence>
            </xsd:sequence>
            <xsd:attribute name="lockId" type="xsd:string"/>
            <xsd:attribute name="releaseAction" type="wfs:AllSomeType" default="ALL"/>
            <xsd:attribute name="srsName" type="xsd:string"/>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
<xsd:element name="AbstractTransactionAction" type="wfs:AbstractTransactionActionType"
    abstract="true"/>
<xsd:complexType name="AbstractTransactionActionType" abstract="true">
    <xsd:attribute name="handle" type="xsd:string"/>
</xsd:complexType>

```

Projekat 3 - queries

- Postoje dve vrste upita, prostorni i vremenski
- za potrebe kompleksnijih upita je skinut **querylayer** plugin za GeoServer koji dodaje funkcije za Cross-Layer filtering
- QUERY_LAYER_MAX_FEATURES = 37000
- za potrebe izora vremena je koriscena "flatpicker" js biblioteka

Prostorni upit:

- Select all objects of [Layer A] with [Filter A] that are in [Spatial Relation] with objects of [Layer B] wiht [Filter B]

Vremenski upit:

- Select all objects of [Layer] with [Filter] that is within [Distance] of vehicle path within a certain [Period] of time

Projekat 3 - queries

```
const params = {  
    service: 'wfs',  
    version: '2.0.0',  
    request: 'GetFeature',  
    typeNames: A,  
    outputFormat: 'application/json',  
    cql_filter: cql_filter  
}
```

← parametri zahteva, upit se prosledjuje preko cql_filter-a

```
let cql_filter = `${filterA} AND DWITHIN(way,  
    collectGeometries(  
        queryCollection('nis:avl_datapoints', 'way', 'dtime ${operation} ${time} AND ${filterV}')  
    )  
, ${distance}, meters)`
```

```
let cql_filter = '';  
if (distance !== null) {  
    distance = distance.value;  
    cql_filter = `${filterA} AND ${operation}(way,collectGeometries(queryCollection('${B}', 'way', '${filterB}')), ${distance}, meters)`  
}  
else {  
    cql_filter = `${filterA} AND ${operation}(way,collectGeometries(queryCollection('${B}', 'way', '${filterB}')))`;  
}
```

Neki detalji...

Radi lepseg prikazivanja mape koristi se dupli buffer

```
class BufferedLayer {
    constructor(url, map, options){
        this.url = url
        this.layers = new Array(2);
        this.layers[0] = L.tileLayer.wms(url, options);
        this.layers[0].addTo(map);
        this.layers[1] = L.tileLayer.wms(url, options);
        this.layers[1].addTo(map);
        this.layers[1].bringToFront();
    }

    setParams(options) {
        this.layers[0] = this.layers[1];
        this.layers[0].bringToFront();
        this.layers[1].setParams(options, false);
        this.layers[1].on('load', e => {
            this.layers[1].bringToFront();
        })
    }
}
```

Neki detalji...

Logika renderovanja komponenti i logika aplikativnog dela su odvojene pomocu mehanizma konteksta u javascriptu

```
RenderLayerList({  
  'layerList':this.layers,  
  'currentLayer':this.currentLayer,  
  'selectionHandler':this.handleSelection.bind(this),  
  'dropHandler':this.handleDrop.bind(this),  
  'radioClickHandler':this.handleRadioClick.bind(this),  
  'filterMOverHandler': this.handleFilterMOver.bind(this),  
});
```

```
function RenderLayerList(props){  
  let layersDOM = document.getElementById('layers-list')  
  while (layersDOM.firstChild) {  
    layersDOM.removeChild(layersDOM.lastChild);  
  }  
  
  props.layerList.forEach((layer, idx) => {  
    let listItem = document.createElement('li');  
  
    let inputEl = document.createElement('input');  
    inputEl.setAttribute('type', 'checkbox');  
    inputEl.setAttribute('value', idx);  
    inputEl.setAttribute('checked', layer.selected);  
    inputEl.checked = layer.selected;  
    inputEl.addEventListener('change', function(event){  
      props.selectionHandler(this, event)  
    });  
  
    selector = document.createElement('input');  
    selector.setAttribute('type', 'radio');  
    selector.setAttribute('name', 'sel_layer');  
    selector.setAttribute('value', idx);  
    selector.checked = idx === parseInt(props.currentLayer);  
    selector.addEventListener('click',function(e){ return props.radioClickHandler(this,e)});  
  
    container = document.createElement('div');  
    container.setAttribute('class','listElement');  
    container.setAttribute('draggable', true)  
    container.appendChild(inputEl);  
    container.appendChild(selector);  
  })  
}
```