



UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

APLIGS

aplikacija za iskanje glasbenikov in glasbenih skupin

IZDELAL: Mitja J. Kotnik

LJUBLJANA, 2017

Predmet: Tehnologija programske opreme



Kazalo vsebine

Podatkovna baza	1
Fizični model podatkovne baze	1
Konceptualni model podatkovne baze	2
Tabele	3
Tabela: igra	3
Kolone	3
Tabela: instrument	3
Kolone	3
Tabela: izvaja	3
Kolone	3
Tabela: lokacija	3
Kolone	3
Tabela: oglas	3
Kolone	3
Tabela: sporočilo	4
Kolone	4
Tabela: uporabnik	4
Kolone	4
Tabela: vloga	4
Kolone	4
Tabela: zvrst	5
Kolone	5
REST Storitve	6
Oglasi	6
Uporabniki	7
Sporočila	8
Šifranti	9



Kazalo slik

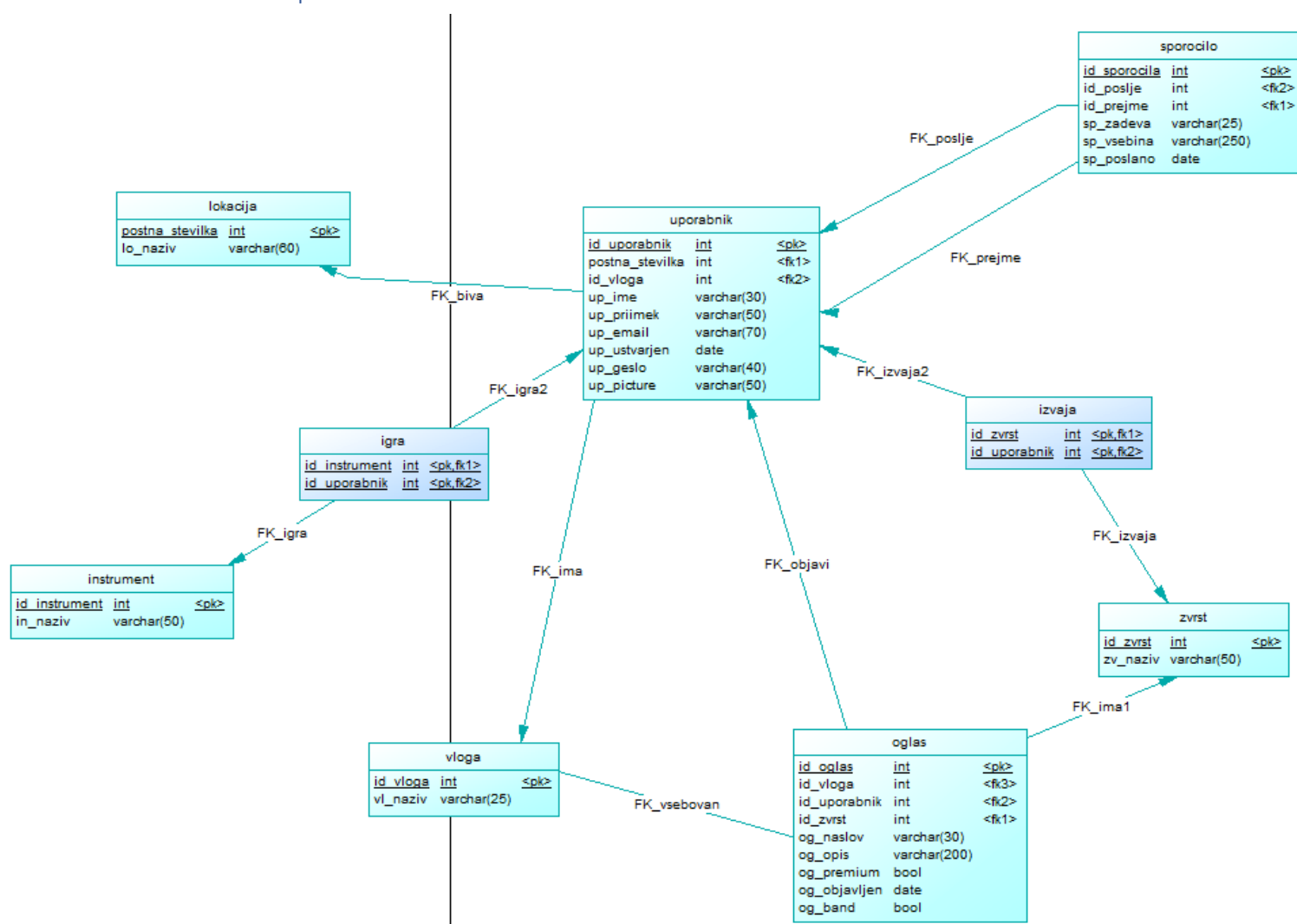
Slika 1: Fizični podatkovni model	1
Slika 2: Konceptualni model podatkovne baze	2
Slika 3: GET odgovor JSON.....	6
Slika 4: JSON format za POST zahtevek	6
Slika 5: GET odgovor za vse uporabnike	7
Slika 6: Format JSON za POST zahtevek	7
Slika 7: GET zahtevek za sporočila	8
Slika 8: Oblika JSON zahtevka za POST	8
Slika 9: GET zvrsti.....	9
Slika 10: GET vloge.....	9
Slika 11: GET lokacije	9
Slika 12: GET inštrumenti	9

Podatkovna baza

Podatkovno bazo smo najprej predvideli katere entitete sploh potrebujemo in s pomočjo pretvorbe v 3. normalno obliko naredili nato tudi povezave med njimi. Konceptualni model smo izdelali s pomočjo orodja PowerDesigner, kjer smo ga nato pretvorili še v fizični model. S pomočjo orodja smo nato ustvarili .sql datoteko apligs_CREATE_DATABASE.sql, ki ustvari našo podatkovno bazo na MySQL strežniku. Vstavili smo ga s pomočjo MySQL Workbench orodja.

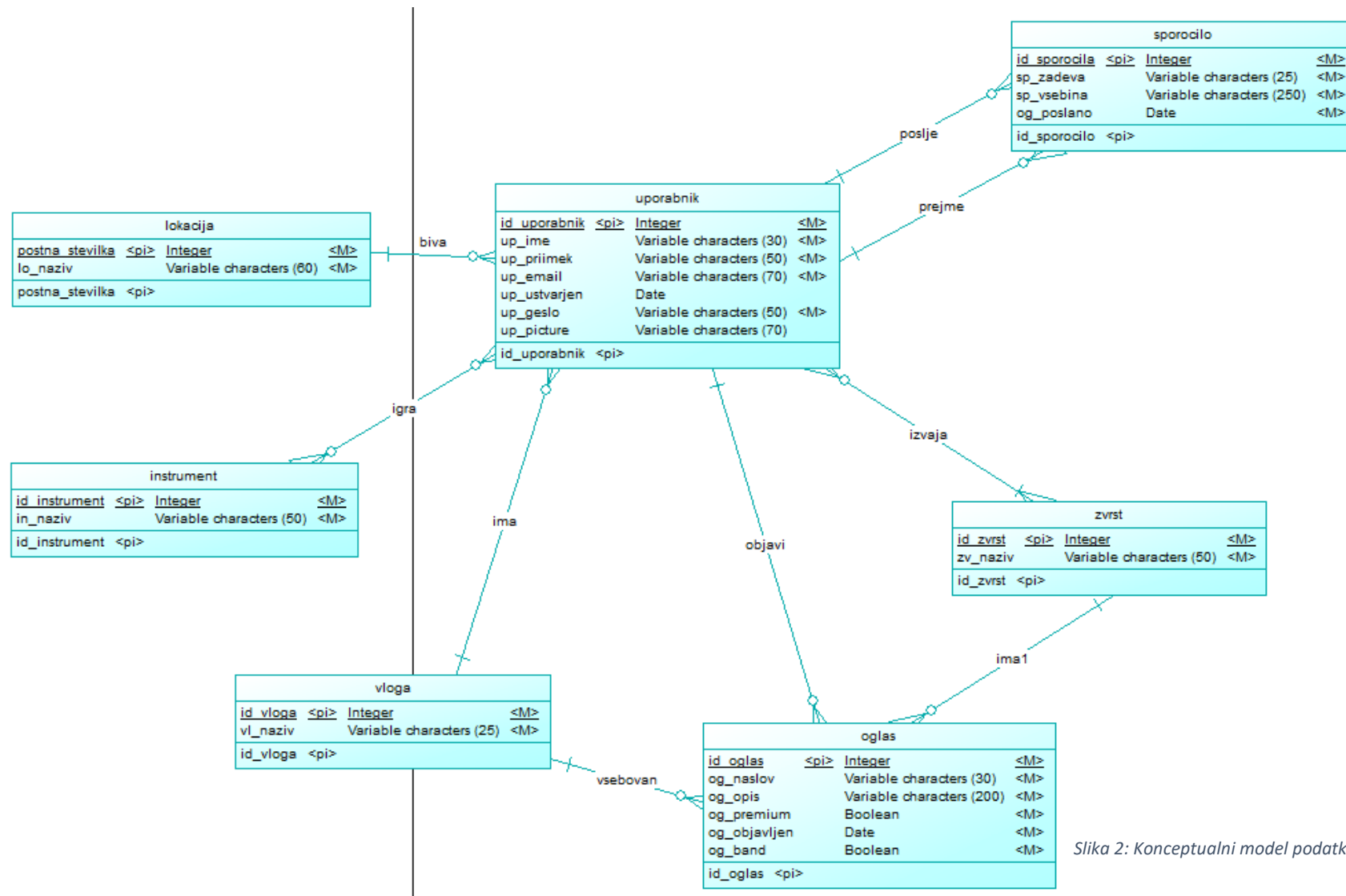
Nato smo ustvarili testne podatke, ki smo jih shranili v datoteko apligsTestData_INSERT_INTOTO.sql.

Fizični model podatkovne baze



Slika 1: Fizični podatkovni model

Konceptualni model podatkovne baze



Slika 2: Konceptualni model podatkovne baze

Tabele

Tabela: igra

Vmesna tabela za povezavo M:M med entitetama inštrument in uporabnik.

Kolone

	Name	Data type	Description
	id_instrument	int(10, 0)	Tuji ključ za inštrument
	id_uporabnik	int(10, 0)	Tuji ključ za uporabnika

Tabela: instrument

Tabela, ki je šifrant za vse inštrumente, ki jih je mogoče igrati.

Kolone


	Name	Data type	Description
	id_instrument	int(10, 0)	Enolični identifikator za inštrument (primary key)
	in_naziv	varchar(50)	Naziv inštrumenta

Tabela: izvaja

Vmesna tabela za povezavo M:M med tabelama uporabnik in zvrst.

Kolone

	Name	Data type	Description
	id_zvrst	int(10, 0)	Tuji ključ za zvrst
	id_uporabnik	int(10, 0)	Tuji ključ za uporabnika

Tabela: lokacija

Tabela za različna mesta po Sloveniji.

Kolone



	Name	Data type	Description
	postna_stevilka	int(10, 0)	Poštna številka, ki določa identifikator za to tabelo
	lo_naziv	varchar(60)	Naziv kraja, ki je enoličen glede na poštno številko

Tabela: oglas

Tabela, kjer so podatki za vse objavljene oglase.

Kolone

	Name	Data type	Description
	id_oglas	int(10, 0)	Identifikator za tabelo oglas (primarni ključ)
	id_vloga	int(10, 0)	Tuji ključ za povezavo med tabelama oglas in vloga
	id_uporabnik	int(10, 0)	Tuji ključ za povezavo med tabelama oglas in uporabnik
	id_zvrst	int(10, 0)	Tuji ključ za povezavo med tabelama oglas in zvrst



	Name	Data type	Description
	og_naslov	varchar(30)	Naslov oglasa
	og_objavljen	date	Datum kdaj je bil oglas objavljen
	og_opis	varchar(200)	Opis oglasa
	og_premium	tinyint(3, 0)	Opisuje ali je oglas premium ali ne (1 - premium, 0 - ni premium)
	og_band	tinyint(3, 0)	Opisuje ali je oglas objavila skupina ali posameznik (1 - skupina, 0 - posameznik)

Tabela: sporocilo

Tabela, kjer se hranijo podatki poslanih/prejetih sporočil.

Kolone

	Name	Data type	Description
🔑	id_sporocila	int(10, 0)	Enolični identifikator za sporočilo (primary key)
	id_poslje	int(10, 0)	Tuji ključ za uporabnika. Pove nam kdo je sporočilo poslal.
	id_prejme	int(10, 0)	Tuji ključ za uporabnika. Pove nam kdo je sporočilo prejel.
	sp_zadeva	varchar(25)	Zadeva (naslov) sporočila
	sp_vsebina	varchar(250)	Vsebina, ki jo pošlje uporabnik.
	sp_poslano	date	Datum kaj je bilo sporočilo poslano.

Tabela: uporabnik

Tabela, kjer so shranjeni podatki vseh uporabnikov.

Kolone

	Name	Data type	Description
🔑	id_uporabnik	int(10, 0)	Enolični identifikator za uporabnike (primary key)
	postna_stevilka	int(10, 0)	Tuji ključ za lokacijo. Določa v katerem mestu živi uporabnik.
	id_vloga	int(10, 0)	Tuji ključ za vlogo. Določa kakšno vlogo ima uporabnik.
	up_ime	varchar(30)	Ime uporabnika.
	up_priimek	varchar(50)	Priimek uporabnika.
	up_email	varchar(70)	E-pošta uporabnika. Prav tako se uporablja za vpis.
	up_geslo	varchar(50)	Uporabniško geslo za vpis uporabnika.
	up_ustvarjen	date	Datum kdaj je uporabnik ustvaril račun.
	up_picture	varchar(50)	Pot kje se na strežniku nahaja slika uporabnika.

Tabela: vloga

Tabela, ki definira obstoječe vloge.

Kolone


	Name	Data type	Description
🔑	id_vloga	int(10, 0)	Enolični identifikator za tabelo vloga (primary key)
	vl_naziv	varchar(25)	Naziv vloge



Tabela: zvrst

Tabela, ki definira vse zvrsti, ki jih lahko izvajajo uporabniki in skupine.

Kolone

	Name	Data type	Description
	id_zvrst	int(10, 0)	Enolični identifikator za zvrst (primary key)
	zv_naziv	varchar(50)	Naziv zvrsti



REST Storitve

Oglasi

GET zahtevki:

<http://localhost:8080/aplign-rest/v1/oglasl>

Vrne celoten seznam vseh oglasov, ki se nahajajo v podatkovni bazi.

<http://localhost:8080/aplign-rest/v1/oglasl?offset=1&limit=3>

Vrne seznam oglasov, katerim določimo offset in limit. Torej offset pomeni, da prične od 1 oglasa in vrne število oglasov, ki je določeno z limit (v tem primeru 3). Uporablja se za odstranjevanje.

POST zahtevki:

<http://localhost:8080/aplign-rest/v1/oglasl>

Ob POST zahtevki moramo v Headers definirati 'Content-Type', 'application/json'. Pošljemo potem podatke v obliki JSON, da shrani v PB. Oblika je podana v Slika 2.

PUT zahtevki:

<http://localhost:8080/aplign-rest/v1/oglasl>

Nam omogoča da posodobimo določen oglas. Pošljemo isti format zapisa, kjer samo uredimo polja, ki jih želimo spremeniti. Obvezno moramo pa poleg še podati polje v JSON format "idOglas", saj to pomaga določiti katerega je potrebno uredit. Primer: "idOglas": "1" bo uredil uporabnika z ID-jem 1.

DELETE zahtevki:

<http://localhost:8080/aplign-rest/v1/oglasl/6>

Z DELETE zahtevkom izbrišemo oglas, ki ga določimo z ID-jem na koncu URL-ja.

```

{
  "idOglas": 1,
  "ogNaslov": "Rock-punk",
  "ogObjavljen": "2016-12-01",
  "ogOpis": "Iscemo pianista za jazz skupino",
  "ogPremium": 0,
  "ogBand": 1,
  "zvvrst": {
    "idZvrst": 5,
    "zvNaziv": "reggae"
  },
  "uporabnik": {
    "idUporabnik": 3,
    "upEmail": "ak1729@student.uni-lj.si",
    "upGeslo": "root",
    "upIme": "Andrej",
    "upPriimek": "Kokelj",
    "upUstvarjen": "2016-12-05",
    "upPicture": "/img/persons/picture.jpg",
    "lokacija": {
      "postnaStevilka": 3301,
      "loNaziv": "Petrovce"
    }
  },
  "vloga": {
    "idVloga": 5,
    "vlNaziv": "kitarist"
  }
},
{
  "vloga": {
    "idVloga": 2,
    "vlNaziv": "basist"
  }
}
}

```

Slika 3: GET odgovor JSON

```

{
  "ogNaslov": "Testiram",
  "ogObjavljen": "2016-12-01",
  "ogOpis": "Test post metode",
  "ogPremium": 0,
  "ogBand": 1,
  "zvvrst": {
    "idZvrst": 3
  },
  "uporabnik": {
    "idUporabnik": 5
  },
  "vloga": {
    "idVloga": 1
  }
}

```

Slika 4: JSON format za POST zahtevek

Uporabniki

GET zahtevki:

<http://localhost:8080/aplign-rest/v1/uporabniki>

Vrne celoten seznam uporabnikov.

<http://localhost:8080/aplign-rest/v1/uporabniki?offset=2&limit=3>

Vrne seznam uporabnikov, ki ga poljubno določimo. Uporablja se za odstranjevanje.

<http://localhost:8080/aplign-rest/v1/uporabniki/prijava?email=mj0073@student.uni-lj.si&password=mitja>

Vrne podatke o uporabniku, če so podatki za prijavo točni. V primeru, da podatki niso pravilni vrne JSON objekt s statusom 204 No Content

POST zahtevek:

<http://localhost:8080/aplign-rest/v1/uporabniki/>

Ob POST zahtevki moramo v Headers definirati 'Content-Type', 'application/json'. Pošljemo potem podatke v obliki JSON, da shrani v PB.

PUT zahtevek:

<http://localhost:8080/aplign-rest/v1/uporabniki/>

Nam omogoča da posodobimo določenega uporabnika. Pošljemo isti format zapisa, kjer samo uredimo polja, ki jih želimo spremeniti.

Obvezno moramo pa poleg še podati polje v JSON format

"idUporabnik", saj to pomaga določiti katerega je potrebno uredit. Primer: "idUporabnik": "10" bo uredil uporabnika z ID-jem 10.

DELETE zahtevek:

<http://localhost:8080/aplign-rest/v1/uporabniki/11>

Omogoča, da izbrišemo uporabnika z željenim ID-jem. V našem primeru je to uporabnik z ID-jem 11.

```
{
  "idUporabnik": 1,
  "upEmail": "rs0424@student.uni-lj.si",
  "upGeslo": "root",
  "upIme": "Rok",
  "upPriimek": "Solar",
  "upUstvarjen": "2016-11-22",
  "upPicture": "/img/persons/picture.jpg",
  "lokacija": {
    "postnaStevilka": 2000,
    "loNaziv": "Maribor"
  },
  "vloga": {
    "idVloga": 5,
    "vlNaziv": "kitarist"
  }
}
```

Slika 5: GET odgovor za vse uporabnike

```
{
  "upEmail": "test@student.uni-lj.si",
  "upGeslo": "root",
  "upIme": "Tester",
  "upPriimek": "Testnik",
  "upUstvarjen": "2016-11-22",
  "upPicture": "/img/persons/picture.jpg",
  "lokacija": {
    "postnaStevilka": 2000,
    "loNaziv": "Maribor"
  },
  "vloga": {
    "idVloga": 5,
    "vlNaziv": "kitarist"
  }
}
```

Slika 6: Format JSON za POST zahtevek

Sporočila

GET zahtevek:

<http://localhost:8080/apligns-rest/v1/sporocila>

Vrne celoten seznam sporočil, ki smo jih imamo v podatkovni bazi. Prav tako ima možnost tudi kot pri prejšnjih dveh opcijo po offset in limit za odstranjevanje.

<http://localhost:8080/apligns-rest/v1/sporocila/1>

Vrne sporočilo z željenim ID-jem, ki ga podamo v URL.

<http://localhost:8080/apligns-rest/v1/sporocila/prejeto/1>

Vrne seznam sporočil, ki jih je prejel uporabnik z ID-jem 1.

<http://localhost:8080/apligns-rest/v1/sporocila/poslano/1>

Vrne seznam sporočil, ki jih je poslal uporabnik z ID-jem 1.

POST zahtevek:

<http://localhost:8080/apligns-rest/v1/sporocila/>

Pošlje sporočilo uporabniku, ki ga določimo v JSON sporočilu. Uporabnik1 je prejemnik, medtem, ko je uporabnik2 pošiljatelj.

PUT zahtevek:

Sporočila se, ko so enkrat poslana **ne** morejo več urejati.

DELETE zahtevek:

<http://localhost:8080/apligns-rest/v1/sporocila/8>

Sporočilo izbrišemo s pomočjo DELETE zahtevka, kjer v URL-ju definiramo ID sporočila, ki ga želimo izbrisati.

```
{
  "idSporocila":1,
  "spPoslano":"2016-12-13",
  "spVsebina":"Test aplikacije",
  "spZadeva":"Pozdrav",
  "uporabnik1":{
    "idUporabnik":1,
    "upEmail":"rs0424@student.uni-lj.si",
    "upGeslo":"root",
    "upIme":"Rok",
    "upPriimek":"Solar",
    "upUstvarjen":"2016-11-22",
    "upPicture":"/img/persons/picture.jpg",
    "lokacija":{
      "postnaStevilka":2000,
      "loNaziv":"Maribor"
    }
  },
  "vloga":{
    "idVloga":5,
    "vNaziv":"kitarist"
  }
},
"uporabnik2":{
  "idUporabnik":2,
  "upEmail":"mj0073@student.uni-lj.si",
  "upGeslo":"mitja",
  "upIme":"Mitja",
  "upPriimek":"Kotnik",
  "upUstvarjen":"2016-02-25",
  "upPicture":"/img/persons/picture.jpg",
  "lokacija":{
    "postnaStevilka":2380,
    "loNaziv":"Slovenj Gradec"
  },
  "vloga":{
    "idVloga":3,
    "vNaziv":"back vocal"
  }
}
}
```

Slika 7: GET zahtevek za sporočila

```
{
  "spPoslano":"2016-12-13",
  "spVsebina":"Test metode za pošiljanje",
  "spZadeva":"POST",
  "uporabnik1":{
    "idUporabnik":5
  },
  "uporabnik2":{
    "idUporabnik":3
  }
}
```

Slika 8: Oblika JSON zahtevka za POST

Šifranti

Predstavljamo vse metode, ki so na voljo za pridobivanje šifrantov v aplikaciji. Ti šifranti so lokacija, inštrument, vloga, ter zvrst.

Prav tako lahko vsakega izmed njih podrobneje izberemo tako, da dodamo njegov ID na konec URL-ja, npr /12.

GET zahtevek (inštrument):

<http://localhost:8080/aplign-rest/v1/instrumenti>

Vrne celoten seznam inštrumentov, ki so v aplikaciji.

GET zahtevek (lokacija):

<http://localhost:8080/aplign-rest/v1/lokacije>

Vrne celoten seznam lokacij, ki so v aplikaciji.

GET zahtevek (vloga):

<http://localhost:8080/aplign-rest/v1/vloge>

Vrne celoten seznam vlog, ki so v aplikaciji.

GET zahtevek (zvrst):

<http://localhost:8080/aplign-rest/v1/zvrsti>

Vrne celoten seznam zvrsti, ki so v aplikaciji.

```
[
  {
    "idInstrument": 1,
    "inNaziv": "kitara"
  },
  {
    "idInstrument": 2,
    "inNaziv": "bas kitara"
  },
  {
    "idInstrument": 3,
    "inNaziv": "bobni"
  },
  {
    "idInstrument": 4,
    "inNaziv": "harmonika"
  },
  {
    "idInstrument": 5,
    "inNaziv": "klavir"
  },
  {
    "idInstrument": 6,
    "inNaziv": "klarinet"
  }
]
```

Slika 12: GET inštrumenti

```
[
  {
    "postnaStevilka": 1000,
    "loNaziv": "Ljubljana"
  },
  {
    "postnaStevilka": 1290,
    "loNaziv": "Grosuplje"
  },
  {
    "postnaStevilka": 2000,
    "loNaziv": "Maribor"
  },
  {
    "postnaStevilka": 2380,
    "loNaziv": "Slovenj Gradec"
  },
  {
    "postnaStevilka": 3301,
    "loNaziv": "Petrovce"
  }
]
```

Slika 11: GET lokacije

```
[
  {
    "idVloga": 1,
    "vlNaziv": "pevec"
  },
  {
    "idVloga": 2,
    "vlNaziv": "basist"
  },
  {
    "idVloga": 3,
    "vlNaziv": "back vocal"
  },
  {
    "idVloga": 4,
    "vlNaziv": "pianist"
  },
  {
    "idVloga": 5,
    "vlNaziv": "kitarist"
  }
]
```

Slika 10: GET vloge

```
[
  {
    "idZvrst": 1,
    "zvNaziv": "rap"
  },
  {
    "idZvrst": 2,
    "zvNaziv": "rock"
  },
  {
    "idZvrst": 3,
    "zvNaziv": "salsa"
  },
  {
    "idZvrst": 4,
    "zvNaziv": "country"
  },
  {
    "idZvrst": 5,
    "zvNaziv": "reggae"
  }
]
```

Slika 9: GET zvrsti