SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

**Matija Čoklica, 45270/16-R**

**Valentina Krhlanko, 45268/16-R**

**Aleksandar Trajkov, 45280/16-R**

FoodDonor

PROJEKT IZ KOLEGIJA ANALIZA I RAZVOJ PROGRAMA

PROJEKTNA DOKUMENTACIJA

[Link na github repozitorij](https://github.com/valkrhlan/FoodDonorWebservice)

Varaždin, 2016.

Sadržaj

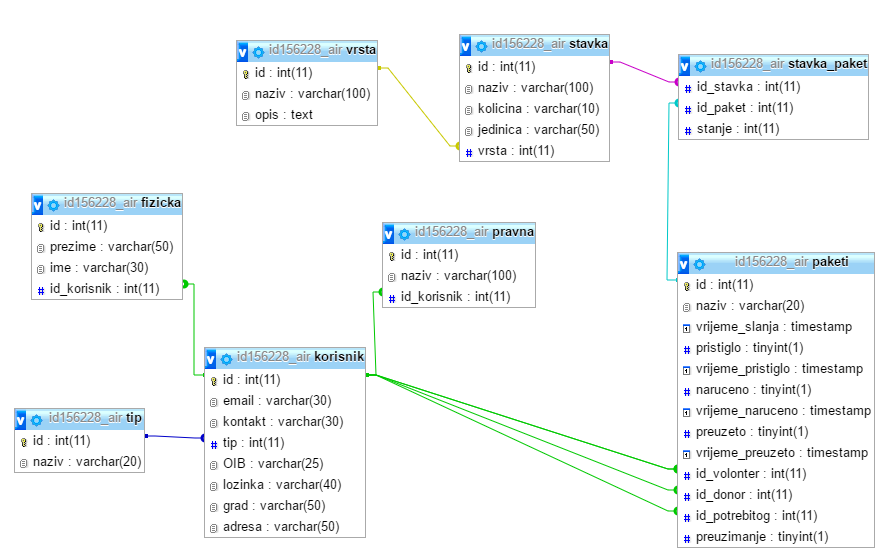
[Baza podataka 1](#_Toc467001534)

[Web servis 2](#_Toc467001535)

[Struktura aplikacije 5](#_Toc467001536)

# Baza podataka

Tablica korisnik je tablica koja sadrži zajednička obilježja za sve tipove korisnika. Kao primarni ključ koristit će se atribut id. Atribut email predstavlja oznaku korisničkog imena . Tablica tip sadrži moguće tipove korisnika koji nam kasnije služe za definiranje akcija kojima korisnik ima pristup. Trenutno su moguće vrijednosti : „volonter“,“ donor“ i „potrebiti“. Tablica fizicka i pravna su dvije potkategorije korisnika. Tablica fizicka predstavlja dodatne informacije o tipu korisnika „volonter“. Tablica pravna predstavlja dodatne atribute koje mora imati potrebiti korisnik ili donor ako pretpostavimo da takvi korisnici moraju biti poduzeća ili urduge, a ne pojedinci. Tablica paket ima prema tablici stavka vezu n-n te se za prikaz svih stavki hrane u paketu koristi pomoćna tablica stavka\_paketa.



Slika : ERA model ([link na sliku](http://i.imgur.com/VM8kQIN.png))

# Web servis

Kako našu aplikaciju istovremeno može koristiti više korisnika koji trebaju podatke iz zajedničke baze napravljen je PHP web servis. Pripadni web servis je RESTfull tipa te se za dohvaćanje ili manipulaciju podataka treba organizirati određena struktura URL-a. Link na github repozitorij web servisa je <https://github.com/valkrhlan/FoodDonorWebservice> Kao nepromjenjivi dio tj. baseUrl u aplikaciji korišten je url: <https://air-web-service.000webhostapp.com/webservice/>. Za prvu fazu su implementirane četiri metode, a njihov naziv, HTTP metodu i url možete vidjeti u tablici 1. Metode web *servisa*. Pristupni url se kreira tako da *baseUrl*-u dodamo url metode.

|  |  |  |
| --- | --- | --- |
| WS METODA | HTTP METODA | URL |
| Dohvati sve gradove | GET | gradovi/all |
| Prijava | GET | prijava/{email}/{lozinka}/ |
| Registracija volontera | GET | registracija/{metoda}/{email}/{lozinka}/{oib}/{grad}/{adresa}/{kontakt}/{ime}/{prezime}/ |
| Registracija volontera | GET | registracijaOstali/{metoda}/{email}/{lozinka}/{oib}/{grad}/{adresa}/{kontakt}/{naziv}/{tip}/ |
| Dohvaćanje svih vrsta hrane i jedinica količine | GET | vrstaJedinica/all/ |

Tablica : Metode web servisa

Svaka funkcija vraća odgovor JSON formata. Glavni atributi u odgovoru su status, nbResults, message i dana. Pri čemu status označava uspješno ili uspješno dohvaćanje podataka sa web servisa, a prikazuju i ispravnost ili neispravnost pri izmjeni ili dodavanju podataka u bazu. Strukturu odgovora i moguće vrijednosti koje može poprimiti možemo vidjeti u tablici *2. Struktura odgovora pri metodi za ispis svih gradova*, tablici *3. Struktura odgovora pri metodi za prijavu* i tablici *4. Struktura odgovora pri metodama za registraciju* nam pokazuje sve moguće vrijednosti odgovora prilikom dohvaćanja svih gradova iz baze.

|  |  |  |
| --- | --- | --- |
| RESPONSE | Nema gradova u bazi | Dohvaćeni su gradovi |
| status | OK | OK |
| nbResults | 0 | Broj dohvaćenih gradova |
| Message | „Nema ni jednog grada u bazi!“ | „Pronađeni su gradovi“ |
| data | Niz pbr = -1 i naziv =““ | Niz svih gradova sa vrijednostima pbr i naziv u JSON formatu |

Tablica : Struktura odgovora pri metodi za ispis svih gradova

Svi mogući odgovori prilikom metode za prijavu vidljivi su u tablici *3. Za registraciju postoje dvije funkcije, jedna omogućava registraciju volontera, tj. fizičkih osoba, a druga omogućava registraciju donora i potrebitih koji su pravne osobe. Obje funkcije za registraciju vraćaju odgovor iste strukture,pa su zajedno obuhvaćene u tablici \_4.Struktura odgovora pri metodama za registraciju*

|  |  |  |  |
| --- | --- | --- | --- |
| RESPONSE | Uspješna prijava | Neuspješna prijava (krivi uneseni podatci) | Neuspješna prijava (nedostupna baza podataka) |
| status | OK | NOT OK | NOT OK |
| nbResults | Tip korisnika ( 1 – „donor“, 2 – „volonter“ i 3 –„ potrebiti“) | 0 | 0 |
| message | „Uspješna prijava“ | tekst greški | "Došlo je do problema na web servisu.“ |
| data | Niz prijava = „OK“ | Niz kojem je prijava = „error“ | Niz kojem je prijava = ''error'' |

Tablica :Struktura odgovora pri metodi za prijavu

|  |  |  |
| --- | --- | --- |
| RESPONSE | Uspješna registracija | Neuspješna registracija |
| status | OK | NOT OK |
| nbResults | 0 | 0 |
| Message | „Uspješna registracija“ | tekst greški |
| data | Niz reg = „OK“ | Niz kojem je reg= „error“ |

Tablica : Struktura odgovora pri metodi za registraciju

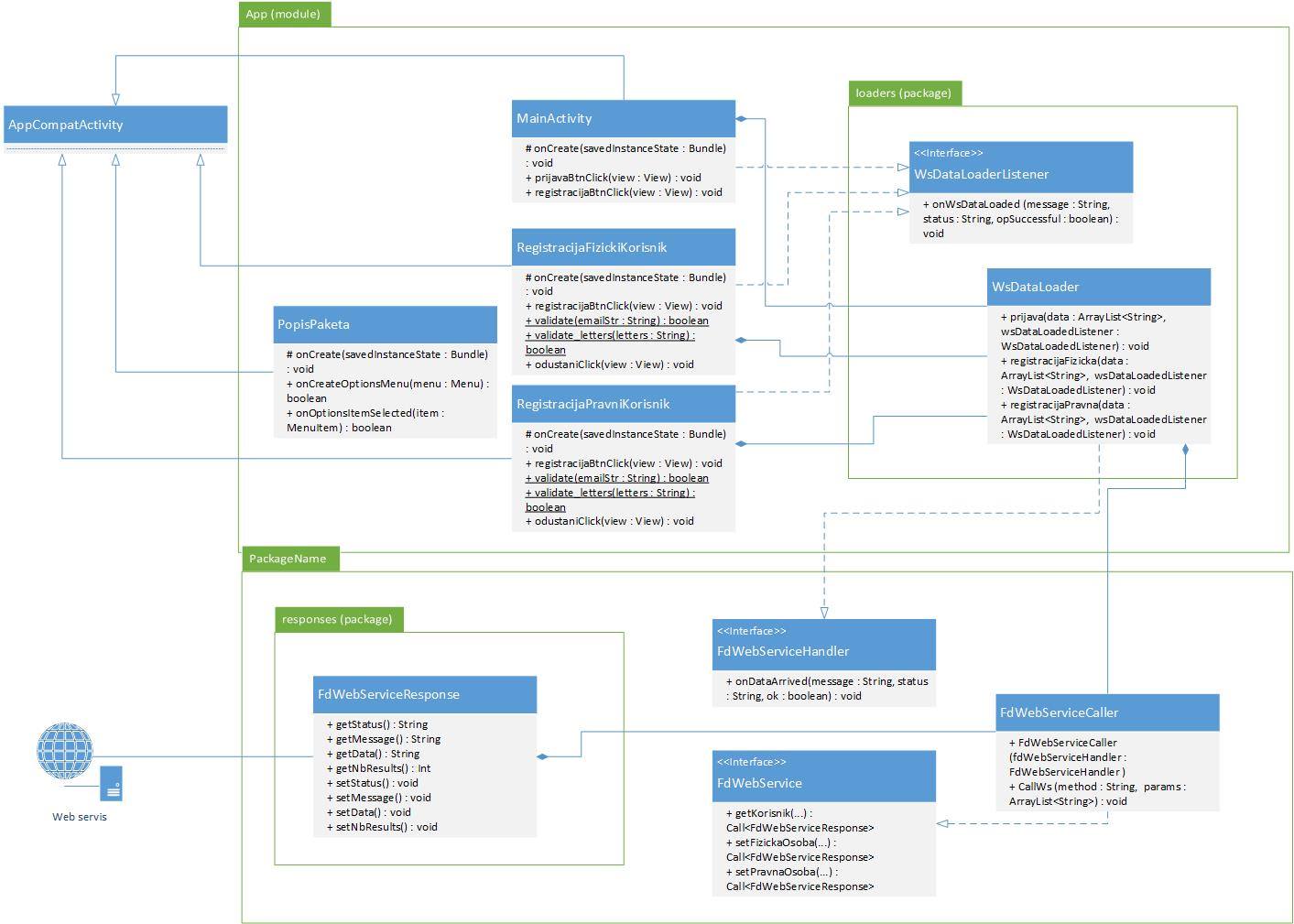
|  |  |  |
| --- | --- | --- |
| RESPONSE | Nema vrsta i hrane u bazi | Dohvaćeni su vrste i jedinice |
| status | OK | OK |
| nbResults | 0 | 0 |
| Message | „Nema zapisa u bazi!“ | „Pronađeni su rezultati“ |
| data | Atributi elemenata u vrsta i jedinice su id = -1 i  naziv =““ | Niz svih gradova sa vrijednostima vrsta i jedinica u JSON formatu. Vrsta i jedinice imaju atribute id i naziv |

# Struktura aplikacije

Aplikacija koristi dva modula - app i webservice, za svoj rad. App i webservice moduli povezani su klasama paketa loaders koji se nalazi unutar modula app. Klase koje su u modulu app ali nisu dio paketa loaders su android aktivnosti (zasebni zasloni sa svojim funkcionalnostima):

* MainActivity : Aktivnost u kojoj je implementirana prijava na aplikaciju, te poveznica (preko pomoćnog prozora s pitanjem) na aktivnosti koje implementiraju registraciju.
* RegistracijaFizickiKorisnik : Aktivnost u kojoj je implementirana registracija fizičkog korisnika. Uspješnom registracijom, ili odabirom na odustani, korisnik se vraća na MainActivity aktivnost.
* RegistracijaPravniKorisnik : Aktivnost u kojoj je implementirana registracija fizičkog korisnika. Uspješnom registracijom, ili odabirom na odustani, korisnik se vraća na MainActivity aktivnost.
* PopisPaketa : Aktivnost u kojoj je trenutno implementirana traka izbornika sa gumbom „Odjava“ (vraćanje na MainActivity). U ovu aktivnost korisnik stupa tek nakon uspješne prijave.

Svaka klasa od gore navedenih aktivnosti implementira WsDataLoaderListener sučelje, te stvara WsDataLoader objekt, koji stvara objekt klase FdWebServiceCaller iz modula webservice, te implementira sučelje FdWebServiceHandler. Stvaranje FdWebServiceCaller objekta, omogućit će slanje upisanih podataka „sa zaslona“ uz odgovarajuću metodu koja će definirati API poziv prema web servisu. Upit API-ju definiran je u sučelju FdWebService, dok je odgovor API-ja smješten u FdWebServiceResponse. Nakon što klasa FdWebServiceCaller pročita odgovor u FdWebServiceResponse, odmah poziva metodu onDataArrived (metoda iz sučelja FdWebServiceHandler) sa podacima kao parametrima metode. Unutar implementacije te metode (implementira ju WsDataLoader klasa) nalazi se poziv metode onWsDataLoaded (koju implementiraju sve aktivnosti osim PopisPaketa). Unutar te metode definirana je povratna poruka za korisnika aplikacije – (ne)uspješna prijava ili (ne)uspješna registracija. Kako bi prijava ili registracija bile uspješne, korisnik mora upisati podatke koje zadovoljavaju ispravno upisan podatak/polje. Taj dio se provjerava unutar slušatelja događaja OnClick koji se nalazi u svim trenutno postojećim aktivnostima osim PopisPaketa.



Slika : Dijagram klasa implementiranog koda aplikacije