Univerzitet u Novom Sadu

Tehnički fakultet »Mihajlo Pupin«

Zrenjanin

SEMINARSKI RAD

Predmet: Mobilne tehnologije i programiranje

Tema: Izrada android aplikacije Oglasi

Profesor: doc. dr Eleonora Brtka Student: Zorica Brkič SI 29/15

Smer: Softversko Inženjerstvo

Zrenjanin, 2019.

Sadržaj

[1. Predmetni cilj 3](#_Toc18947656)

[2. Uvod 4](#_Toc18947657)

[3. Sloj baze podataka 5](#_Toc18947658)

[4. Sloj REST servisa 6](#_Toc18947659)

[5. Frontend sloj 18](#_Toc18947660)

[5.1. Stanica oglasi pretraga 18](#_Toc18947661)

[5.2. Stranica kategorije pretraga 20](#_Toc18947662)

[5.3. Stranica dodavanje oglasa 20](#_Toc18947663)

[5.4. Stranica o aplikaciji 21](#_Toc18947664)

[6. Zaključak 23](#_Toc18947665)

# 1. Predmetni cilj

Cilj predmeta je sticanje osnovnih znanja i veština iz oblasti računarskog softvera i Interneta, kao suštinskih preduslova za dalje izučavanje informacionih i komunikacionih tehnologija i programiranja uopšte. Upoznavanje sa osnovama programiranja na mobilnim uređajima i osposobljavanje studenata da projektuju, kreiraju i održavaju aplikacija namenjene mobilnim platformama, sa posebnim akcentom na Android platformu u Android Studio razvojnom okruženju upotrebom programskog jezika JAVA. Predmet uvodi studente u programiranje mobilnih uređaja smart telefona i tableta.

# 2. Uvod

Android aplikacija Oglasi realizovana je kao troslojna aplikacija, slojevi su:

* Sloj baze podataka
* Sloj REST servisa
* Frontend sloj

Sloj baze podataka realizovan je putem phpMyAdmina, sloj REST servisa putem php programskog jezika, a frontend je radjen u Android Studio aplikaciji. Sva tri sloja čine jednu jedinstvenu celinu i jedan bez drugog nemaju smisla.

Svrha ove aplikacije jeste da olakša korisnicima da pristupaju oglasima, da ih lakše pronalaze i postavljaju. Ideja je da se napravi nešto slično sajtu “kupujem-prodajem”.

U današnje vreme kada su mobilni telefoni zastupljeniji od računara, veoma je važno biti konkurentan, pa tako sem računara imati i aplikaciju koja će moći da se instalira na mobilnom telefonu.

# 3. Sloj baze podataka

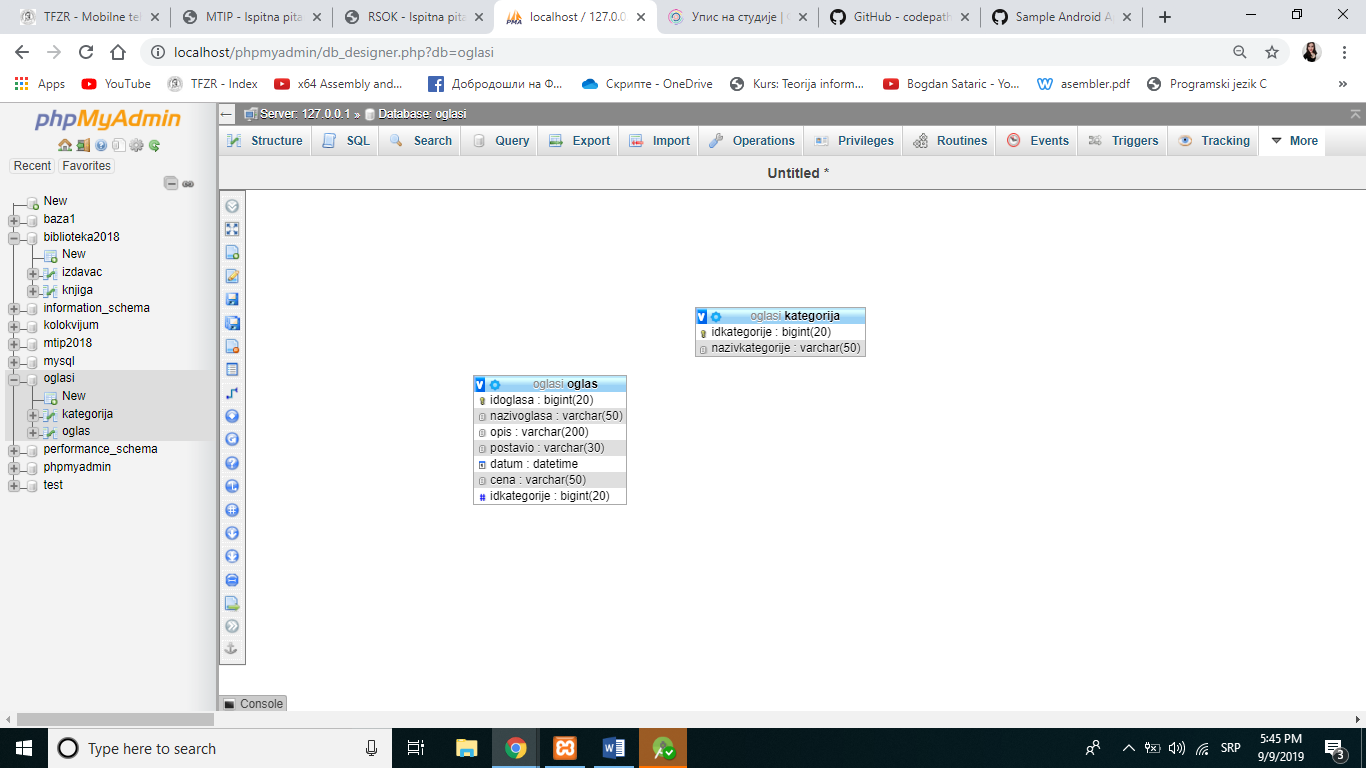
Već je spomenuto da je ovaj sloj realizovan putem phpMyAdmina. Baza se sastoji od dve tabele. Glavna tabela je tabela oglasi, a sporedna tabela kategorija. Tabela oglasi sadrži polja:

* idoglasa
* nazivoglasa
* opis
* postavio
* datum
* cena
* idkategorije (strani ključ)

, a tabela kategorija:

* idkategorije
* nazivkategorije

Ove dve tabele povezane su u relaciji jedan na prema više putem stranog ključa idkategorije.



*Slika 1. Izgled baze podataka*

# 4. Sloj REST servisa

REST servis povezuje frontend sloj sa slojem baze podataka. On putem HTTP metoda omogućuje dobavljanje podataka iz baze, upisivanje i njihovu izmenu. On služi kao međusloj. Format podataka koji se prezentuju na frontendu i format podataka u bazi nije isti, zato se ti podaci iz baze moraju konvertovati, u ovom slučaju u JSON format koji će se prezentovati na ekranima mobilnog telefona.

REST servis čine se od 2 foldera: lib i services. U folderu lib nalaze se podfolderi: class, DAL i utility. Folder class čine fajlovi:

* IFieldValidator
* Ktegorija
* Oglas
* inputFormaterCustom

U njemu su definisane klase koje imaju iste atribute kao i polja u tabelama u bazi podataka.

<?php

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

/\*\*

\* Description of Oglas

\*

\* @author Zorica

\*/

require\_once 'Kategorija.php';

class Oglas implements JsonSerializable, IFieldValidator{

//put your code here

protected $\_idoglasa;

protected $\_nazivoglasa;

protected $\_opis;

protected $\_postavio;

protected $\_datum;

protected $\_cena;

protected $\_kategorija;

public function \_\_construct() {

$this->\_idoglasa = 0;

$this->\_nazivoglasa = "";

$this->\_opis = "";

$this->\_postavio = "";

$this->\_datum = "";

$this->\_cena = 0;

$this->\_kategorija = new Kategorija();

}

function get\_idoglasa() {

return $this->\_idoglasa;

}

function get\_nazivoglasa() {

return $this->\_nazivoglasa;

}

function get\_opis() {

return $this->\_opis;

}

function get\_postavio() {

return $this->\_postavio;

}

function get\_datum() {

return $this->\_datum;

}

function get\_cena() {

return $this->\_cena;

}

function get\_kategorija() {

return $this->\_kategorija;

}

function set\_idoglasa($\_idoglasa) {

$this->\_idoglasa = $\_idoglasa;

}

function set\_nazivoglasa($\_nazivoglasa) {

$this->\_nazivoglasa = $\_nazivoglasa;

}

function set\_opis($\_opis) {

$this->\_opis = $\_opis;

}

function set\_postavio($\_postavio) {

$this->\_postavio = $\_postavio;

}

function set\_datum($\_datum) {

$this->\_datum = $\_datum;

}

function set\_cena($\_cena) {

$this->\_cena = $\_cena;

}

function set\_kategorija($\_kategorija) {

$this->\_kategorija = $\_kategorija;

}

public function ValidateFields() {

$this->\_idoglasa = formatInput($this->\_idoglasa);

$this->\_nazivoglasa = formatInput($this->\_nazivoglasa);

$this->\_opis = formatInput($this->\_opis);

$this->\_postavio = formatInput($this->\_postavio);

$this->\_datum = formatInput($this->\_datum);

$this->\_cena = formatInput($this->\_cena);

if ($this->\_kategorija!=null) $this->\_kategorija->ValidateFields();

}

public function jsonSerialize() {

return (object) get\_object\_vars($this);

}

public function jsonDeserialize($data) {

$this->\_idoglasa = $data->{'\_idoglasa'};

$this->\_nazivoglasa = $data->{'\_nazivoglasa'};

$this->\_opis = $data->{'\_opis'};

$this->\_postavio = $data->{'\_postavio'};

$this->\_datum= $data->{'\_datum'};

$this->\_cena = $data->{'\_cena'};

$kategorija = new Kategorija();

$kategorija->jsonDeserialize($data->{'\_kategorija'});

$this->\_kategorija = $kategorija;

}

}

*Listing 1. Klasa Oglas*

U folderu DAL smešteni su fajlovi koji komuniciraju sa bazom podataka. DAL je skraćenica od Data Access Layer, što bi značilo sloj za pristup bazi podataka. Njega čine sledeći fajlovi:

* DAL
* DALRezultatUpita
* KnjigaDAL
* OglasDAL
* ICommonDatabaseMethods

<?php

require\_once '../lib/DAL/DAL.php';

require\_once '../lib/DAL/KategorijaDAL.php';

require\_once '../lib/class/Oglas.php';

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

/\*\*

\* Description of OglasDAL

\*

\* @author Zorica

\*/

class OglasDAL extends DAL implements ICommonDatabaseMethods{

//put your code here

public function \_\_construct() {

parent::\_\_construct();

}

public function PrikaziOglasePretraga($search)

{

try

{

$tempOgl = new Oglas();

$tempOgl->set\_nazivoglasa($search);

$tempOgl->ValidateFields();

$sqlQuery="SELECT \* FROM oglas WHERE nazivoglasa LIKE :nazivoglasa";

$params = array(":nazivoglasa"=>"%".$tempOgl->get\_nazivoglasa()."%");

$results = $this->IzvrsiUpit($sqlQuery, $params);

$oglasResults = array();

foreach ($results as $k)

{

$oglasResult = new Oglas();

$oglasResult->set\_idoglasa($k->idoglasa);

$oglasResult->set\_nazivoglasa($k->nazivoglasa);

$oglasResult->set\_opis($k->opis);

$oglasResult->set\_postavio($k->postavio);

$oglasResult->set\_datum($k->datum);

$oglasResult->set\_cena($k->cena);

$kategorijaDAL = new KategorijaDAL();

$oglasResult->set\_kategorija($kategorijaDAL->GetOne($k->idkategorije));

$oglasResults[] = $oglasResult;

}

return $oglasResults;

}

catch (Exception $e)

{

return false; // ako nesto ne valja sa upitom, ne izvrsi se kako treba, vrati false

}

}

public function AddOne($object) {

try

{

$object->ValidateFields();

$sqlQuery="INSERT INTO oglas(nazivoglasa, opis, postavio, datum, cena, idkategorije) VALUES(:naziv, :opis, :postavio, :datum, :cena, :idkategorije)";

$params = array(":naziv"=>$object->get\_nazivoglasa(),

":opis" =>$object->get\_opis(),

":postavio"=>$object->get\_postavio(),

":datum"=>$object->get\_datum(),

":cena"=>$object->get\_cena(),

":idkategorije"=>$object->get\_kategorija()->get\_idkategorije());

return $this->IzvrsiUpit($sqlQuery, $params);

}

catch (Exception $e)

{

return false; // ako nesto ne valja sa upitom, ne izvrsi se kako treba, vrati false

}

}

public function DeleteOne($object) {

try

{

$object->ValidateFields();

$sqlQuery="DELETE FROM oglas WHERE idoglasa = :idoglasa";

$params = array(":idoglasa"=>$object->get\_idoglasa());

return $this->IzvrsiUpit($sqlQuery, $params);

}

catch (Exception $e)

{

return false; // ako nesto ne valja sa upitom, ne izvrsi se kako treba, vrati false

}

}

public function Duplicate($object) {

try

{

$object->ValidateFields();

$sqlQuery="SELECT \* FROM oglas WHERE nazivoglasa = :nazivoglasa AND opis = :opis AND postavio = :postavio AND datum = :datum AND cena =:cena AND idkategorije = :idkategorije";

$params = array(":nazivoglasa"=>$object->get\_nazivoglasa(),

":opis" =>$object->get\_opis(),

":postavio"=>$object->get\_postavio(),

":datum" =>$object->get\_datum(),

":cena"=>$object->get\_cena(),

":idkategorije"=>$object->get\_kategorija()->get\_idkategorije());

$results = $this->IzvrsiUpit($sqlQuery,$params);

if (count($results)>0) return true;

else return false;

}

catch (Exception $e)

{

return false; // ako nesto ne valja sa upitom, ne izvrsi se kako treba, vrati false

}

}

public function EditOne($object)

{

try

{

$object->ValidateFields();

$sqlQuery="UPDATE Oglas SET nazivoglasa = :nazivoglasa, opis = :opis, postavio = :postavio, datum = :datum, cena = :cena, idkategorije = :idkategorije WHERE idoglasa = :idoglasa";

$params = array(":nazivoglasa"=>$object->get\_nazivoglasa(),

":opis" =>$object->get\_opis(),

":postavio"=>$object->get\_postavio(),

":datum"=>$object->get\_datum(),

":cena"=>$object->get\_cena(),

":idkategorije"=>$object->get\_kategorija()->get\_idkategorije(),

":idoglasa"=>$object->get\_idoglasa());

return $results = $this->IzvrsiUpit($sqlQuery, $params);

}

catch (Exception $e)

{

return false; // ako nesto ne valja sa upitom, ne izvrsi se kako treba, vrati false

}

}

public function GetAll() {

try

{

$sqlQuery="SELECT \* FROM oglas";

$params = array();

$results = $this->IzvrsiUpit($sqlQuery,$params);

$studResults = array();

foreach ($results as $k)

{

$oglasResult = new Oglas();

$oglasResult->set\_idoglasa($k->idoglasa);

$oglasResult->set\_nazivoglasa($k->nazivoglasa);

$oglasResult->set\_opis($k->opis);

$oglasResult->set\_postavio($k->postavio);

$oglasResult->set\_datum($k->datum);

$oglasResult->set\_cena($k->cena);

$kategorijaDAL = new KategorijaDAL();

$oglasResult->set\_kategorija($kategorijaDAL->GetOne($k->idkategorije));

$oglasResults[] = $oglasResult;

}

return $oglasResults;

}

catch (Exception $e)

{

return false; // ako nesto ne valja sa upitom, ne izvrsi se kako treba, vrati false

}

}

public function GetOne($object) {

try

{

$oglasResult = new Oglas();

$oglasResult->set\_idoglasa($object);

$sqlQuery="SELECT \* FROM oglas WHERE idoglasa = :idoglasa";

$params = array(":idoglasa" =>$oglasResult->get\_idoglasa());

$results = $this->IzvrsiUpit($sqlQuery,$params);

if (count($results)>0)

{

$k = $results[0];

$oglasResult = new Oglas();

$oglasResult->set\_idoglasa($k->idoglasa);

$oglasResult->set\_nazivoglasa($k->nazivoglasa);

$oglasResult->set\_opis($k->opis);

$oglasResult->set\_postavio($k->postavio);

$oglasResult->set\_datum($k->datum);

$oglasResult->set\_cena($k->cena);

$kategorijaDAL = new KategorijaDAL();

$oglasResult->set\_kategorija($kategorijaDAL->GetOne($k->idkategorije));

}

return $oglasResult;

}

catch (Exception $e)

{

return false; // ako nesto ne valja sa upitom, ne izvrsi se kako treba, vrati false

}

}

public function RecordExists($object) {

try

{

$sqlQuery="SELECT \* FROM oglas WHERE idoglasa = :idoglasa";

$params = array(":idoglasa" =>(int)$object);

$results = $this->IzvrsiUpit($sqlQuery,$params);

if (count($results)>0) return true;

else return false;

}

catch (Exception $e)

{

return false; // ako nesto ne valja sa upitom, ne izvrsi se kako treba, vrati false

}

}

}

*Listing 2. Izgled klase OglasDAL*

# 5. Frontend sloj

Ovaj sloj je sloj koji komunicira sa bazom podataka putem REST servisa. Takođe ovo je sloj koji komunicira i sa korisnicima, pa je on zato interakrivan. Ovaj sloj sem što se prezentuje korisnicima, tj. u njemu se definiše sam izgled aplikacije, on mora i da omogući sve funkcionalnosti aplikacije.

Ovaj sloj rađen je u Android Studi-u u korespodenciji sa Alcatel android mobilnim telefonom koji je služio za testiranje aplikacije.

Aplikacija se sastoji od 4 foldera, a to su:

* manifests
* java
* generatedJava
* res

U folderu manifests smešten je fajl AndroidManifest.xml u kome se nalazi konfiguracija cele aplikacije.

U folderu java nalaze se sve java klase koje komuniciraju sa REST servisom. U njoj su definisani fragmenti koji kontrolišu ponašanje aplikacije. U ovom folderu je definisan funkcionalni deo aplikacije. Svi fajlovi pisani su na programskom jeziku JAVA.

U folderu generatedJava smeštena je konfiguracija builda aplikacije.

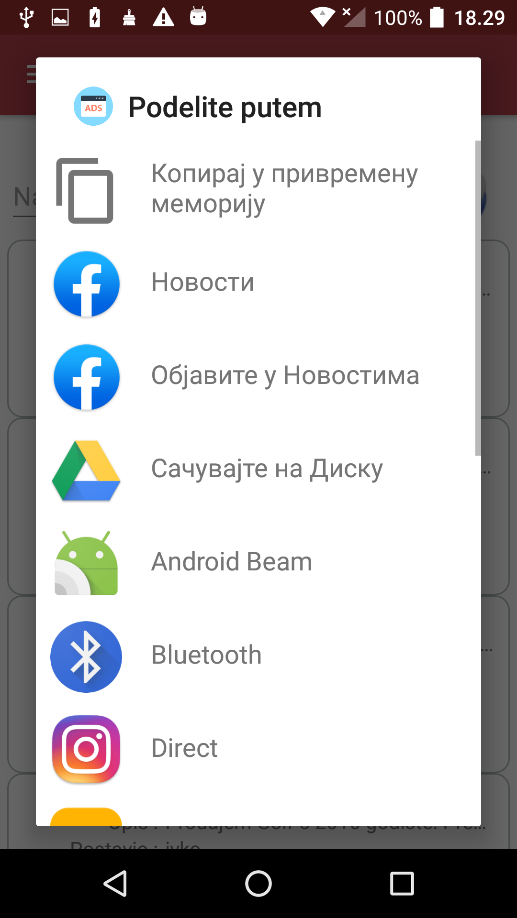
Folder res bavi se izgledom aplikacije. U njemu su definisane boje, slike, ikonice i stringovi koji će se prikazivati na ekranima mobilnih telefona.

## 5.1. Stanica oglasi pretraga

Ova stranica služi za pretraživanje postojećih oglasa, kao i njihovog deljenja (share). Olgasi se prikazuju u formi liste. Omogućen je skrol stranice u slučaju prevelikog broja oglasa.



*Slika 2. Izgled stranice oglasi pretraga*

**

*Slika 3. Izgled share dodatka putem stranice pretraga oglasi*

## 5.2. Stranica kategorije pretraga

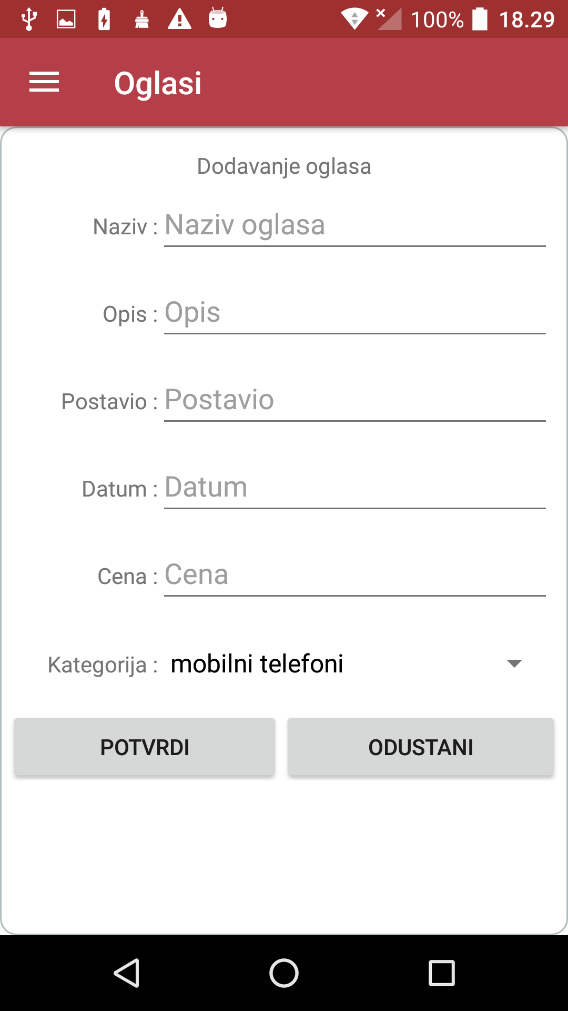
Ova stranica služi za pretraživanje postojećih kategoria, kao i njihovog deljenja (share). Kategorije se prikazuju u formi liste. Omogućen je skrol stranice u slučaju prevelikog broja oglasa.



*Slika 4. Izgled stranice kategorije pretraga*

## 5.3. Stranica dodavanje oglasa

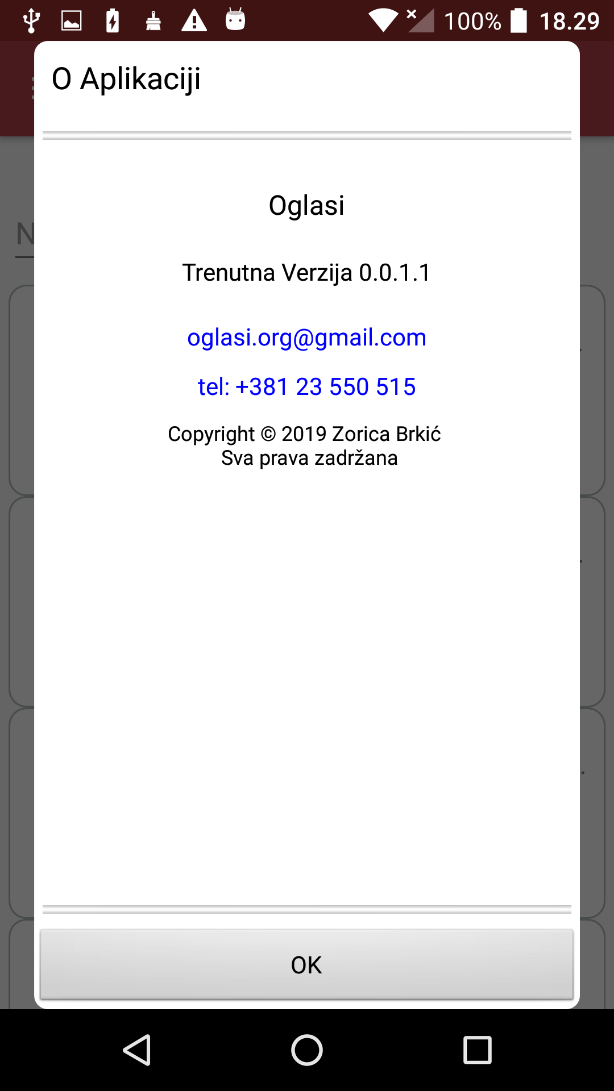
Putem ove stranice korisnicima se omogućuje da sem pregleda oglasa i kategorija oni mogu da sami postavljaju svoje oglase. Da bi oglas bio uspešno postavljen sva polja moraju biti popunjena, a polje kategorije bira se iz ponuđene liste postojećih kategorija. Međutim korisniku je omogućeno da doda kategoriju putem stranice za dodavanje kategorija.



*Slika 6. Izgled stranice za dodavanje oglasa*

## 5.4. Stranica o aplikaciji

U ovoj stranici nalaze se neki osnovni podaci o samoj aplikaciji kao i o tvorcu aplikacije. Takođe tu se i nalaze e-mail adresa i broj telefona putem kojih se mogu kontaktirati osobe zadužene za održavanje aplikacije u slučaju nekih nepravilnosti ili nedoumica od strane korisnika.



*Slika 7. Izgled stranice o aplikaciji*

# 6. Zaključak

Sa obzirom na to da u današnje vreme mobilni telefoni postaju računari u malom i postaju sve rasprostranjeniji, budućnost android aplikacija tek počinje. Mobilni telefoni danas postoje u većem broju u odnosu na računare i njihova prednost jeste to što su lako prenosivi i ljudi ih imaju u svakom trenutku uz sebe. Ova aplikacija ima dosta prednosti, ali i mana. Ustvari to nisu prave mane, već neke funkcionalnosti koje mogu biti lako nadograđene. Jedna od njih bila bi da se korisnicima omogući logovanje, pa tako bi mogli samo oni koji imaju svoj profil da kače oglase. Takođe bilo bi korisno omogućiti i razmenu poruka izmešu registrovanih korisnika. To su samo neke od ideja koje bi mogle biti ostvarene već u nekim novim verzijama aplikacije.