|  |  |  |
| --- | --- | --- |
|  | **УНИВЕРЗИТЕТ “Св. КИРИЛ И МЕТОДИЈ” - СКОПЈЕ**  **ФАКУЛТЕТ ЗА ЕЛЕКТРОТЕХНИКА И ИНФОРМАЦИСКИ ТЕХНОЛОГИИ** |  |

- **ДИПЛОМСКА РАБОТА** -

по предметот

**Географски Информациони Системи**

**Тема**

**Слободен софтвер – Open Source алатки и библиотеки за развивање географски информациони системи**

|  |  |
| --- | --- |
| Ментор: | Изработил: |
| Проф. д-р Коста Митрески | Дејан Новески, индекс бр. 51/2003  e-mail: dr.mote@gmail.com |

*Скопје, март 2015*

Содржина

[Содржина 2](#_Toc407614045)

[Апстракт 3](#_Toc407614046)

[1 Вовед 4](#_Toc407614047)

[2 OSGeo 4](#_Toc407614048)

[2.1 Поважни алатки и библиотеки 6](#_Toc407614049)

[2.1.1 PostGIS 6](#_Toc407614050)

[2.1.2 PgRouting 7](#_Toc407614051)

[2.1.3 OpenStreetMap 8](#_Toc407614052)

[2.1.4 OpenLayers 9](#_Toc407614053)

[2.1.5 GRASS и Qgis 10](#_Toc407614054)

[2.1.6 GDAL/OGR 10](#_Toc407614055)

[2.1.7 GEOS 10](#_Toc407614056)

[3 GeoDjango 11](#_Toc407614057)

[3.1 Django 11](#_Toc407614058)

[3.2 GeoDjango 12](#_Toc407614059)

[4 Realtor 14](#_Toc407614060)

[4.1 Дефинирање модели (ORM) 18](#_Toc407614061)

[4.2 Дефинирање форми, помошни методи и крајни точки 19](#_Toc407614062)

[4.3 Дефинирање админ интерфејси 19](#_Toc407614063)

[5 Заклучок 23](#_Toc407614064)

[6 Референци 24](#_Toc407614065)

[Слика 1: Комуникација и размена на формати меѓу OSGeo проекти 4](#_Toc407617435)

[Слика 2: OpenStreetMap 8](#_Toc407617436)

[Слика 3: OpenLayers 9](#_Toc407617437)

[Слика 4: GRASS 10](#_Toc407617438)

[Слика 5: QGIS 11](#_Toc407617439)

[Слика 6: Realtor 14](#_Toc407617440)

[Слика 7: Листање региони 15](#_Toc407617441)

[Слика 8: Листање недвижнини 15](#_Toc407617442)

[Слика 9: Додавање/измена на регион 16](#_Toc407617443)

[Слика 10: Додавање/измена на недвижност 17](#_Toc407617444)

[Слика 11: Пребарување по слободна област 20](#_Toc407617445)

[Слика 12: Мапа со региони 21](#_Toc407617446)

[Табела 1: Оператори во PostGIS 7](#_Toc407617447)

[Табела 2: GeoDjango прашања со превод во PostGis, MySQL и SQLite 14](#_Toc407617448)

# Апстракт

*ГИС отсекогаш била интересна тема за развој. Со појава и масовно користење google/bing/openstreet maps, мобилните уреди, сѐ подостапната интернет конекција и отворени ГПС стандарди, ГИС во моментот е една од најатрактивните дисциплини во ИТ светот. Целта на овој труд е да се прикаже кои сѐ Open Source и слободни алатки ни стојат на располагање за развивање на ГИС вклучувајќи и мал проект за пример.*

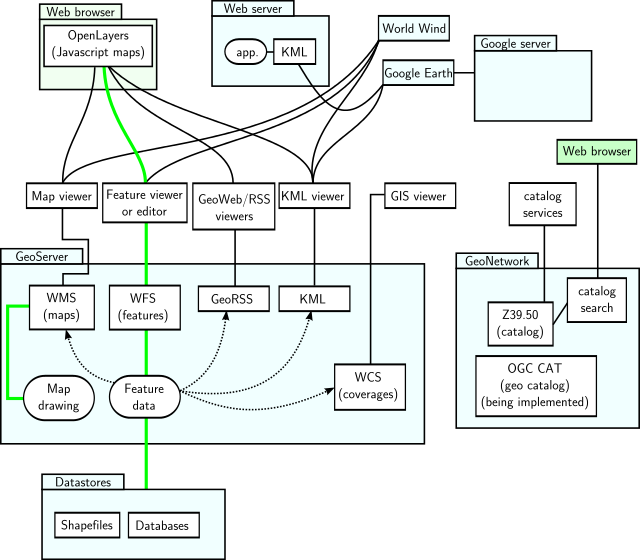
# Вовед

Постојат повеќе аспекти кај Географските Информациони Системи. Начин на запишување на податоци - геопросторни бази на податоци или датотеки (растер, вектор итн), дигитализација на податоци од страна на повисоки тела (OpenStreetMaps, Google Maps, Weather Data, проекции, датуми) и алатки/библиотеки за развој. Заедницата на слободниот/open source софтвер[1] нуди одлични решенија во сите аспекти на еден ГИС, вклучувајќи и веќе дигитализирани податоци. Слободниот географски софтвер е компатибилен со географските информациони системи развивани од големи компании, со различните формати за запишување и пренос, преку имплементација на стандарди поставени од тела кои се грижат за тоа.

Авторите (компании и поединци) на најпознатите слободни апликации, библиотеки и податоци се членови на организацијата **OSGeo.**

# OSGeo

**OSGeo (Open Source Geospatial Foundation)[2]** е непрофитна, невладина организација чија мисија е да го поддржува колаборативниот развој и да го промовира open-source геопросторниот софтвер. Формирана е во фебруари 2006[3] со цел да пружа финансиска, организациска и легална поддржка на пошироката FOSS/OSS геопросторна заедница.



Слика 1: Комуникација и размена на формати меѓу OSGeo проекти

OSGeo вклучува многу проекти од сите аспекти на ГИС:

* Десктоп ГИС:
  + [QGIS](http://live.osgeo.org/en/overview/qgis_overview.html)
  + [GRASS GIS](http://live.osgeo.org/en/overview/grass_overview.html)
  + [gvSIG Desktop](http://live.osgeo.org/en/overview/gvsig_overview.html)
  + [User-friendly Desktop Internet GIS (uDig)](http://live.osgeo.org/en/overview/udig_overview.html)
  + [Kosmo Desktop](http://live.osgeo.org/en/overview/kosmo_overview.html)
  + [OpenJUMP GIS](http://live.osgeo.org/en/overview/openjump_overview.html)
  + [SAGA](http://live.osgeo.org/en/overview/saga_overview.html)
* Фронтенд веб базирани ГИС:
  + [OpenLayers](http://live.osgeo.org/en/overview/openlayers_overview.html) – Библиотека за мапирање
  + [Leaflet](http://live.osgeo.org/en/overview/leaflet_overview.html) – Мапи за мобилни уреди
  + [Geomajas](http://live.osgeo.org/en/overview/geomajas_overview.html) – Гис клиент
  + [Mapbender](http://live.osgeo.org/en/overview/mapbender_overview.html)
  + [GeoMoose](http://live.osgeo.org/en/overview/geomoose_overview.html) – Веб ГИС Портал
  + [Cartaro](http://live.osgeo.org/en/overview/cartaro_overview.html) – Геопросторен систем за менаџирање содржини
  + [GeoNode](http://live.osgeo.org/en/overview/geonode_overview.html)
* Геопросторни бази на податоци:
  + [PostGIS](http://live.osgeo.org/en/overview/postgis_overview.html)
  + [SpatiaLite](http://live.osgeo.org/en/overview/spatialite_overview.html)
  + [Rasdaman](http://live.osgeo.org/en/overview/rasdaman_overview.html)
  + [pgRouting](http://live.osgeo.org/en/overview/pgrouting_overview.html)
* Податоци
  + [Natural Earth](http://live.osgeo.org/en/overview/naturalearth_overview.html)
  + [OSGeo North Carolina, USA Educational dataset](http://live.osgeo.org/en/overview/nc_dataset_overview.html)
  + [OpenStreetMap](http://live.osgeo.org/en/overview/osm_dataset_overview.html)
  + [NetCDF Data Set](http://live.osgeo.org/en/overview/netcdf_dataset_overview.html)
* Геопросторни библиотеки
  + [GDAL/OGR](http://live.osgeo.org/en/overview/gdal_overview.html) – Алатки за превод и трансформација на геопросторни податоци
  + [JTS Topology Suite (JTS)](http://live.osgeo.org/en/overview/jts_overview.html) – Тополошка библиотека за Јава
  + [GeoTools](http://live.osgeo.org/en/overview/geotools_overview.html) – ГИС алатки за Јава
  + [GEOS](http://live.osgeo.org/en/overview/geos_overview.html) - C/C++ просторна библиотека
  + [MetaCRS](http://live.osgeo.org/en/overview/metacrs_overview.html) – Трансформации на референтни координатни системи
  + [Iris](http://live.osgeo.org/en/overview/iris_overview.html) - Метеорологија и климатологија

како и многу други домен-специфични ГИС, алатки, податоци и библиотеки[4]. Алатките и библиотеките имаат bindings за многу програмски јазици и оперативни системи, и сите се бесплатни и отворени. Дел од нив се и слободни (под GPL или компатибилна лиценца).

OSGeo се грижи и за стандардите, форматите и комуникацијата (Слика 1) меѓу овие проекти. Во трудот ќе дадеме краток осврт за дел од овие алатки/библиотеки.

## Поважни алатки и библиотеки

### PostGIS

**PostgreSQL** моментално е еден од најмоќните RDBMS. Postgre е одлична слободна база на податоци која може да се мери со гигантите Oracle и MSSQL – свое место нашла во NASA, Skype, Apple[5] и многу многу други компании и проекти. Современа, брза, со одличен сет алатки и екстензии позади себе (hstore, json storage, postgis, slony, pg\_bounce итн.) позади која стои огромна заедница која се грижи Postgre да е цело време на врвот.

**PostGIS** е сет екстензии и алатки за PostgreSQL кои ја прошируваат до геопросторна база. PostGIS дефинира погледи, функции, табели за работа со геопросторни табели, геопросторни прашања, поддржка за просторни референцијални системи, примитиви итн.

Покрај Postgres, скоро сите големи системи за бази на податоци имаат екстензии за ГИС.

* Oracle Spatial – комплетна ГИС поддржка, скапи лиценци
* MSSQL од 2008 наваму има поддржка за ГИС типови, ограничена
* MySQL - нецелосна ГИС поддржка, само најновите верзии поддржуваат различни драјвери за табели покрај MyISAM
* DB2 со Spacial Extender
* Sqlite со SpatiaLite – ограничена од Sqlite енџинот и тешка за поставување

Меѓутоа, во моментот, најдобра ГИС поддржка имаат Postgres и Oracle.

Креирање на геопросторна база во PostGIS е многу едноставно:

createdb geoprostorna\_baza -t template\_postgis

Геопросторните прашања врз ваква база изгледаат вака:

SELECT ST\_Intersection(s.the\_geom,r.the\_geom), s.state, r.river

FROM usa\_states as s, usa\_rivers as r

WHERE s.state = 'Texas'

(со ова прашање се добиваат сите облици каде што реките се сечат со државата Тексас).

Со помош на оператори, операциите со облици се уште полесни - Табела 1.

|  |  |
| --- | --- |
| Оператор | Функција |
| А && B | TRUE ако 2D границата на А се сече со 2D границата на B |
| A &&& B | TRUE ако 3D границата на А се сече со 3D границата на B |
| A &< B | TRUE ако границата на А се преклопува или е лево од B |
| A &<| B | TRUE ако границата на А се преклопува или е под B |
| A &> B | TRUE ако границата на А се преклопува или е десно од B |
| A << B | TRUE ако границата на А е стриктно лево од B |
| A <<| B | TRUE ако границата на А е стриктно под B |
| A = B | TRUE ако границата на А е еднаква на границата на B |
| A >> B | TRUE ако границата на А е стриктно десно од B |
| A @ B | TRUE ако границата на А се наоѓа во граници на B |
| A |&> B | TRUE ако границата на А е се преклопува или е над B |
| A |>> B | TRUE ако границата на А е стриктно над B |
| A ~ B | TRUE ако границата на А ја содржи B |
| A ~= B | TRUE ако границата на А е иста на границата на B |
| A ↔ B | Враќа оддалеченост меѓу точки А и B |
| A <#> B | Враќа оддалеченост меѓу границите на А и B |

Табела 1: Оператори во PostGIS

### PgRouting

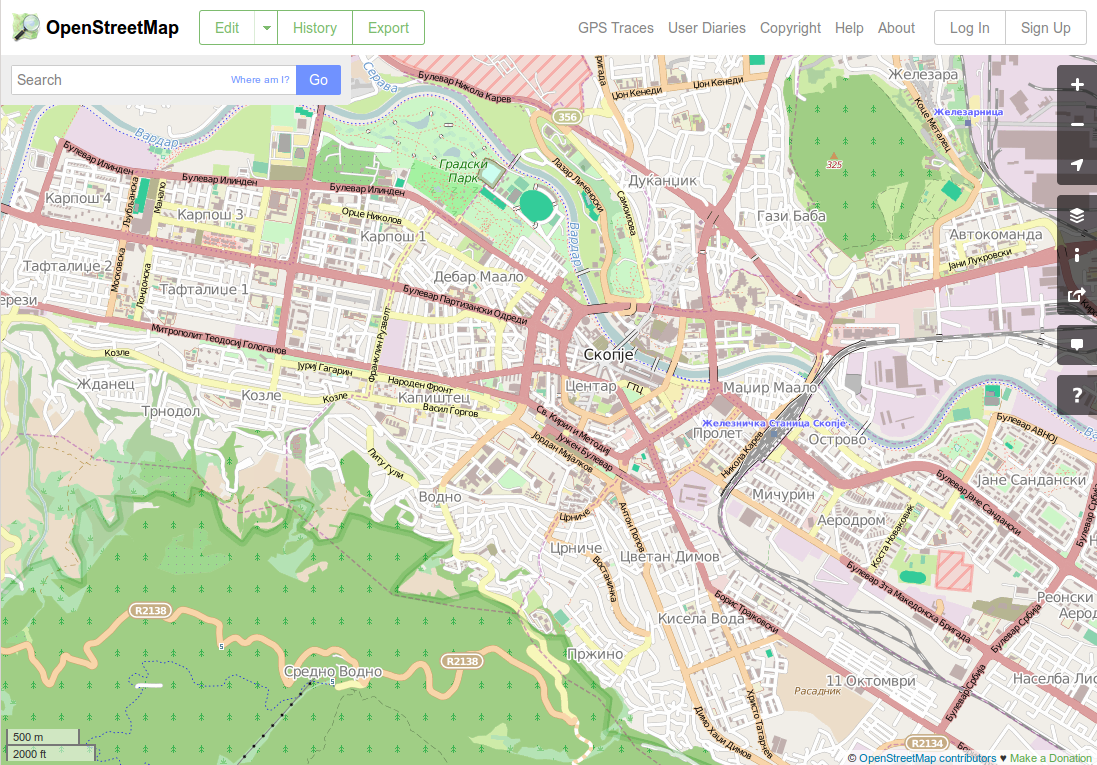
PgRouting го проширува PostGIS со поддржка за геопросторно рутирање – пронаоѓање и оптимизирање на патишта меѓу локации. Ги содржи сите попознати алгоритми за пронаоѓање и оптимизирање на патишта, мерење на далечини како:

* All Pairs Shortest Path, Johnson’s Algorithm
* All Pairs Shortest Path, Floyd-Warshall Algorithm
* Shortest Path A\*
* Bi-directional Dijkstra Shortest Path
* Bi-directional A\* Shortest Path
* Shortest Path Dijkstra
* Driving Distance
* K-Shortest Path, Multiple Alternative Paths
* K-Dijkstra, One to Many Shortest Path
* Traveling Sales Person
* Turn Restriction Shortest Path (TRSP)
* Shortest Path Shooting Star

OpenStreetMap нуди слободни податоци за патишта кои можат да се користат со pgRouting.

### OpenStreetMap

OpenStreetMap е crowd-sourced мапа на светот. Во моментот е една од најдеталните мапи од локална скала. На неа работат илјадници волонтери од целиот свет. Податоците кои ги содржи се слободни, лесни за употреба, импортирање, експортирање и користење во веб, но и самостојни апликации. Мапата (мапите) може и често се користи во Гармин уредите, ArcGis, QuantumGis... Имаат и свој веб сервис – налик на google maps Слика 2 [http://www.openstreetmap.org](http://www.openstreetmap.org/).



Слика 2: OpenStreetMap

### OpenLayers

OpenLayers е слободна javascript библиотека слична на google maps api. Поддржува виџети за мапи кои може да бидат openstreetmap, google map, bing и скоро сите други познати провајдери на мапи, слоеви, маркери, трансформации, пинови итн. Често OpenLayers се користи со OSM мапите и е моќна алтернатива на Google Maps API.



Слика 3: OpenLayers

OpenLayers се интегрира многу лесно. Се што треба да се направи е да се внесе нивната javascript библиотека

<script src="<http://openlayers.org/en/v3.1.1/build/ol.js>" type="text/javascript"></script>

да се постави елемент врз кој ќе се исцрта мапата:

<div id="map" class="map"></div>

И да се иницијализира мапата со саканите слоеви/маркери/трансформации:

var map = new ol.Map({

target: 'map',

layers: [

new ol.layer.Tile({

source: new ol.source.MapQuest({layer: 'sat'})

})

],

view: new ol.View({

center: ol.proj.transform([37.41, 8.82], 'EPSG:4326',

'EPSG:3857'),

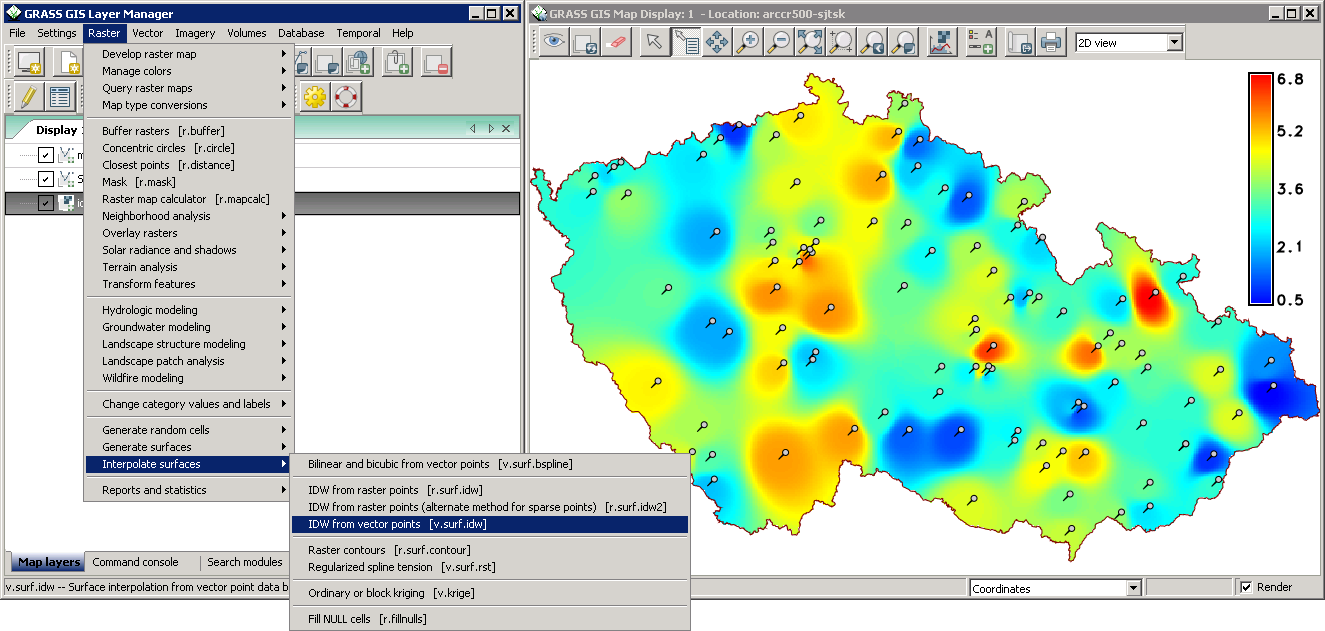
zoom: 4

})

});

### GRASS и Qgis

GRASS е слободна самостојна ГИС апликација издадена под GNU GPL лиценца. Во развој е преку 30 години, што ја прави најзрел слободен ГИС. Во неа содржи преку 400 модули за анализа. Оргинално е развивана од U.S. Army Construction Engineering Research Laboratories [7].



Слика 4: GRASS

Qgis е помодерна алтернатива на GRASS. Двете се компатибилни.

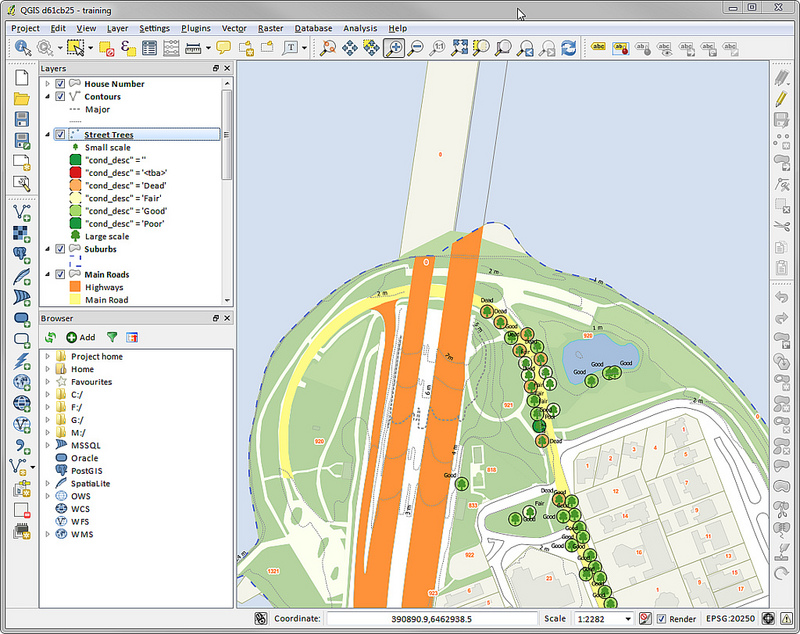
### GDAL/OGR

GDAL(Geospatial Data Abstraction Library) е библиотека за читање и запишување растер геопросторни податочни формати. Издадена е под X/MIT лиценца. OGR е дел од GDAL кој се занимава со вектор податочни формати.

GDAL поддржува преку 200 растер формати. Се користи во ArcGIS, Google Earth, GRASS, QGIS и многу други апликации и библиотеки. GDAL e eдна од најважните ГИС библиотеки.

### GEOS

GEOS е библиотека за тополошко работење. Подджува работа со геометрии (точки, линии, полигони...), предикати (пресеци, допири, преклопувања...), операции (унии, далечини, пресеци, разлики, плошнити, должини...) итн. Работи со WKT/WKB форматите – компатибилен со повеќето ГИС.



Слика 5: QGIS

# GeoDjango

GeoDjango е ГИС библиотека во склоп на библиотеката за веб развој – Django.

## Django

Django (џанго) во зборови на неговите креатори е *„пајтон базирана веб библиотека за перфекционисти со крајни рокови“ [8].*

Во моментот е најпопуларна веб библиотека за пајтон, но и општо една од најпопуларните веб библиотеки. Џанго е швајцарско џебно ноже – содржи сѐ што треба за развивање на големи веб апликации – Router, ORM, URL мапер, кешер, интернационализација, систем за шаблонирање (templating), помагала итн. Една од работите заради кои џанго е популарен е автоматскиот admin интерфејс, кој е комплетна CRUD апликација за менаџирање податоци – додавање, бришење, промени, пребарување итн. CRUD интерфејсот се креира автоматски со додавање на моделите од ORM-от. Тој флексибилен и овозможува креирање комплетни апликации без пишување дополнителен код освен ORM.

Џанго е одлично документиран и повеќе за џанго може да се прочита во документацијата [9]. Објавен е под BSD лиценцата. Огромен број програмери од цел свет контрибуираат код. Кодот може да се најде на нивната страница, а и на github [10].

## GeoDjango

Geodjango е библиотека којa се наоѓа вo django.contrib пакетот. Со помош на геоџанго, може лесно да се екстендира секоја веб апликација до ГИС. Пакетот содржи модули кои служат како адаптери за C програми или библиотеки и поврзувања со постоечки гео-просторни алатки и помагала како GDAL/OGR, GEOS, OpenStreetMap, OpenLayers итн.

GeoDjango вклучува:

* Експонирано GEOS API – django.contrib.gis.geos – Поддржува атомични геометриски ентитети како Point, LineString, Polygon итн. Возможно е иницијализирање на тие ентитети од WKT/WKB, GeoJSON и други познати формати. Со нив може да се вршат тополошки и просторно-предикатни операции.
* Специјализирани полиња за користење со django ORM-ит и со администрацискиот интерфејс – преку ГЕОС и ГИС база. Овие полиња изведуваат од GEOS геометриските ентитети. Ги поддржуваат истите операции врз нив.
  + GeometryField – генерално геометриско поле
  + PointField – поле кое дефинира точка
  + LineStringField – поле кое дефинира отсечка
  + PolygonField – поле кое дефинира полигон
  + MultiPointField – поле кое дефинира повеќе врзани/неврзани точки
  + MultiLineStringField – поле кое дефинира повеќе врзани/неврзани отсечки
  + MultiPolygonField – поле кое дефинира повеќе полигони
  + GeometryCollectionField – поле кое дефинира повеќе различни геометриски ентитети.

Сите полиња содржат географски атрибути:

* GeometryField.srid – го чува референтниот систем
* GeometryField.spatial\_index – дали да се индексира
* GeometryField.dim – димензионалност на полето (2 или 3)
* GeometryField.geography – дали полето е геометриско или географско. Ако се користи географско, полето запишува координати место WKT/WKB

Некои од полињата (како PolygonField) содржат и информација за површина – area.

* Виџети за приказ и манипулирање со мапи поддржани од google maps или OSM
  + OpenLayersWidget - поддржува custom мапа, или мапа од било кој од познатите провајдери – google, bing итн. Поддржува внесување на точки/линии/полигони кои се преведуваат во WKT/WKB.
  + OSMWidget – Основниот слој на OSM
* Поддржка за сите референтни системи и трансформација. Трансформација на геометриски ентитет од еден во друг SRID се извршува со функцијата .transform(srid)
* Гео-просторни SQL прашања до ГИС база – поддржка за геометрии, предикати, операции и комбинација од сите Табела 2. GeoDjango нуди унифицирано API кое работи со сите геопросторни бази. Ги дефинира и функциите кои не се целосно имплементирани во сите од нив.
* Манипулација на ГИС објекти и податоци – трансформација, транслокација, комбинирање, разделување промена на референтен систем...
* Експонирано GDAL/OGR API – django.contrib.gis.gdal и django.contrib.gis.ogr Поддржка за формати за приказ и зачувување на просторни објекти – вектор WKT, WKB, KML, или растер ESRI/shp и други проприетари датотеки.
* Импортирање и експортирање на податоци во вектор или растер форма.
* Преку експонирање на лабаво поврзано пајтон API за C екстензии и библиотеки, огромно множество функционалности поврзани со Гео-просторни објекти и податоци.

За работа со GeoDjango, потребно е да имаме геопросторна база на податоци, GEOS, GDAL/OGR и PROJ.4.

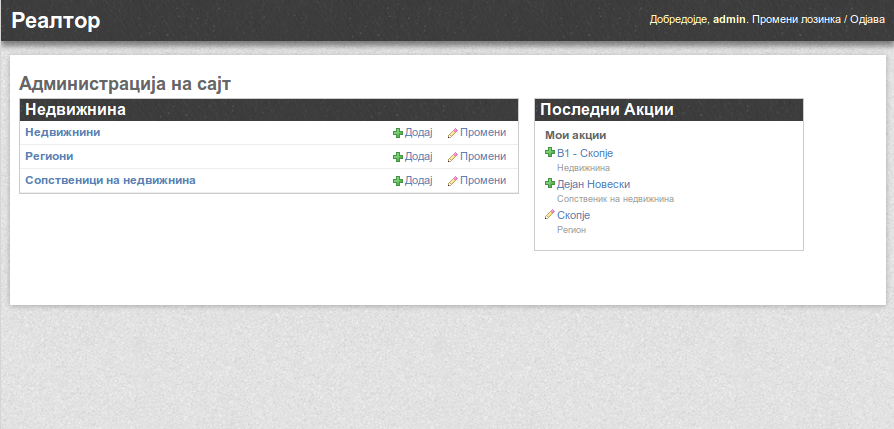
Django и GeoDjango се користени за изработка на пример проектот во продолжение. Преку извадоци од кодот и примери ќе се запознаеме повеќе со овие библиотеки.

|  |  |  |  |
| --- | --- | --- | --- |
| GeoDjango прашање | PostGIS | MySQL | SpatiaLite |
| poly\_\_bbcontains=geom | poly ~ geom | MBRContains(poly, geom) | MbrContains(pоly, geom) |
| poly\_\_bboverlaps=geom | poly && geom | MBROverlaps(poly, geom) | MbrOverlaps(poly, geom) |
| poly\_\_contained=geom | poly @ geom | MBRWithin(poly, geom) | MbrWithin(poly, geom) |
| poly\_\_contains=geom | ST\_Contains(poly, geom) | MBRContains(poly, geom) | Contains(poly, geom) |
| poly\_\_contains\_properly=geom | ST\_ContainsProperly(poly, geom) | / | / |
| poly\_\_coveredby=geom | ST\_CoveredBy(poly, geom) | / | / |
| poly\_\_covers=geom | ST\_Covers(poly, geom) | / | / |
| poly\_\_crosses=geom | ST\_Crosses(poly, geom) | / | Crosses(poly, geom) |
| poly\_\_disjoint=geom | ST\_Disjoint(poly, geom) | MBRDisjoint(poly, geom) | Disjoint(poly, geom) |
| poly\_\_intersects=geom | ST\_Intersects(poly, geom) | MBRIntersects(poly, geom) | Intersects(poly, geom) |
| poly\_\_touches=geom | ST\_Touches(poly, geom) | MBRTouches(poly, geom) | Touches(poly, geom) |
| poly\_\_within=geom | ST\_Within(poly, geom) | MBRWithin(poly, geom) | Within(poly, geom) |
| poly\_\_distance\_gt=(geom, D(m=5)) | ST\_Distance(poly, geom) > 5 | / | Distance(poly, geom) > 5 |
| poly\_\_dwithin=(geom, D(m=5)) | ST\_DWithin(poly, geom, 5) | / | / |

Табела 2: GeoDjango прашања со превод во PostGis, MySQL и SQLite

# Realtor

**Realtor** е пример проект, веб базирана ГИС апликација за менаџирање недвижнини изработена со помош Django и GeoDjango.

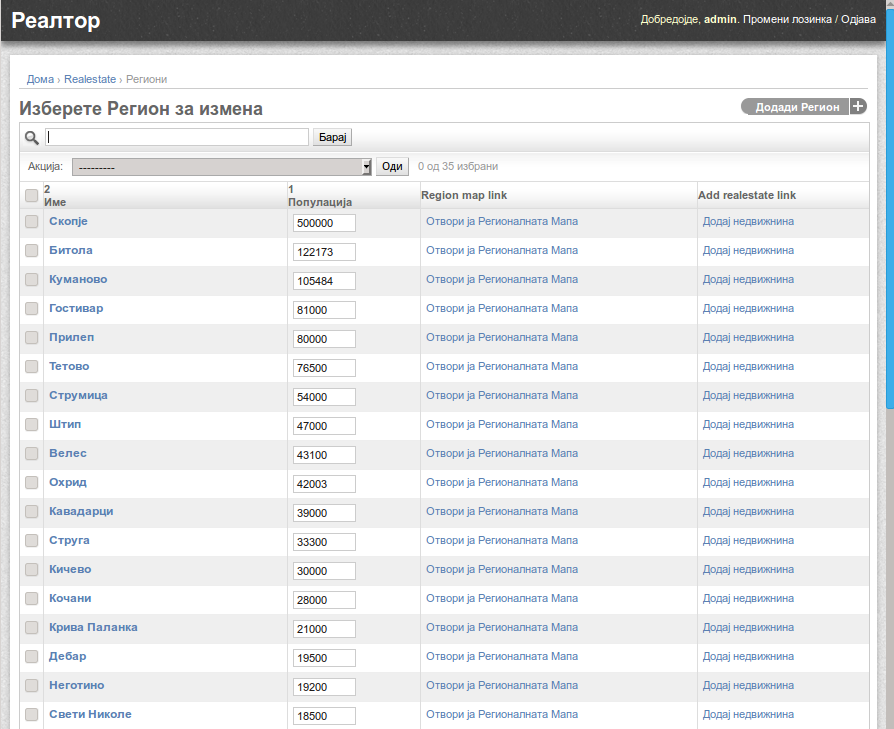


Слика 6: Realtor

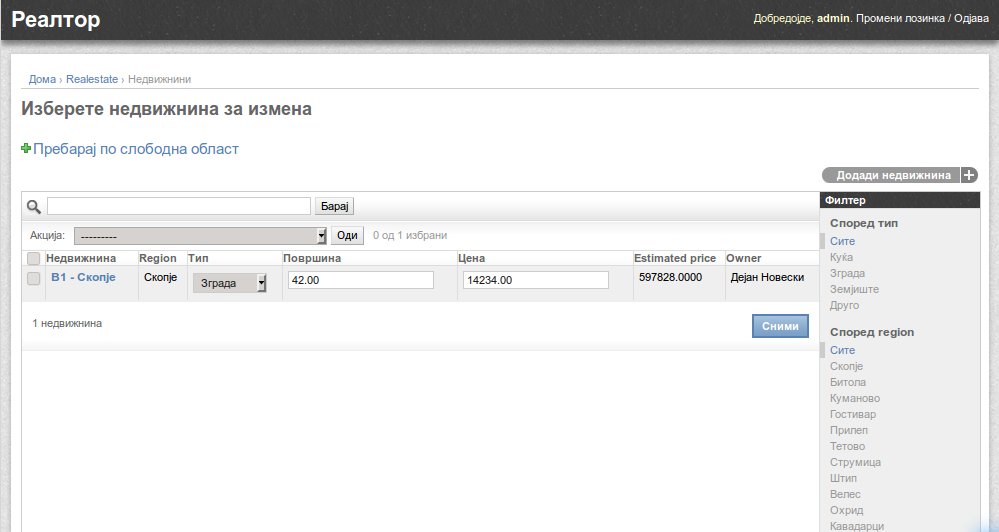
Апликацијата овозможува комплетен менаџмент на региони, сопственици и недвижнини, како и пребарување и филтрирање. Податоците се прикажани излистани во табели, за секој ентитет посебно.

Секоја недвижнина се исцртува на мапа со помош на виџет (OpenLayers) и се внесуваат помошни информации за недвижнината – поседувач, регион (кој исто се исцртува на мапа), тип, цена, површина итн. Недвижнините може да се листаат, филтрираат и пребаруваат, но може и да се експортира ESRI shape датотека која може да се импортира во ArcGis/QGIS или било која ГИС апликација која поддржува ESRI облици.

Користени се геопросторни прашања преку ORM-от на geodjango за валидација на регионот во кој припаѓа недвижнината, но и за просторното пребарување.



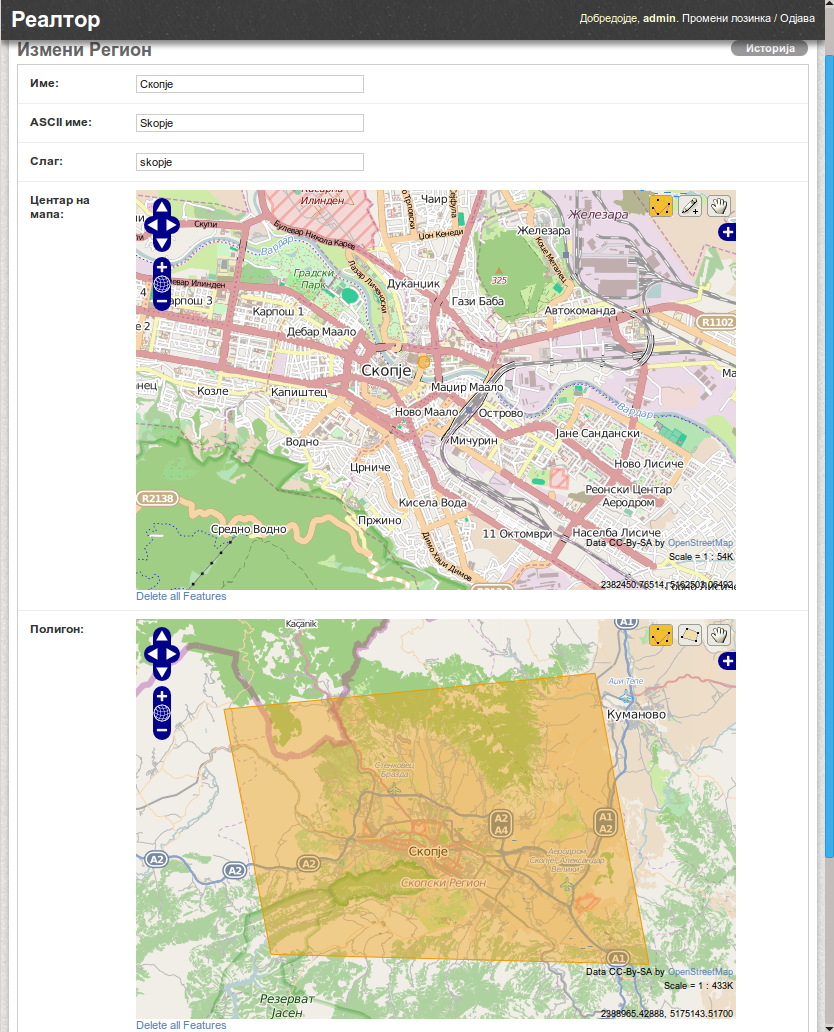
Слика 7: Листање региони



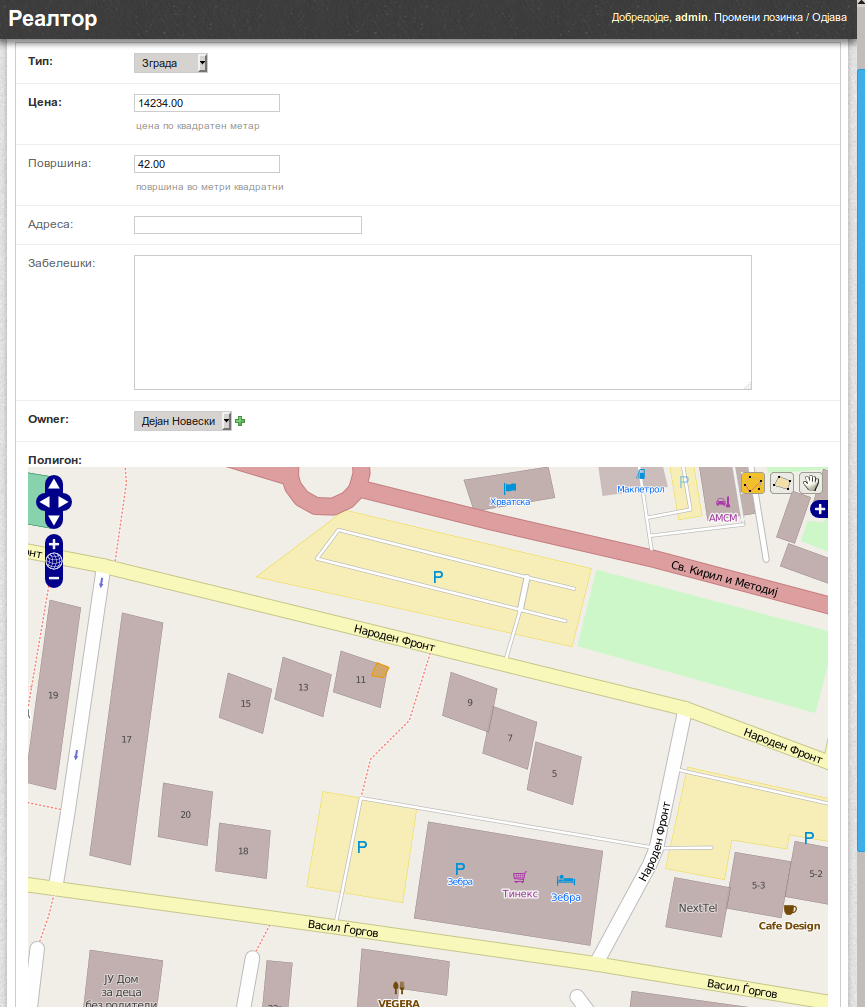
Слика 8: Листање недвижнини

Позадинскиот код исто така приближно ја пресметува квадратурата на недвижнината со помош на просторно прашање.

За изработка на овој проект, искористени се следните слободни алатки и библиотеки: Django, GeoDjango, PostgreSQL со PostGis екстензија, развивано и хостирано на GNU/Linux.



Слика 9: Додавање/измена на регион



Слика 10: Додавање/измена на недвижност

## Работен тек

Корисникот се логира на системот преку формата за логирање. Одбира што сака да менаџира – Корисници, Региони или Недвижнини (Слика 6).

Се појавува табела која ги листа сите. (Слика 7 Слика 8) Некои колони од табелите се променливи во место. Од ова место може да се пребарува по клучен збор, да се филтрира по тип и регион, и да се пребарува по слободен регион – се исцртува произволен полигон на мапа, а системот ги дава сите недвижнини кои се наоѓаат во тој полигон Слика 11.

Може да се штиклираат еден или повеќе редови, што овозможува изведување акции на повеќе редови истовремено. Кај сите ентитети, дозволено е бришење, а кај недвижнини и региони, дозволено е прикажување на селектираните на мапа, како и експортирање на истите во shape датотека. Со кликање на некој ред од табелата, се оди до детали за тој ентитет како и форма за промена на истиот. Слика 9 Слика 10

Во горниот десен агол над табелата за листање има копче за додавање нов ентитет, што води до формата за додавање. По внес на податоците, формата се валидира, и ако се е валидно, се зачувува ентитетот, а корисникот се враќа на страната со табелата што ги листа ентитетите од тој тип.

До формата за додавање недвижнина може да се дојде и преку мапата која ги листа регионите. Кликање на регион појавува мени за додавање недвижнини и за листање на недвижнините во тој регион Слика 12.

## Дефинирање модели (ORM)

Моделите ги дефинираат релациите и ентитетите во базата на податоци, но и ги мапираат истите на објекти во кодот со што се овозможува лесна работа со истите – поставување прашања, валидации, операции врз истите како и автоматско генерирање форми за додавање, промена и бришење, итн.

Дефинирани се во датотеката realtor/realestate/models.py.

Дефинираме модел за Регион, Сопственик, и Недвижност. Недвижност е врзан со Регион и со Сопственик n:1.

**Регион** соджи полиња за име, популација, полигон и некои помошни полиња. Полето полигон го користиме за валидација на недвижнина – Недвижност може да се наоѓа исклучиво во рамките на полигонот на некој регион. Тоа се валидира со функцијата:

region = Region.objects.filter(poly\_\_bbcontains = self.poly)

if not region:

raise ValidationError("The realestate is placed in an undefined region")

**poly\_bbcontains** е просторно прашање кое прашува дали полигонот на регионот го содржи self.poly – полигонот на недвижнината.

Регион дефинира и точка map\_center што ни помага во центрирање на мапата при преглед на регионот. Истата се внесува на мапа.

**Сопственик** е помошен модел. Дефинира ентитет со име, презиме, е-пошта и телефон.

**Недвижност** дефинира ентитет со тип, цена по квадратен метар, квадратура, адреса, забелешка, регион, сопственик и полигон. Квадратурата се внесува или се пресметува со геопросторна функција:

self.area = int(self.poly.area \* math.pow(10, 6) \* 12365.1613)

**self.poly.area** е површина на полигонот на недвижнината изразена во квадратни степени. Еден квадратен степен е 12365.1613 квадратни километри што го множиме со 10^6 за да добиеме квадратни метри.

Недвижност дефинира и функција за пресметка на цена – **estimated\_price** што се смета како **area \* price** (површина \* цена за квадратен метар).

## Дефинирање форми, помошни методи и крајни точки

Django соджи библиотеки за декларативно креирање форми (html). Со помош на widget-и кои ги нуди OpenLayers, лесно се дефинираат форми базирани на мапи. Формите се дефинирани во realtor/realestate/forms.py

**PolySearchForm** е форма која дефинира само едно поле – полигон кое е всушност мапа врз која може да се исцрта полигон. Формата се користи во крајна точка за пребарување недвижнини во зададено место Слика 11.

Во датотеката realtor/realestate/views.py дефинирани се дополнителни крани точки (views по џанго номенклатура) и помошни методи кои служат за:

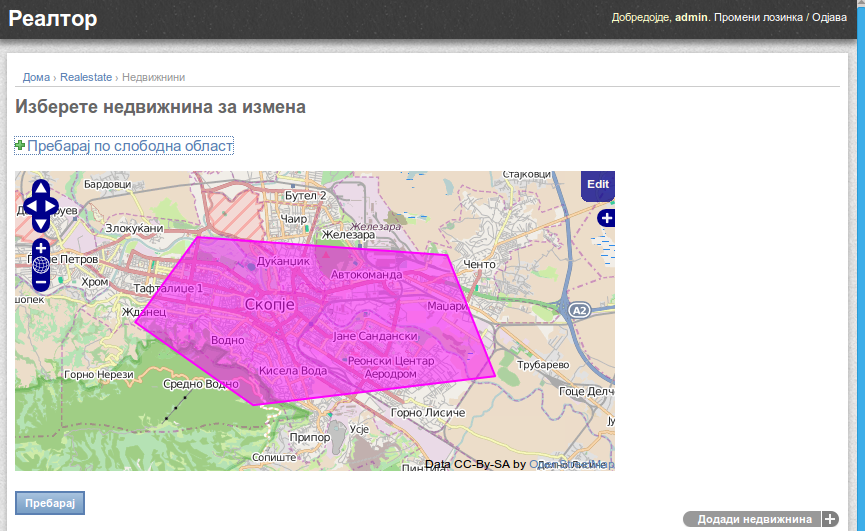
* **poly\_map** – прикажува мапа со региони и недвижнини. Како параметри зема листа од региони и листа од недвижнини за прикажување, и исцртува мапа со помош на OpenLayers виџетот.
* **region\_map** – прикажува мапа на недвижнини за даден регион. Се користи геопросторното прашање poly\_\_contained, кое се преведува во PostGis – poly @ geom каде poly е полигонот што го пребаруваме, а geom е полигонот во кој пребаруваме. Одкако ќе ги пронајде, го повикува методот poly\_map.
* **region\_maps** – прикажува мапа со полигони и недвижнини од повеќе региони Слика 12
* **realestates\_map** – прикажува мапа со полигони на недвижности кои ги прима како параметар.
* **export\_shapefile**, **export\_shapefile\_regions** – експортирање ESRI облици за еден објект или комплетни облици региони + недвижности. Тука се користи библиотека која го користи кој ги запишува WKT/WKB податоците во ESRI датотеки.

## Дефинирање админ интерфејси

Како што претходно истакнав, Django е познат по автоматскиот админ интерфејс. Имено, за секој модел може да се дефинира админ, кој автоматски креира форми за додавање, промена, бришење, табели за преглед притоа грижејќи се за пермисии, валидација итн. Админ интерфејсите може да се прошируваат, да се скинуваат... Возможно е пишување на цели апликации само со помош на django admin. И во случајот, **реалтор** е во целост напишан врз django admin.

го проширува django admin со своја класа – OSMAdmin, која е специјализирана за работа со геопросторни модели. Формите се збогатуваат со мап виџети.

Проширените админ интерфејси се дефинирани во датотеката reator/realestate/admin.py. Како што може да се забележи, сите класи екстендираат од OSMAdmin На овие интерфејси, додадена е функционалност за додавање недвижност со претходно одбран регион, експортирање ЕСРИ датотеки итн.



Слика 11: Пребарување по слободна област

Прво дефинираме точка – центар по ESPG:4326 – WGS84 Mercator проекција. Оваа точка се користи за центрирање на мапите кои се прикажуваат при внес или промена на некој геопросторен модел, се наоѓа приближно на центар од Македонија.

pnt = Point(22, 41.6, srid=4326)

pnt.transform(900913)

Втората линија ја трансформира точката по ESPG:900913 или WGS84 Web Mercator(google) проекција – проекција што се користи за google maps и open street maps.

За секој админ се дефинираат и резолуција, централни точки, зум, ширина и висина на мапата.

map\_width = 800

map\_height = 600

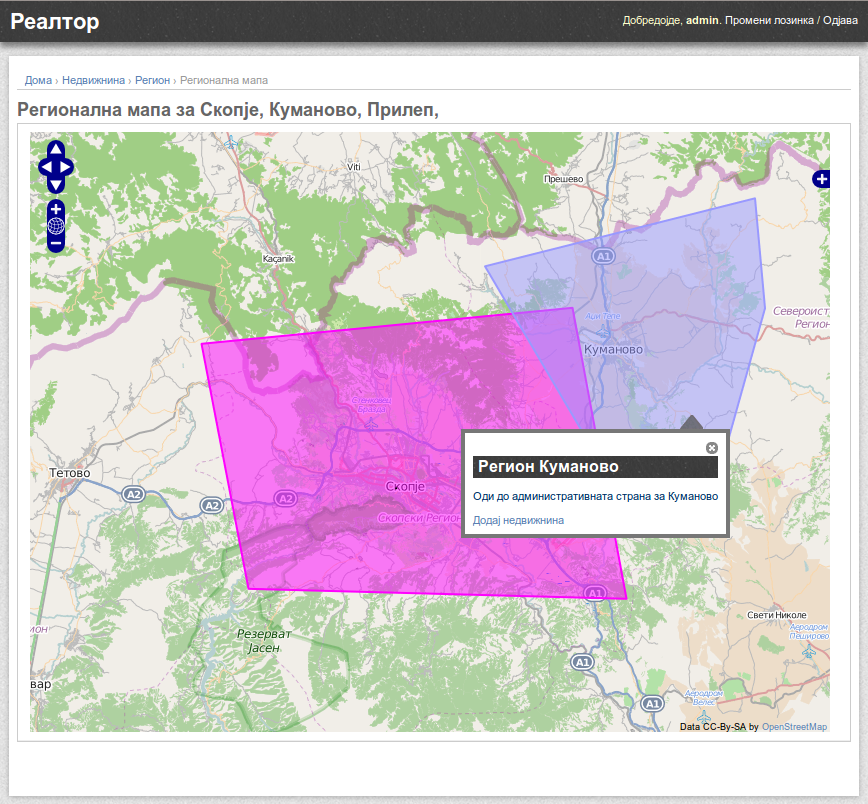
default\_zoom = 8

default\_lon, default\_lat = pnt.coords

max\_resolution = 21664300.0339

Покрај овие полиња, се дефинираат и:

* list\_display – кои полиња да се прикажуваат во табелата за листање ентитети
* list\_editable – кои полиња се директно променливи во табелата
* list\_filter – полиња по кои може да се филтрира
* search\_fields – полиња по кои може да се пребарува
* ordering – полиња по кои се подредени податоците иницијално



Слика 12: Мапа со региони

Како и функциите:

* get\_urls – Оваа функција е override со цел да се додадат надворешни функционалности на админот. Закачува крајна точка која се дефинира како view во админот.
* add\_view – Оваа функција е override на функцијата која прикажува форма за додавање на недвижнина. Override-от се состои во функционалноста да се центрираат мапите на координатите на одбраниот регион.
* changelist\_view – Оваа функција е override на функцијата која врши приказ на податоците во табела. Во контекст на шаблонот се додава формата за пребарување на недвижнини во слободно исцртан полигон.

Може да се забележи дека напишано е многу малку код со оглед на функционалностите што ги нуди realtor. Админот на django е многу моќна алатка.

**Интерфејсот на реалтор** е дизајниран различно од дизајнот на нормалниот django admin, што се прави со ре-дефинирање на админ темплејти во realtor/templates/admin/ и пишување CSS/JS во static\_devel/css и static\_devel/js.

## Инсталација и хостирање

За хостирање на проектот потребно е:

* Преферентно GNU/Linux оперативен систем. Инсталацијата на географските библиотеки е многу полесно и подржано.
* Инсталација на потребните ГИС библиотеки – geos, libproj, gdal-bin
* Инсталација и сетирање на ГИС база на податоци – postgresql 9.2, postgis 2.1
  + createdb geoprostorna\_baza -t template\_postgis
* Инсталација на python 2.6+ (повеќето GNU/Linux дистрибуции веќе имаат)
* Инсталација на python пакетите:
  + Django – Geodjango доаѓа со него
  + Olwidget – OpenLayers widget
  + Django-shapes – овој пакет нуди класа ShpResponder која стримува зипувани shp датотеки. Користи ogr
  + Pip/virtualenv – полесен менаџмент на пајтон пакети (опционално)
  + Gunicorn – едноставен начин за хостирање на апликацијата на HTTP (опционално)

Апликацијата се хостира со WSGI сервер – Apache + mod\_wsgi, gunicorn…

При иницијално хостирање, се повикува менаџмент командата syncdb, што ги креира табелите во база мапирани со ORM-от, креира admin корисник и ги полни иницијалните региони. Апликацијата може привремено да се хостира и со серверот за развивање што доаѓа со џанго, што се повикува со менаџмент командата runserver.

# Заклучок

Ако го разгледаме кодот на реалтор ќе забележиме дека апликацијата е многу мала (во линии код) ако се земе во предвид се што може да се направи со неа. Благодарение на библиотеките и алатките кои ги искористив, голем дел од low-level функционалностите се апстрахирани, некомпатибилноста меѓу различи геопросторни бази е средена, апликацијата е стабилна и оптимизирана. Богатата документација придонесе времето на развивање значајно да се намали. Најважно од се, сите библиотеки кои ги искористив се бесплатни и со отворен код. Тоа уште повеќе ја олеснува работата бидејќи отворениот код е најдобра документација на една библиотека. Искористените пакети се BSD лиценцирани што овозможува користење на истите и за комерцијални пакети и затворени апликации.

Заедницата на Слободен Софтвер и отворен код нуди одлични решенија и во гранките на ГИС. Алатките и библиотеките кои ни се на располагање се добри колку и нивните proprietary пандани, редовно се развиваат, бесплатни се, одлично се документирани и тестирани во продукциони системи.

# Референци

1. <http://en.wikipedia.org/wiki/Free_and_open-source_software>

1. <http://www.osgeo.org/>

1. <http://en.wikipedia.org/wiki/Open_Source_Geospatial_Foundation>

1. <http://live.osgeo.org/en/overview/overview.html>
2. PostgreSQL 9 Administration Cookbook: Solve Real-world PostgreSQL Problems - Simon Riggs, Hannu Krosing 25

1. <http://postgis.net/docs/reference.html> Секција 8.7. Operators

1. <http://grass.osgeo.org/home/history/>

1. <https://speakerdeck.com/jacobian/django-the-web-framework-for-perfectionists-with-deadlines> – Jacob Caplan Moss

1. [https://docs.djangoproject.com](https://docs.djangoproject.com/)

1. <https://github.com/django/django>