

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

**Tim AIR1813**

**WeBeChattin**

**TEHNIČKA DOKUMENTACIJA**

**Varaždin, 2018.**

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

**Ana Guštek**

**Andrea Kljaić**

**Domagoj Milić**

**Kolegij: Analiza i razvoj programa**

**WeBeChattin**

**TEHNIČKA DOKUMENTACIJA**

GitHub: <https://github.com/dmilic-foi/WeBeChattin>

Trello: <https://trello.com/air1813/home>

**Mentor/Mentorica:**

Doc. dr. sc. Zlatko Stapić

**Varaždin, studeni 2018.**

# Sadržaj

Sadržaj .....	iii
1. Uvod .....	1
2. Funkcionalnosti aplikacije WeBeChattin .....	2
2.1. Wireframe aplikacije .....	2
3. Korišteni alati i tehnologije .....	5
4. Struktura aplikacije.....	6
4.1. ERA model .....	6
4.2. Moduli .....	7
4.3. Komunikacija klijenta i servera.....	8

# 1. Uvod

Tehnička dokumentacija opisuje tijek izrade mobilne aplikacije i način rada na projektu. Uvidom u dokumentaciju dobivaju se informacije o funkcionalnosti aplikacije WeBeChattin i u strukturu aplikacije.

Aplikacija WeBeChattin je mobilna chat aplikacija koja se izrađuje u suradnji s poduzećem CS Computer Systems d.o.o. U tijeku izrade aplikacije potrebno je proučiti open source protokole sa end to end enkripcijom. Aplikacija za popunjavanje baze korisnika koristi telefonski imenik, razgovori su grupirani prema korisniku i moguće je vidjeti povijest razgovora.

## 2. Funkcionalnosti aplikacije WeBeChattin

Aplikacija sadrži sljedeće funkcionalnosti:

- Prijava/registracija korisnika
- Slanje sigurnosnog koda za registraciju
- Popunjavanje baze iz telefonskog imenika
- Slanje i primanje poruka
- Grupiranje poruka
- Uređivanje profila korisnika

Korisnik se obavezno mora prijaviti, odnosno registrirati kako bi koristio aplikaciju, jer bez prijave nije moguće popuniti bazu kontaktima iz telefonskog imenika, a zatim ni korištenje ostalih funkcionalnosti aplikacije.

### 2.1. Wireframe aplikacije

Wireframe aplikacije koji je prikazan u nastavku nije pretjerano kompliciran jer sama bit chat aplikacije je slanje i primanje poruka.

Kada korisnik prvi put otvori aplikaciju prikaže mu se početni ekran na kojem ima mogućnost prijave, odnosno registracije u aplikaciju. Kako bi se korisnik prijavio najprije mora unijeti svoj broj mobitela, na koji mu se potom automatski šalje verifikacijski kod. Kada korisnik dobije verifikacijski kod, mora ga unijeti u aplikaciju, na odgovarajuće mjesto. Postoji mogućnost da korisnik ne unese ispravan kod ili da ga uopće ne dobije iz različitih razloga. U tom slučaju aplikacija korisnika vraća na početni ekran, odnosno ekran u kojem unosi broj mobitela na koji se ponovno šalje verifikacijski kod.

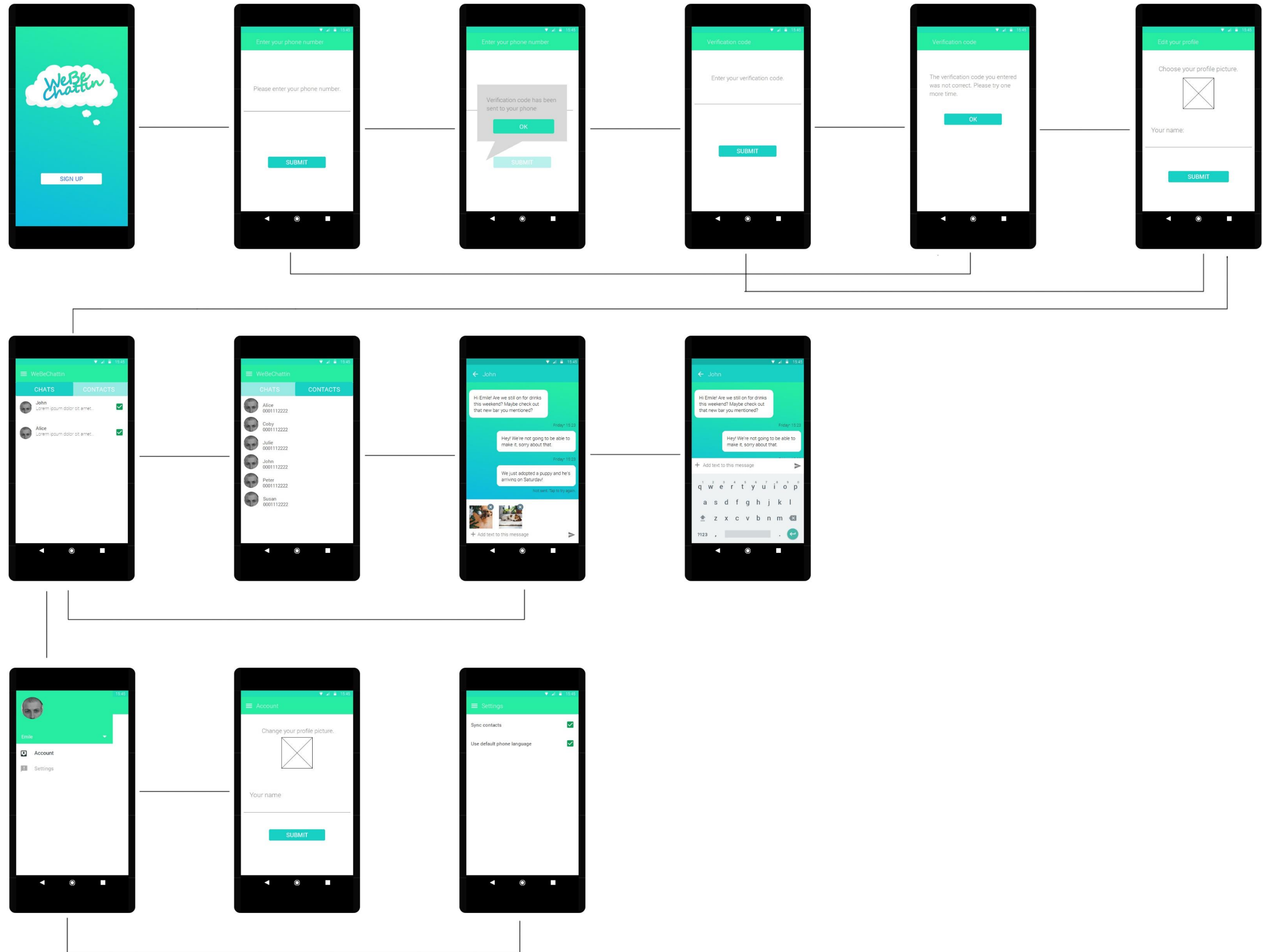
Nakon uspješne prijave i unosa verifikacijskog koda korisnik upisuje svoje ime i može odabrati svoju profilnu sliku. Nakon toga se otvara glavni ekran aplikacije. Ovdje korisnik može pregledati svoje razgovore, može pregledati kontakte koji se uvoze automatski iz telefonskog imenika korisnika prema unesenom broju mobitela.

Klikom na pojedini razgovor otvara se ekran u kojem je moguće pregledati povijest razgovora i u kojem se može napisati i poslati poruka. U oblačićima koji pozicionirani lijevo nalaze se poruke koje su primljene, dok su desno pozicionirane poruke koje šalje korisnik.

Na glavnom ekranu aplikacije moguće je doći do glavnog izbornika aplikacije u kojem korisnik može urediti svoj profil, odnosno može promijeniti sliku, te može promijeniti svoje ime unutar aplikacije.

Unutar izbornika postoji još nekoliko regularnih postavki kojima korisnik može upravljati prema izboru.

U nastavku se nalazi prikaz skica ekrana povezanih u wireframe. Točnije oni ekrani koji su na neki način povezani jedan sa drugim su u ovom prikazu spojeni crtom.



### 3. Korišteni alati i tehnologije

Za izradu i testiranje aplikacije i samog projekta potreban je određen hardware i software. Pri izradi aplikacije korištena su prijenosna računala sa Windows 10 operacijskim sustavom. Aplikacije se testira na pametnim telefonima koji koriste Android sustav. Pametni telefoni imaju različite verzije Android sustava, što omogućuje bolje testiranje aplikacije jer se može vidjeti kako se aplikacija ponaša na kojoj verziji.

Naziv resursa	Količina resursa
Prijenosno računalo (Windows OS)	3
Pametni telefon (Android OS)	3

Tablica 1. Korišteni resursi

Tehnologija	Alat
IDE Android / SDK Tools	Android Studio
Izrada dijagrama	draw.io
Praćenje projekta	Trello
Dokumentacija projekta	Microsoft Office
Verzioriranje koda	GitHub

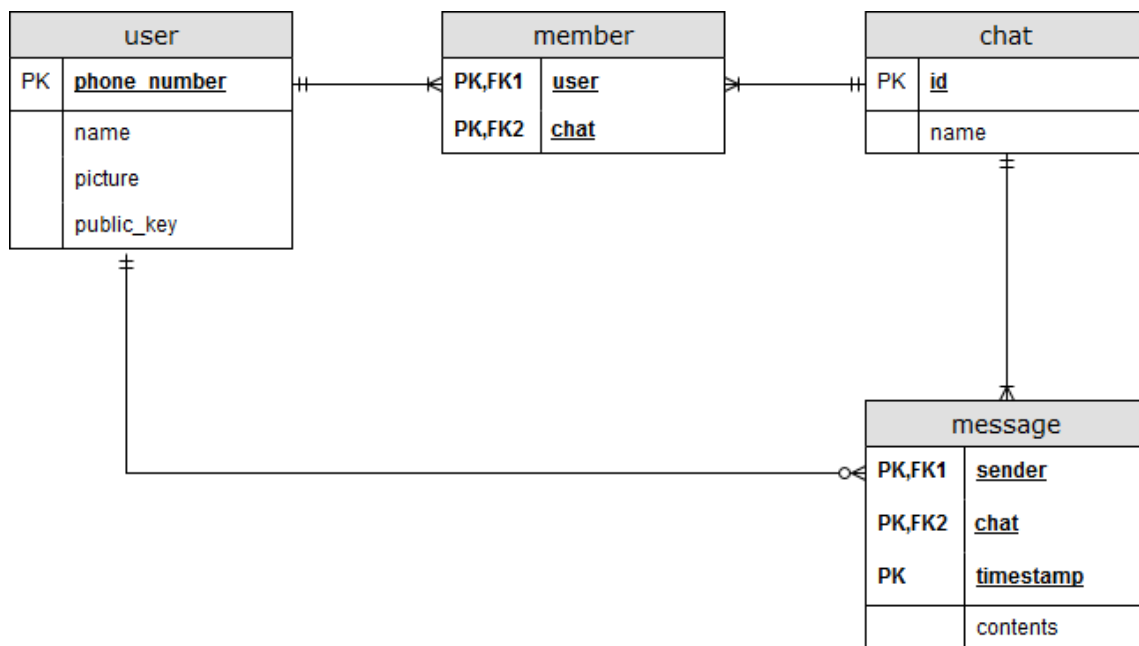
Tablica 2. Korištene tehnologije



## 4. Struktura aplikacije

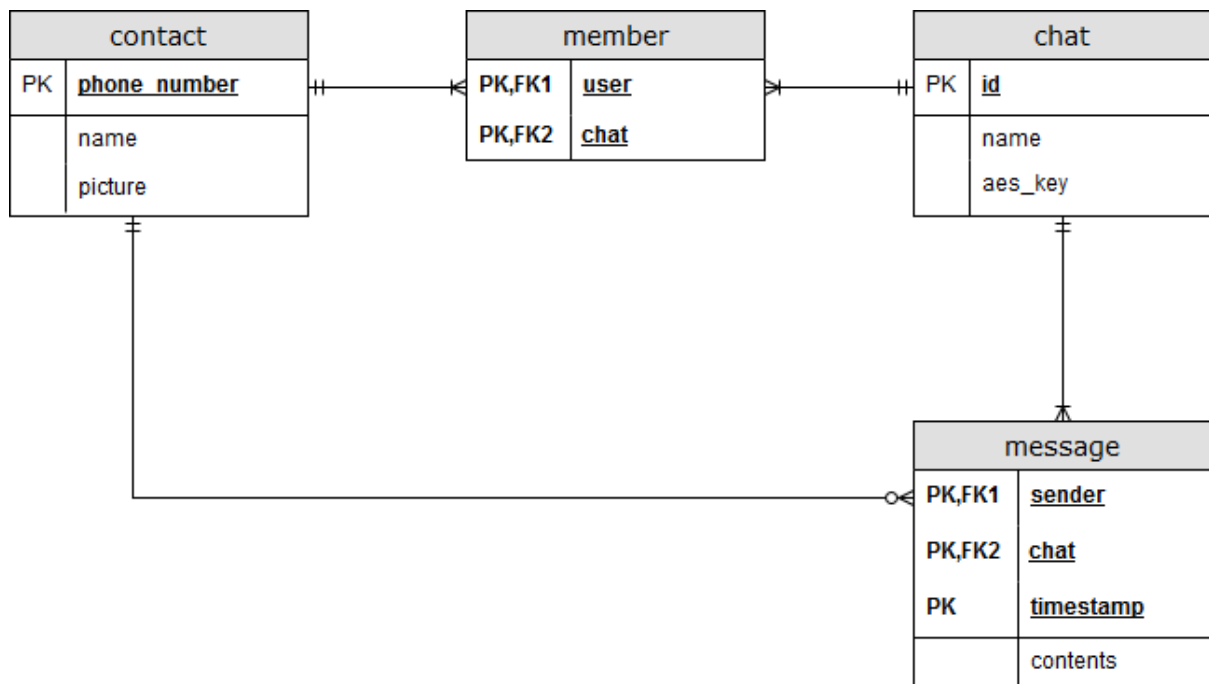
### 4.1. ERA model

S obzirom da se u aplikaciji šalju i primaju poruke, potreban nam je server i klijent, te stoga imamo dva ERA dijagrama, jedan za klijenta i drugi za server.



Slika 1. ERA model – server

Na ERA dijagramu koji se odnosi na server postoje 4 tablice. To su: user, member, chat i message. Korisnik (user) sadrži primarni ključ koji je zapravo broj mobitela, a osim toga sadrži ime, sliku i javni ključ. Korisnik je povezan s tablicama član (member) i poruka (message). Tablica član povezuje tablice korisnik i chat, odnosno definira koji je korisnik član kojeg razgovora. Tablica poruka sadrži informacije o tome tko je poslao poruku, u kojem razgovoru, u koje vrijeme i koji je zapravo sadržaj te poruke.

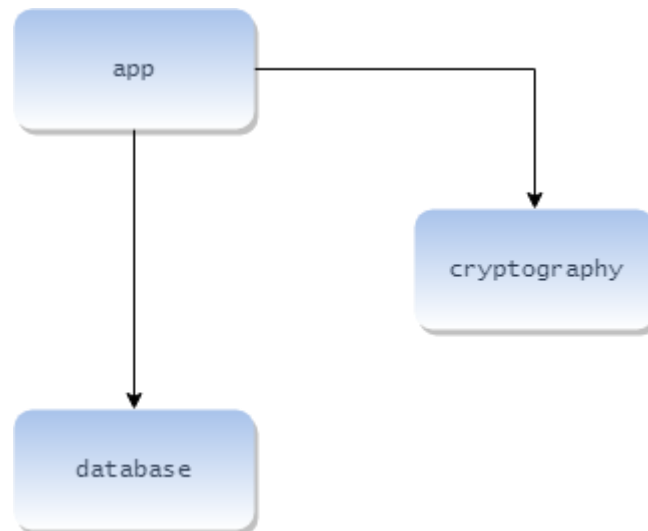


Slika 2. ERA model – klijent

ERA model za klijenta se ne razlikuje pretjerano od prošlog dijagrama. Ovdje se umjesto tablice user pojavljuje tablica contact koja sadrži također telefonski broj kao primarni ključ, te ime i sliku. Tablica chat se također nešto razlikuje od chat tablice servera jer ova sadrži i AES ključ koji se odnosi na enkripciju.

## 4.2. Moduli

Za izradu aplikacije koriste se tri modula koja međusobno komuniciraju. Glavni modul je app modul i on sadrži glavne funkcionalnosti aplikacije. Database modul služi za komuniciranje s bazom podataka i za dohvaćanje podataka. Treći modul je cryptography koji sadrži izdvojenu funkcionalnost kriptiranja poruka za slanje.



Slika 3. Moduli aplikacije

### 4.3. Komunikacija klijenta i servera

Kako bi se u aplikaciji poruke mogle slati i primiti, potrebna je komunikacija između klijenta i servera. Tijekom slanja poruke server najprije otvara TCP socket koji radi na portu 51345 i na tom portu sluša nadolazeće veze od klijenta. Nakon što prihvati vezu, server prima poruke od klijenta i šalje im odgovor. Primjerice, kada se klijent poveže na server, tada mu server pošalje odgovor da je sve u redu i server zatim čeka poruku od klijenta. Klijent šalje poruku u JSON formatu nakon čega server tu poruku dekodira u polje i sastavlja SQL upit za unos, što i šalje klijentu kao string.

S druge strane, klijent također kreira TCP socket i povezuje se na server i na port 51345 i kao što je već spomenuto, nakon što od servera dobije odgovor da je sve u redu može poslati poruku.