

**SVEUČILIŠTE U ZAGREBU**  
**FAKULTET ORGANIZACIJE I INFORMATIKE**  
**V A R A Ž D I N**

**Goran Drmenčić**

**Danijel Filipović**

**Matija Jurman**

**Danijel Sokač**

**ChatUP aplikacija**  
**TEHNIČKA DOKUMENTACIJA**

Github repozitorij:

- <https://github.com/DanijelFilipovic/Chat-aplikacija>
- <https://github.com/DanijelFilipovic/ChatUp-Dodatno> (Web servisi)

**Varaždin, 2015.**

**SVEUČILIŠTE U ZAGREBU**  
**FAKULTET ORGANIZACIJE I INFORMATIKE**  
**V A R A Ž D I N**

**Oznaka tima: T18**

**Članovi tima:**

- Goran Drmenčić, 0246018410
- Danijel Filipović, 0016090066
- Matija Jurman, 0016084901
- Danijel Sokač, 0016090066

**ChatUP aplikacija**  
**TEHNIČKA DOKUMENTACIJA**

Github repozitirij:

- <https://github.com/DanijelFilipovic/Chat-aplikacija>
- <https://github.com/DanijelFilipovic/ChatUp-Dodatno> (Web servisi)

**Mentor:**

Ivan Švogor, mag. inf.

**Varaždin, studeni 2015.**

# Sadržaj

|   |   |
|---|---|
| 1. Uvod .....                                 | 1 |
| 2. Use-Case Dijagram.....                     | 2 |
| 3. Konfiguracija servera i baze podataka..... | 4 |
| 4. Web servisi .....                          | 8 |
| 4.1. server.js.....                           | 8 |
| 4.2. register.js .....                        | 8 |
| 4.3. log_in.js .....                          | 8 |
| 4.4. log_out.js .....                         | 8 |
| 5. Dijagram klasa.....                        | 9 |

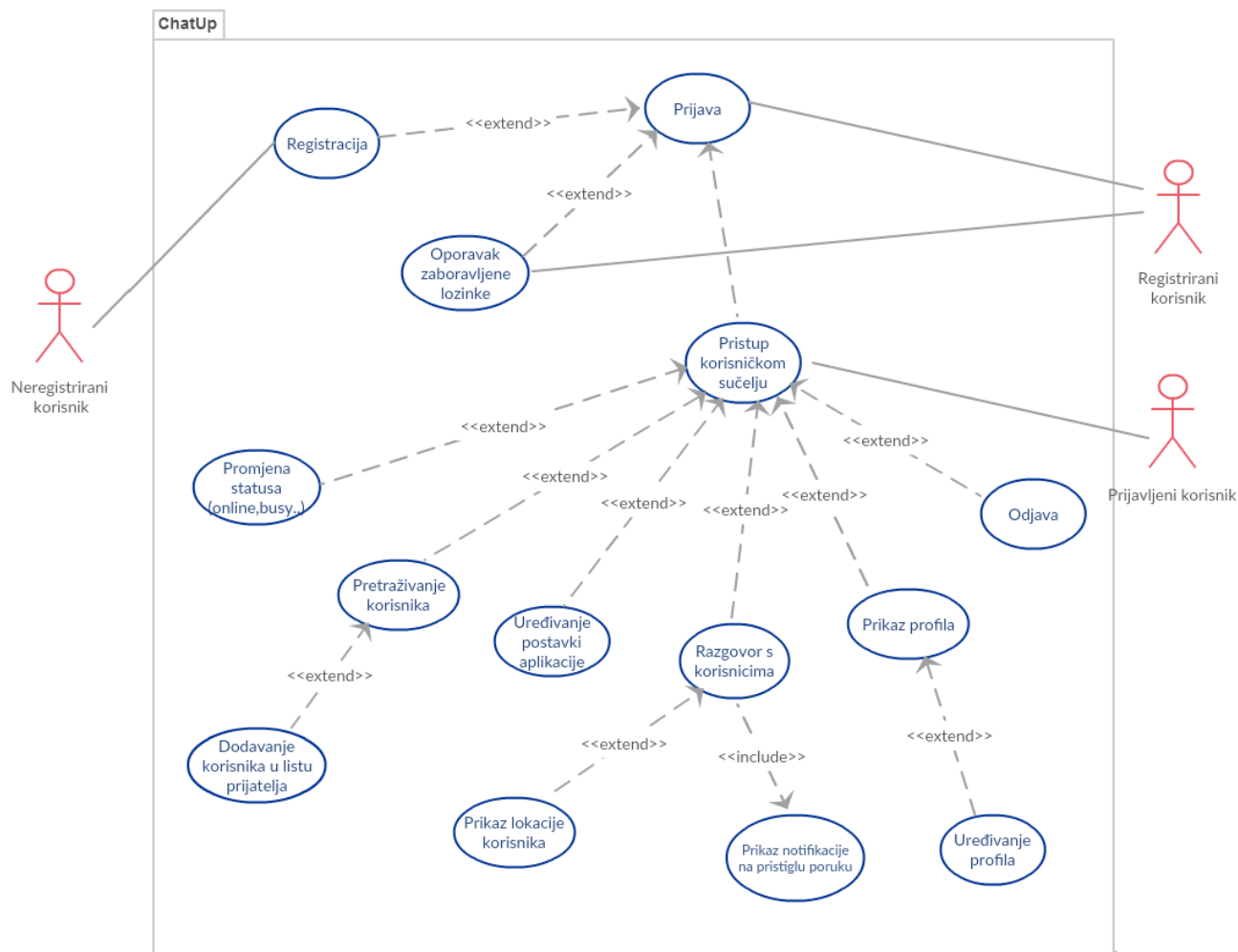
# 1. Uvod

Ovo je tehnička dokumentacija za izradu android aplikacije ChatUp kao projekt za kolegij Analiza i razvoj programa. U ovom dokumentu se mogu pronaći svi tehnički artefakti koji su bili potrebni za nastanak aplikacije.

Pod tehničke artefakte ubrajamo: izvorni kod aplikacije (ukoliko bude potrebno objasniti neke detalje), dijagrami (koji objašnjavaju aplikaciju) i tehnologija koja se koristila pri izradi aplikacije.

## 2. Use-Case Diagram

Slučajevi korištenja aplikacije s tehničke razine biti će prikazani Use-Case diagramom.



*Slika 1: Use-Case dijagram ChatUp aplikacije*

Iz tog je dijagrama vidljivo sljedeće. Imamo tri tipa korisnika aplikacije: neregistrirani, registrirani i prijavljeni korisnik. Neregistrirani korisnik može se samo registrirati. Registrirani korisnik može se prijaviti ili zatražiti oporavak izgubljene lozinke. Treći tip korisnika je prijavljeni korisnik. Riječ je o korisniku koji to postaje nako uspješne prijave u aplikaciju. Prijavljeni korisnik automatski pristupa korisničkom sučelju, a opcionalno, može izvoditi sljedeće akcije:

- Odjaviti se iz aplikacije
- Prikazati i urediti svoj profil

- Promjeniti vlastiti status
- Pretraživati druge korisnike aplikacije
- Dodati drugog prijatelja u svoju listu prijatelja, što uključuje dobivanje notifikacije prilikom dobivanja poruke od drugog korisnika
- Urediti vlastite postavke aplikacije
- Razgovarati s drugim korisnicima, što uključuje dobivanje notifikacije o primljenoj poruci kada mu drugi korisnik pošalje poruku
- Prikazati vlastitu lokaciju, ili vidjeti lokaciju od drugog prijatelja koji šalje poruku [opcionalno]

### 3. Konfiguracija servera i baze podataka

Za bazu podataka odlučili smo koristiti CouchDB<sup>1</sup>. CouchDB baza podataka sprema podatke u obliku JSON-a (engl. JavaScript Object Notation) i spada u NoSQL baze podataka. Pristup podacima i upiti mogu se odvijati preko web preglednika putem HTTP protokola. Indeksiranje, spajanje i transformiranje dokumenata vrši se s JavaScript programskim jezikom. CouchDB odlično radi s modernim web i mobilnim aplikacijama. Također podržava inkrementalnu replikaciju za distribuciju podataka ili aplikaciju putem mreže.

CouchDB dokument je JSON objekt koji se sastoji od imenovanih polja. Vrijednosti polja mogu biti tekstualni, numerički, datumski pa čak i posložena lista ili mapa. Za rješavanje problema dodavanja strukture natrag u polustrukturirane podatke, CouchDB integrira model pogleda koristeći JavaScript za opis. Pogledi (engl. Views) su načini agregiranja i izvještavanja o dokumentima unutar baze.

Da bi baza bila dostupna svima, potrebno je CouchDB smjestiti na server. U ovom projektu koristio se jedan od servera na Digital Ocean<sup>2</sup> stranici. Nakon izrade droplet-a dobije se javna IP adresa preko koje se može dohvaćati baza.

Slika 1 prikazuje korisničko sučelje na stranici DigitalOcean-a.

| Img   | Name      | IP Address    | Memory | Disk  | Region |
|---|-----------|---------------|--------|-------|--------|
|  | ubuntuNew | 104.236.58.50 | 512 MB | 20 GB | NYC3   |

Slika 1 Podaci droplet-a korištenog na projektu

Nakon instalacije baze potrebno je kreirati bazu i njezine dokumente. Od dokumenata u projektu koristit će se dvije vrste dokumenata za pohranu podataka. Prvi dokument sadrži podatke o pojedinom korisniku dok drugi dokument bi sadržavao pohranu razgovora između dva ili više korisnika.

<sup>1</sup> Apache CouchDB - <http://couchdb.apache.org/>

<sup>2</sup> DigitalOcean, Simple Cloud Hosting - <https://www.digitalocean.com/>

Izgled dokumenta koji sadrži podatke o korisniku u prvoj fazi izrade projekta izgleda ovako:

```
{
  "_id": "mirko@mail.hr",
  "_rev": "2-d765a93614eeeca829c0c859f3c47482",
  "mail": "mirko@mail.hr",
  "username": "mirko",
  "password": "mirko123",
  "status": "offline",
  "gender": "m",
  "dateOfBirth": "2011-02-02",
  "friends": "darko",
  "type": "user"
}
```

Također smo za početak odredili da bi dokument koji bi spremao razgovor trebao izgledati ovako:

```
{
  "_id": "e99c446798df424685560eb5fb0013cb",
  "_rev": "1-efd67a6538a03aa677a32b06bad2e5ee",
  "participants": [
    {
      "mail": "darko@mail.hr"
    },
    {
      "mail": "mirko@mail.hr"
    }
  ],
  "chat": [
    {
      "sender": "darko",
      "text": "Pozdrav!",
      "timeSend": "1235482156",
      "location": ""
    },
    {
      "sender": "mirko",
      "text": "Bok. Kako je",
      "timeSend": "1235482158",
      "location": ""
    }
  ]
}
```

Vjerujemo da će se struktura dokumenata s vremenom nešto promijeniti zbog dodavanja novih mogućnosti za aplikaciju. Također baza sadrži i nekoliko pogleda koje koriste web servisi.

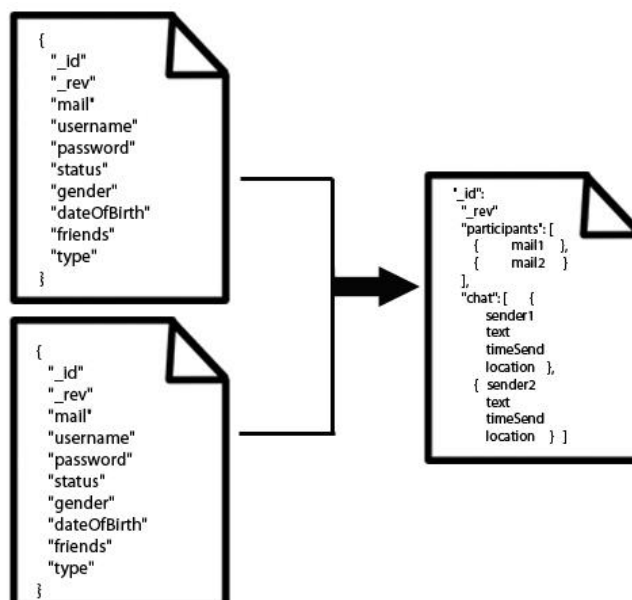


Da bi CouchDB znao da je dokument ustvari pogled na bazu, njegov *\_id* mora započeti sa *\_design/*, u nastavku je prikazan programski kôd našeg dokumenta za poglede:

```
{
  "_id": "_design/view",
  "_rev": "3-3d9caecc3eb8f1ace2cfca7fa165aa5e",
  "views": {
    "getUserIDs": {
      "map": "function(doc) { if (doc.type && doc.type == 'user') emit(doc._id); }"
    },
    "getMailAndPasswords": {
      "map": "function(doc) { if(doc.mail && doc.password) { emit(doc._id, { Mail: doc.mail, Lozinka: doc.password}); } }"
    },
    "getOnlineUsers": {
      "map": "function(doc) { if (doc.type && doc.type == 'user' && doc.status == 'online') emit(doc); }"
    },
    "getOfflineUsers": {
      "map": "function(doc) { if (doc.type && doc.type == 'user' && doc.status == 'offline') emit(doc); }"
    }
  }
}
```

Vidimo da trenutno postoje četiri pogleda (getUserIDs, getMailAndPasswords, getOnlineUsers i getOfflineUsers). Kako će se aplikacija razvijati povećavati će se i broj samih pogleda.

Slika 2 grafički prikazuje sadržaj glavnih dokumenata i njihovu vezu.



Slika 2 Grafički prikaz dokumenata i njihovih veza

Dva lijeva dokumenta predstavljaju pojedinog registriranog korisnika. Kada započne razgovor kreira se poseban dokument za njihov razgovor u koji se spremaju sve poruke poslane između tih korisnika. Razgovor može sadržavati dva ili više sudionika.

Da bi baza i aplikacija na mobitelu mogli međusobno komunicirati potrebno je izraditi web servise. Oni će detaljno biti razrađeni u sljedećem poglavlju.

## 4. Web servisi

Web servisi služe kao posrednici između baze podataka i android aplikacije. Web servisi su realizirani pomoću Node.js tehnologije za izradu *backend* Web aplikacija.

### 4.1. server.js

Ovaj Web servis je glavni Web servis koji će se vrtiti na poslužitelju. Svi ostali Web servisi su zapisani u zasebnim JavaScript datotekama kao funkcije. U ovoj datoteci se te funkcije uključuju i pridružuju se odgovarajućoj ruti preko koje se određeni Web servis može zatražiti.

### 4.2. register.js

Web servis za registraciju novog korisnika. Prima POST parametre koji sadrže sljedeće informacije o korisniku: korisničko ime, e-mail adresu, spol, datum rođenja i lozinka. Web servis će zatražiti pogled iz CouchDB baze podataka koji će izlistati popis svih registriranih korisnika. Ako ne postoji korisnik sa istim korisničkim imenom koji je poslan Web servisu, tada se novi korisnik zapisuje u bazu podataka.

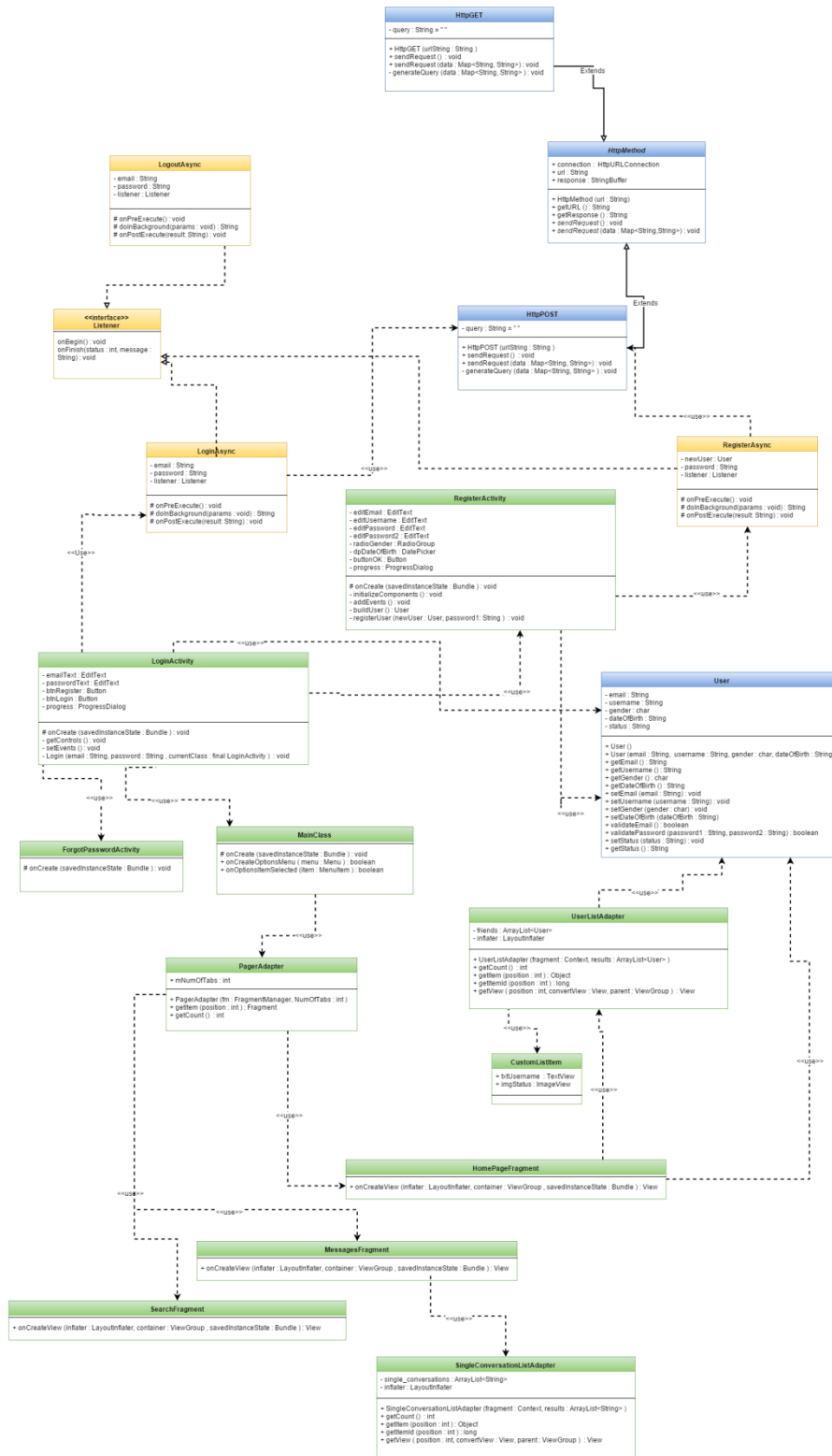
### 4.3. log\_in.js

Web servis za prijavu korisnika služi za prijavljivanje korisnika na aplikaciju. Parametri koje prima su e-mail korisnika te njegova lozinka. Nakon toga se pokreće pogled iz baze koji vraća sve neprijavljene korisnike i provjerava odgovaraju li parametri i je li korisnik već prijavljen ili nije. Ako je sve uredu onda promijeni status korisnika na „online“ i tako je korisnik prijavljen.

### 4.4. log\_out.js

Web servis za odjavu korisnika vrlo je jednostavan. U njemu se provjera da li je u bazi korisnik koji se želi odjaviti zapravo online, te ako je, web servis mijenja polje *status* u offline.

## 5. Dijagram klasa



Plava boja predstavlja modul *core*, zelena boja predstavlja modul *app* , a žuta boja predstavlja modul *ws*.