



UNIVERZITET U SARAJEVU  
ELEKTROTEHNIČKI FAKULTET SARAJEVO



## **Tema: Društvena aplikacija – Friend**

-završni rad-

Mentor:

Doc. dr. Vensada Okanović, dipl.ing.el.

Kandidat:

Palalić Semin

Sarajevo, 2017

Univerzitet u Sarajevu  
Naziv fakulteta/akademije \_\_\_\_\_  
Naziv odsjeka i/ili katedre \_\_\_\_\_  
Predmet \_\_\_\_\_

### Izjava o autentičnosti radova

Seminarski rad, završni (diplomski odnosno magistarski) rad za I i II ciklus studija i integrirani studijski program I i II ciklusa studija, magistarski znanstveni rad i doktorska disertacija<sup>1</sup>

Ime i prezime \_\_\_\_\_  
Naslov rada \_\_\_\_\_  
Vrsta rada \_\_\_\_\_  
Broj stranica \_\_\_\_\_

#### Potvrđujem:

- da sam pročitao/la dokumente koji se odnose na plagijarizam, kako je to definirano Statutom Univerziteta u Sarajevu, Etičkim kodeksom Univerziteta u Sarajevu i pravilima studiranja koja se odnose na I i II ciklus studija, integrirani studijski program I i II ciklusa i III ciklus studija na Univerzitetu u Sarajevu, kao i uputama o plagijarizmu navedenim na web stranici Univerziteta u Sarajevu;
- da sam svjestan/na univerzitetskih disciplinskih pravila koja se tiču plagijarizma;
- da je rad koji predajem potpuno moj, samostalni rad, osim u dijelovima gdje je to naznačeno;
- da rad nije predat, u cjelini ili djelimično, za stjecanje zvanja na Univerzitetu u Sarajevu ili nekoj drugoj visokoškolskoj ustanovi;
- da sam jasno naznačio/la prisustvo citiranog ili parafraziranog materijala i da sam se referirao/la na sve izvore;
- da sam dosljedno naveo/la korištene i citirane izvore ili bibliografiju po nekom od preporučenih stilova citiranja, sa navođenjem potpune reference koja obuhvata potpuni bibliografski opis korištenog i citiranog izvora;
- da sam odgovarajuće naznačio/la svaku pomoć koju sam dobio/la pored pomoći mentora/ice i akademskih tutora/ica.

Mjesto, datum \_\_\_\_\_  
Potpis \_\_\_\_\_

<sup>1</sup> U radu su korišteni slijedeći dokumenti: *Izjava autora* koju koristi Elektrotehnički fakultet u Sarajevu; *Izjava o autentičnosti završnog rada* Centra za interdisciplinarnu studiju – master studij „Evropske studije”; *Izjava o plagijarizmu* koju koristi Fakultet političkih nauka u Sarajevu

## Sažetak

U ovom radu je opisan cjelokupni razvoj društvene aplikacije za mobilne uređaje. Detaljno su opisani softveri koji su omogućili razvoj Friend društvene aplikacije. Prezentirani su folderi od kojih se aplikacija sastoji i struktura baze podataka koju aplikacija koristi. Sav aplikacijski interfejs prikazan je pomoću slika, a rad sadrži i prikaz koda koji implementira određene funkcionalnosti aplikacije.

## Abstract

In this paper it will be described the overall social application development for mobile devices. The software that enabled the development of Friend social application has been described in detail. The folders from which the application is made are presented, and the database structure that the application uses is also presented. All application interfaces are displayed with images, and the paper also contains a code view that implements certain application functionality.

## Sadržaj

1. Uvod .....	6
2. Ciljevi aplikacije.....	8
3. Razvojno okruženje .....	9
3.1. Java Development Kit .....	9
3.2. Android SDK.....	9
3.3. Java Runtime Environment .....	9
3.4. Android studio.....	10
4. Sadržaj aplikacije.....	11
4.1 Manifest folder .....	11
4.2 Java folder .....	11
4.3 Res folder .....	12
4.3.1 Drawable folder .....	12
4.3.2. Layout folder .....	13
4.3.3. Menu folder .....	13
4.3.4. Mipmap folder .....	14
4.3.5. Values folder.....	14
5. Baza podataka (Firebase).....	16
5.1. Firebase autentifikacija .....	16
5.2. Firebase baza podataka u realnom vremenu .....	16
5.3. Firebase cloud pohrana.....	19
6. Interfejsi i funkcionalnosti aplikacije .....	20
6.1. Početna stranica aplikacije .....	20
6.1.1. Prijava.....	20
6.1.2. Registracija .....	21
6.2. Korisnička stranica.....	24
6.3. Glavna stranica.....	27
6.3.1. Izbor tab stranice .....	29
6.3.2. Objavljivanje statusa.....	32
6.3.3. Dostupni korisnici.....	33
6.3.4. Pretraga.....	35

6.3.5. Izbor dodatnih opcija .....	35
7. Zaključak .....	36
8. Reference .....	37

## 1. Uvod

Trenutno najbolji način povezivanje ljudi širom planete Zemlje predstavljaju društvene mreže. Uz pomoć društvenih mreža i postojeće tehnologije svako mjesto na planeti je lako i brzo dostupno. One su sastavni dio ljudskog života i svakodnevno se koriste za sklapanje novih prijateljstava, za igradnju poslovnih veza ili jednostavno za informiranje ljudi.

O ogromnoj zastupljenosti društveni mreža govore i statistički podaci. Najpopularnija društvena mreža Facebook ima oko 2 milijarde mjesečno aktivnih korisnika, što predstavlja 26% ukupne svjetske populacije. Društvene mreže WhatsApp i Messenger imaju oko 1,2 milijarde mjesečno aktivni korisnika, što predstavlja 16% ukupne svjetske populacije.

U nastavku ćemo se upoznati sa razvojem Friend društvene aplikacije za mobilne uređaje. Detaljno će biti opisano šta je sve potrebno za razvoj društvene aplikacije. Govorit će se o razvojnom okruženju u kojem se razvijala aplikacija. Površno će se proći kroz klase i XML fajlove od kojih se sastoji aplikacija. Govorit će se o bazi podataka i njenoj strukturi. Kroz rad će biti prikazana implementacija pojedinih funkcionalnosti aplikacije. Također, biće prikazan sav interfejs Friend aplikacije.

U prvom poglavlju biće opisano razvojno okruženje u kojem se razvijala aplikacija. Nakon govora o softverskom razvojnom okruženju koje omogućava razvoj aplikacija, govorit će se o kolekciji alata koje omogućavaju izgradnju, testiranje i debugiranje aplikacija. Također, biće govora o kolekciji alata bez kojih se aplikacija ne bi mogla izvršavati. Na kraju poglavlja spomenuto je integrisano razvojno okruženje u kojem se razvijala aplikacija.

U drugom poglavlju biće opisana tri foldera koja su sastavni dio svakog projekta u android razvojnom okruženju. Govorit će se o klasama koje čine backend dio aplikacije i folderima u kojima se definiše izgled aplikacije. Svaki folder i podfolder će biti detaljno opisani, a sadržaj foldera prikazan uz pomoć slika.

U trećem poglavlju govorit će se o bazi podataka koju aplikacija koristi. Nakon kratkog historijskog uvoda u Firebase i njegovog opisa, slijedi opis tri usluge koje nudi Firebase, a koristi ih Friend aplikacija. Ukratko će biti opisana usluga autentifikacije korisnika. Govorit će se i o bazi podataka u realnom vremenu i njenim osobinama. Struktura baze podataka će biti detaljno prikazana i opisana. Na kraju poglavlja govorit će se o pohrani različitih vrsta fajlova.

U četvrtom poglavlju govorit će se o interfejsu aplikacije i njenim funkcionalnostima. Na početku poglavlja prikazan je interfejs početne stranice aplikacije. Detaljno će biti opisana prijava i registracija korisnika na aplikaciju, a uz opis prikazan je i kod koji implementira ove dvije funkcionalnosti. Nakon toga, slijedi prikaz izgleda korisničke stranice i stranice za dopisivanje. Uz prikaz izgleda stranice za dopisivanje nalazi se i kod koji omogućava razmjenu poruka između korisnika aplikacije, kao i kod za ispis poruka. U nastavku poglavlja slijedi prikaz izgleda glavne stranice aplikacije. Kroz kod će biti opisana implementacija

dvije opcije sa menija. Nakog toga, slijedi opis i prikaz News, Chat i Friends stranica. Detaljno će se opisati stranica za objavljivanje statusa i stranica za pregled dostupnih korisnika. Na kraju poglavlja govorit će se o pretrazi i dodatnim opcijama.

## 2. Ciljevi aplikacije

Prvenstveni cilj ove aplikacije je pokazati šta je sve potrebno za razvoj jedne društvene aplikacije za mobilne uređaje. Kada se gledaju funkcionalnosti aplikacije, ona ima sljedeće ciljeve:

- Povezati ljude sa zajedničkim interesima
- Olakšati komunikaciju
- Omogućiti brz dolazak informacija
- Omogućiti slobodu izražavanja
- Pomoći u edukaciji
- Pomoći u promovisanju poslova i različitih proizvoda
- Pomoći u informisanju ljudi



### 3. Razvojno okruženje

Softver potreban za programiranje i razvoj Friend aplikacije je:

- Java Development Kit
- Android SDK
- Java Runtime Environment
- Android studio

#### 3.1. Java Development Kit

Java Development Kit (JDK) je softversko razvojno okruženje koje omogućava razvoj Java programa. JDK se sastoji od Java Runtime Environmenta, interpretera, kompajlera, arhivera, generatora dokumenata i drugih alata potrebnih za Java programiranje. Napisani kod JDK prvo pretvara u bajt kod. Pretvaranje u bajt kod vrši kompajler. Nakon toga, interpreter preuzima bajt kod i parsira ga u mašinski kod. Cilj cijelog ovog procesa je omogućavanje hardveru da razumije napisani izvorni kod. Uloga arhivera je pakovanje određenih klasa u JAR fajlove i pomaganje pri upravljanju JAR fajlova. Generator dokumenata služi za generisanje dokumentacije iz komentara izvornog koda.

#### 3.2. Android SDK

Android Software Development Kit (SDK) je kolekcija alata koja omogućava razvoj aplikacija za određene hardverske platforme, kompjuterske sisteme, operativne sisteme, softverske pakete, video konzole za igru i softverske frameworke. Pod razvojem aplikacija podrazumjeva se izgradnja, testiranje i debugiranje aplikacija.

#### 3.3. Java Runtime Environment

Java Runtime Environment (JRE) je kolekcija alata koja omogućava izvršavanje Java programa. JRE se sastoji od Java Virtual Machine (JVM), osnovnih klasa i fajlova potrebnih za izvršavanje Java programa. JVM je softverski program koji može interpretirati i izvršavati Java bajt kod.

### 3.4. Android studio

Android studio je integrisano razvojno okruženje (IRO) koje pruža obimnu podršku programerima za razvoj aplikacija na android platformi. Kod integrisanog razvojnog okruženja svo programiranje se vrši u jednom programu.

## 4. Sadržaj aplikacije

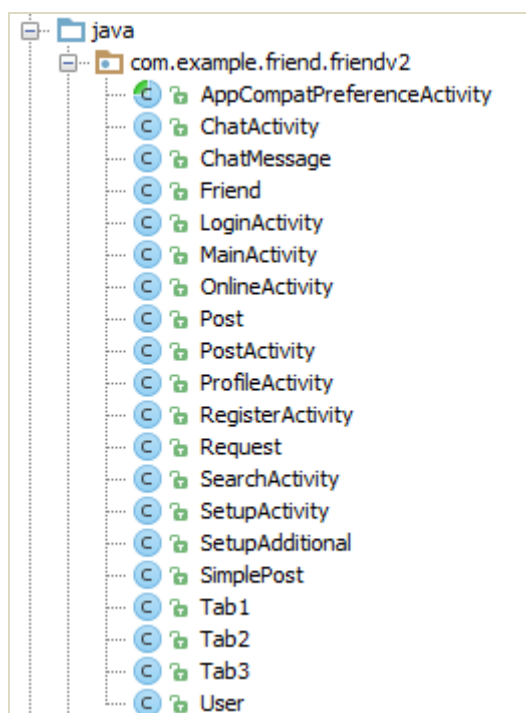
U aplikaciji se kreiraju tri foldera: manifest, java i res folderi.

### 4.1 Manifest folder

U manifest folderu se nalazi XML manifest fajl. Ovaj fajl sadrži osnovne informacije o aplikaciji, kao što su informacije o paketima i komponentama aplikacije. Osnovne informacije su neophodne za pokretanje aplikacije na android platformi.

### 4.2 Java folder

Java folder sadrži klase koje predstavljaju backend dio aplikacije. To je dio aplikacije koji korisnik ne vidi, ali je neophodan za rad aplikacije.



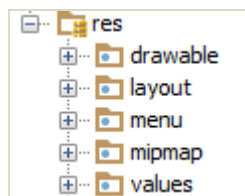
*Slika 4.2 Prikaz java foldera*

ChatActivity je klasa koja omogućava prikaz i razmjenu poruka između korisnika. ChatMessage se sastoji od atributa i metoda koje sadrže informacije o poruci. Za prijavu korisnika na aplikaciju je zadužena klasa LoginActivity. OnlineActivity služi za pregled dostupnih korisnika. PostActivity omogućava objavljivanje statusa, a Post klasa sadrži

informacije o statusu. ProfileActivity prikazuje korisničke informacije. RegisterActivity, SetupActivity i SetupAdditional služe za registraciju korisnika. Informacije o zahtjevima za prijateljstvo sadrži klasa Request. SearchActivity omogućava pretragu. Friend i User sadrže osnovne informacije o korisniku. Osnovni zadatak klase MainActivity je omogućavanje rada fragment klasa. Tab1, Tab2 i Tab3 su fragment klase, koje služe za prikaz statusa, prikaz chatova, odnosno prikaz prijatelja korisnika respektivno.

### 4.3 Res folder

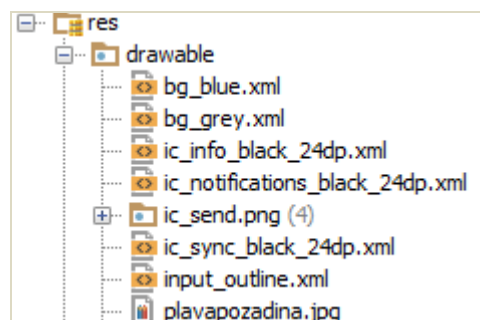
Res folder se sastoji od pet foldera. U tim folderima se definiše izgled aplikacije i oni predstavljaju frontend dio aplikacije.



*Slika 4.3. Prikaz res foldera*

#### 4.3.1 Drawable folder

Drawable folder služi za smještanje slika koje se neće skalirati, to jest neće mjenjati svoju širinu i dužinu, bez obzira na veličinu ekrana na uređaju. U folderu se mogu naći i XML fajlovi.

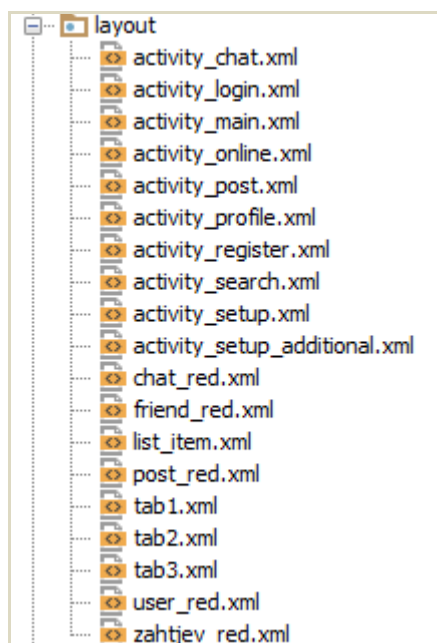


*Slika 4.3.1. Prikaz drawable foldera*

XML fajlovi prikazani na predhodnoj slici služe za definisanje pozadinske boje ili radijusa na uglovima određenih elemenata.

#### 4.3.2. Layout folder

U layout folderu se nalaze XML fajlovi koji definišu vizuelnu strukturu korisničkog interfejsa.

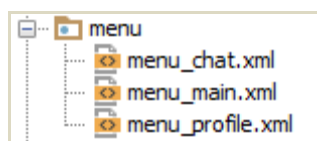


Slika 4.3.2. Prikaz layout foldera

Svaki XML fajl pripada jednoj klasi, to jest klasa koristi XML fajl kako bi prikazala određene informacije ili omogućila unos određenih informacija. Tako klasa ChatActivity koristi activity\_chat.xml fajl kako bi prikazala poruke koje se razmjenjuju između dva korisnika, klasa LoginActivity koristi activity\_login.xml kako bi omogućila prijavu korisnika na aplikaciju ili pokrenula proces registracije korisnika, itd.

#### 4.3.3. Menu folder

Meni služi za prikazivanje osnovnih opcija aplikacije, koji imaju globalni utjecaj na aplikaciju, npr. pretraga korisnika.

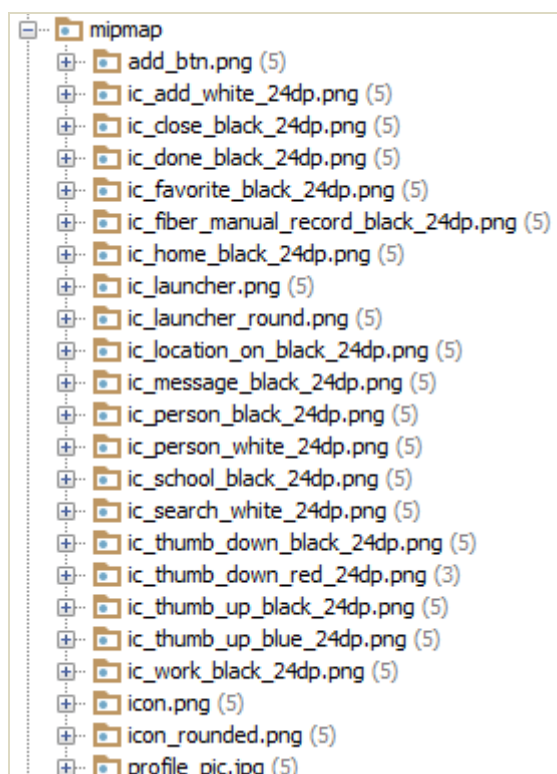


Slika 4.3.3. Prikaz menu foldera

Aplikacija sadrži tri menija. Menu\_chat.xml definiše izgled menija na stranici za dopisivanje, menu\_main.xml definiše izgled menija na glavnoj stranici i menu\_profile.xml definiše izgled menija na korisničkoj stranici.

#### 4.3.4. Mipmap folder

U mipmap folderu nalaze se slike koje se skaliraju u zavisnosti od veličine ekrana na uređaju. Najčešće su to ikone.

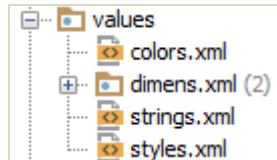


*Slika 4.3.4. Prikaz mipmap foldera*

Na slici iznad je prikazana lista foldera. Svaki folder sadrži pet različitih dimenzija iste slike, kako bi se osigurao isti kvalitet slike za bilo koju veličinu ekrana.

#### 4.3.5. Values folder

Values folder se sastoji od četiri XML fajla, a to su: colors.xml, dims.xml, strings.xml i styles.xml.



*Slika 4.3.5. Prikaz values foldera*

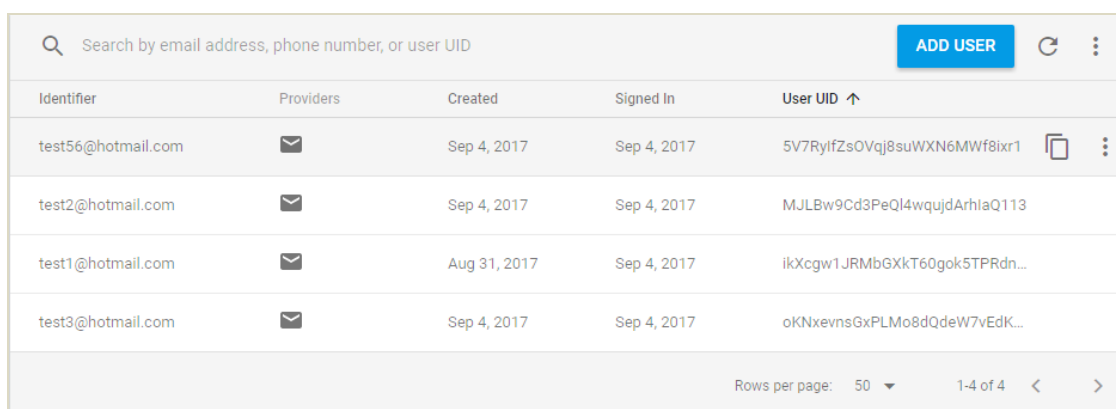
Colors.xml fajl služi za definisanje boja. Svakoj boji u colors.xml fajlu se pridružuje ime te boje, kako bi je programer lakše koristio u svom kodu. Ime bira programer i može biti proizvoljno. Dimens.xml fajl služi za definisanje dimenzija grafičkih elemenata. Definisanje stringova je omogućeno u strings.xml fajlu, a definisanje formata i izgleda grafičkih elemenata se radi u styles.xml fajlu.

## 5. Baza podataka (Firebase)

Firebase je provajder cloud i backend usluga. Osnovan je 2011., a osnivači su Andrew Lee i James Tamplin. U aprilu 2012. puštena je u upotrebu baza u realnom vremenu. Osnovni cilj Firebase je osloboditi programere od upravljanja serverima i pisanja API koda za prihvatanje podataka sa servera. Koristeći Firebase, programer se može u potpunosti fokusirati na razvoj aplikacije. Firebase podržava web, iOS, OS X i android klijente. On je dio trenutno rastućeg brenda BaaS (Backend as a service). BaaS predstavlja model koji omogućava povezivanje aplikacija sa cloud servisima koristeći aplikacioni programski interfejs (API) i komplet za razvoj softvera (SDK). Friend aplikacija koristi tri usluge koje nudi Firebase. Te usluge su: Firebase autentifikacija, Firebase baza podataka u realnom vremenu i cloud pohrana.

### 5.1. Firebase autentifikacija

Firebase podržava različite načine autentifikacije, kao što su autentifikacija pomoću emaila, Facebooka, Twittera, GitHuba i Googlea. Firebase olakšava autentifikaciju programerima i korisnicima. Programerima je ostavljeno na izbor da li da koriste već gotov interfejs za autentifikaciju ili da prave novi interfejs po svojoj želji. Nakon završene autentifikacije korisnik dobija jedinstvenu šifru koja služi za njegovu identifikaciju. Identifikaciona šifra se ne može mjenjati. Omogućene su i korisničke sesije. Korisničke sesije osiguravaju da korisnik ostane prijavljen nakon što se aplikacija restartuje.



The screenshot shows the Firebase Authentication console. At the top, there is a search bar with the placeholder text 'Search by email address, phone number, or user UID' and an 'ADD USER' button. Below the search bar is a table with the following columns: Identifier, Providers, Created, Signed In, and User UID. The table contains four rows of test users. At the bottom right, there is a pagination control showing 'Rows per page: 50' and '1-4 of 4'.

Identifier	Providers	Created	Signed In	User UID ↑
test56@hotmail.com	✉	Sep 4, 2017	Sep 4, 2017	5V7RyIfZsOVqj8suWXN6MWf8ixr1
test2@hotmail.com	✉	Sep 4, 2017	Sep 4, 2017	MJLBw9Cd3PeQl4wqujdArhlaQ113
test1@hotmail.com	✉	Aug 31, 2017	Sep 4, 2017	ikXcgw1JRMbGXkT60gok5TPRdn...
test3@hotmail.com	✉	Sep 4, 2017	Sep 4, 2017	oKNxevnsGxPLMo8dQdeW7vEdK...

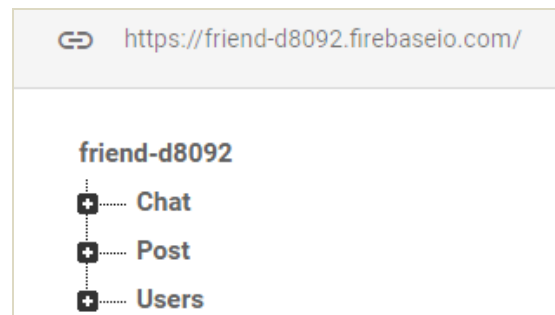
Slika 5.1. Izgled Firebase autentifikacije Friend aplikacije

### 5.2. Firebase baza podataka u realnom vremenu

Firebase baza podataka omogućava pohranu i sinhronizaciju podataka između korisnika u realnom vremenu. Podaci u bazi podataka su pohranjeni u JSON formatu. Za



razliku od većine baza podataka, koje koriste HTTP protokol za povezivanje aplikacije sa bazom, Firebase baza podataka koristi WebSocket za povezivanje, koji je puno brži od HTTP protokola. Nakon što korisnik ažurira podatke u bazi, gotovo istovremeno svi konektovani korisnici dobijaju te ažurirane podatke. Baza podataka omogućuje i „offline“ način rada. Kada korisnik izgubi konekciju sa bazom, SDK koristi lokalni keš na uređaju za pohranu podataka. Nakon što korisnik ponovno uspostavi konekciju sa bazom, podaci u lokalnom kešu se automatski kopiraju u bazu podataka. Sigurnost podataka u bazi obezbeđuje se definisanjem koji korisnik, kojem podatku, može pristupiti.



*Slika 5.2.1. Struktura baze podataka*

Baza podataka se sastoji iz tri dijela. U prvom dijelu „Chat“ su smješteni podaci vezani za dopisivanje između korisnika.



*Slika 5.2.2. Prikaz chat dijela*

Na predhodnoj slici je prikazan prostor u bazi podataka, koji omogućava smještanje podataka vezanih za dopisivanje između dva korisnika. Naziv tog prostora se dobije spajanjem identifikacioni šifri oba korisnika, kako bi se osiguralo da će prostor imati jedinstven naziv. Prostor se sastoji iz četiri dijela. Prva dva dijela omogućavaju obavješćavanje korisnika da imaju nove poruke. U trećem dijelu „Messages“ nalaze se poruke i podaci vezani za te poruke, npr. datum poruke. Zadnji dio „UsersInformation“ sadrži osnovne podatke o oba korisnika, kao što su: profilna slika, identifikaciona šifra i ime i prezime korisnika.

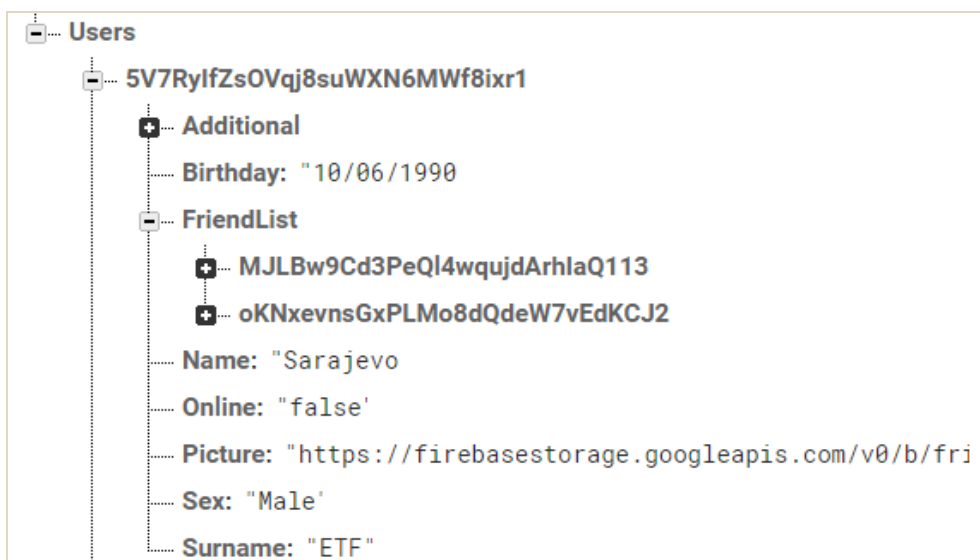
U drugom dijelu „Post“ su smješteni podaci koji predstavljaju objavljenje statusa korisnika.



*Slika 5.2.3. Prikaz post dijela*

Na slici 5.2.3. su prikazane informacije o jednom statusu. Kao i svaki korisnik, i svaki status posjeduje jedinstvenu identifikacionu šifru, koja predstavlja naziv statusa u bazi podataka. Status se sastoji od osnovnih informacija, kao što su: datum objave, identifikaciona šifra korisnika koji je objavio status, ime i prezime korisnika, opis statusa, profilna slika korisnika i slika statusa. Dodatne informacije o statusu su „Comments“ i „Like“. „Comments“ sadrži komentare korisnika vezanih za status, a „Like“ sadrži listu korisnika kojima se status sviđa. Na isti način status može imati i dodatnu informaciju „Dislike“. Kako u ovom slučaju nema korisnika kojima se status ne sviđa, tako nema ni dodatne informacije „Dislike“.

U trećem dijelu „Users“ su smješteni podaci vezani za korisnike. U ovom dijelu se nalaze podaci o svim korisnicima aplikacije Friend.



Slika 5.2.4. Prikaz users dijela

Na slici 5.2.4. su prikazane informacije o jednom korisniku. Identifikaciona šifra korisnika predstavlja naziv korisnika u bazi podataka. U bazi podataka korisnik je predstavljen preko osnovnih i dodatnih informacija. Osnovne informacije su: ime, prezime, profilna slika, spol i godina rođenja. Dodatne informacije su „Additional“ i „FriendList“. „Additional“ sadrži informacije o trenutnom prebivalištu korisnika, mjestu rođenja, završenoj srednjoj školi, završenom fakultetu i trenutnom poslu. „FriendList“ predstavlja listu prijatelja. Prikazan korisnik ima dva prijatelja, jer sadrži dvije identifikacione šifre u „FriendList“.

### 5.3. Firebase cloud pohrana

Firebase cloud pohrana omogućava slanje i skidanje različitih vrsta fajlova. Svi transferi podataka se odvijaju po sigurnoj konekciji. U slučaju gubitka konekcije tokom transfera podataka, Firebase omogućava nastavak transfera na mjestu gdje je i stao. Cloud pohrana se skalira automatski. To korisniku daje osjećaj da ima neograničen prostor za pohranu.



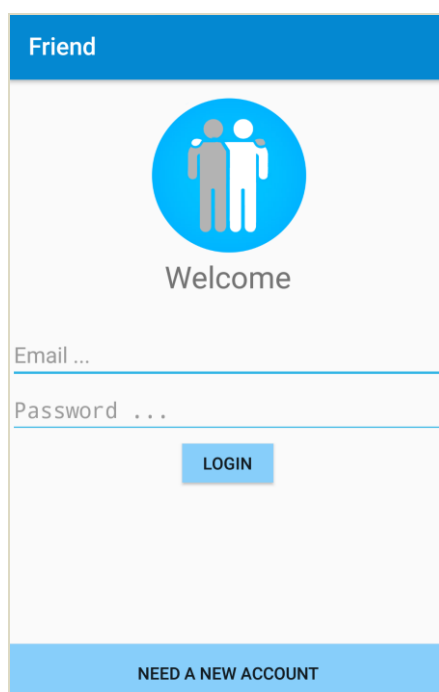
Slika 5.3. Friend cloud pohrana

## 6. Interfejsi i funkcionalnosti aplikacije

Interfejs aplikacije može se podijeliti na tri dijela: početna stranica aplikacije, korisnička stranica i glavna stranica.

### 6.1. Početna stranica aplikacije

Početna stranica aplikacije omogućava korisniku registraciju i prijavu ukoliko korisnik već posjeduje profil.



*Slika 6.1. Početna stranica aplikacije*

#### 6.1.1. Prijava

Pritiskom na dugme „Login“ korisnik započinje prijavu na aplikaciju. Prijava podrazumjeva provjeru podataka koje je korisnik unio u odgovarajuća polja. Ukoliko su uneseni podaci isti kao i oni koji se nalaze u bazi podataka, korisnik je uspješno prijavljen. U suprotnom korisnik će biti obavješten o neuspješnoj prijavi. U nastavku je tabela u kojoj je prikazan kod koji omogućava prijavu:

```
String email_s = email.getText().toString().trim();
String password_s = password.getText().toString().trim();
FirebaseAuth auth = FirebaseAuth.getInstance();
auth.signInWithEmailAndPassword(email_s, password_s).addOnCompleteListener(new
OnCompleteListener<AuthResult>() {
    @Override
```

```

    public void onComplete(@NonNull Task<AuthResult> task) {
        if(task.isSuccessful()){
            provjeraDaLiPostojiKorisnik();
        }
    }
});
public void provjeraDaLiPostojiKorisnik(){
    final String id = FirebaseAuth.getInstance().getCurrentUser().getUid();
    DatabaseReference bazaKorisnik =
        FirebaseDatabase.getInstance().getReference().child("Users");
    bazaKorisnik.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(DataSnapshot dataSnapshot) {
            if (dataSnapshot.hasChild(id)) {
                Intent intent = new Intent(LoginActivity.this, MainActivity.class);
                startActivity(intent);
                finish();
            }
        }
    });
}
}

```

*Tabela 6.1.1. Kod za prijavu korisnika*

### 6.1.2. Registracija

Nakon što korisnik pritisne dugme „Need a new account“ na početnoj stranici, otvara se stranica za kreiranje računa koja je prvi dio registracije korisnika.

*Slika 6.1.2. Stranica za kreiranje računa*

Stranica za kreiranje računa sadrži sljedeća polja:

- Polje za ime
- Polje za prezime
- Polje za email

- Polje za šifru
- Polje za potvrdu šifre
- Polje za godinu rođenja

Postoji mogućnost i odabira spola. Sva polja je neophodno popuniti da bi se račun uspješno kreirao. Polje za šifru i polje za potvrdu šifre se moraju poklapati po sadržaju. Pritiskom na dugme „Create“ račun je kreiran i osnovni podaci korisnika se nalaze u bazi podataka. Kod za kreiranje računa je prikazan u sljedećoj tabeli:

```
auth.createUserWithEmailAndPassword(email_s, password_s).addOnCompleteListener(new
OnCompleteListener<AuthResult>() {
@Override
public void onComplete(@NonNull Task<AuthResult> task) {
    if(task.isSuccessful()){
        String id = auth.getCurrentUser().getUid();
        DatabaseReference trenutniKorisnik = baza.child(id);
        trenutniKorisnik.child("Name").setValue(ime_s);
        trenutniKorisnik.child("Surname").setValue(prezime_s);
        trenutniKorisnik.child("Sex").setValue(spol);
        trenutniKorisnik.child("Picture").setValue("default");
        trenutniKorisnik.child("Birthday").setValue(godinaRodjenja_s);
    }
}
});
```

*Tabela 6.1.3. Kod za kreiranje računa*

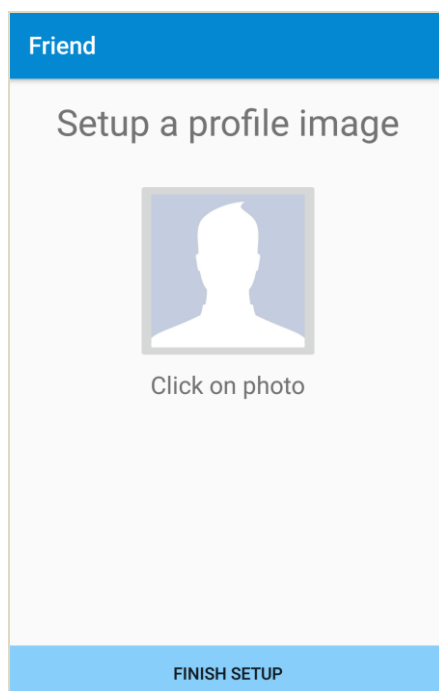
Kreiranjem računa završava se prvi dio registracije korisnika. Drugi dio registracije korisnika je unos dodatnih informacija o korisniku.

*Slika 6.1.4. Stranica za dodatne informacije*

Stranica za dodatne informacije o korisniku sadrži polja:

- Polje za trenutno prebivalište
- Polje za mjesto rođenja
- Polje za srednju školu
- Polje za fakultet
- Polje za trenutni posao
- Polje za status veze
- Polje za opis korisnika

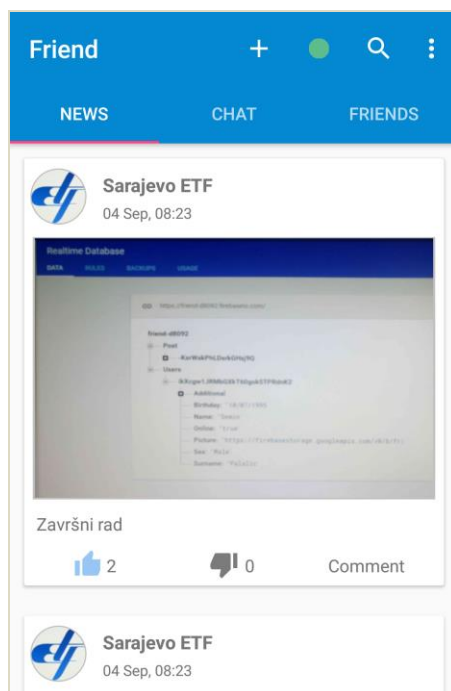
Polja nisu neophodna, dakle registracija se može nastaviti i bez popunjavanja ovih polja. Pritiskom na dugme „Submit“ podaci se unose u bazu podataka. Ukoliko je korisnik neko od polja ostavio prazno, u bazi će se naznačiti da je to polje bilo prazno. Nakon završenog unosa podataka prelazi se na treći dio registracije, postavljanje profilne slike.



*Slika 6.1.5. Stranica za postavljanje profilne slike*

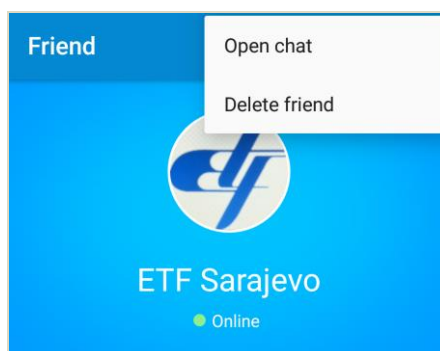
Pritiskom na sliku otvara se nova stranica gdje se vrši izbor profilne slike. Nakon odabira, vrši se obrada slike i na kraju pritiskom na dugme „Finish setup“ se završava registracija korisnika.

## 6.2. Korisnička stranica



Slika 6.2. Korisnička stranica

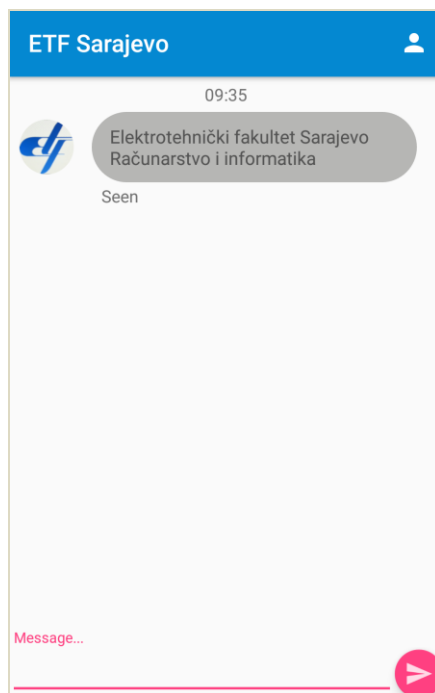
Korisnička(Profilna) stranica nudi pregled svih informacija o korisniku. Na njoj je prikazana profilna slika korisnika, ime i prezime korisnika, dodatne informacije o korisniku i pregled svih objavljenih postova korisnika. Dugme „Add friend“ omogućava slanje zahtjeva za prijateljstvo, to jest dodavanje korisnika u vlastitu listu prijatelja.



Slika 6.3. Opcije otvaranje chata i brisanje prijatelja

U gornjem desnom uglu korisničke stranice postoje dvije opcije. Prva opcija je otvaranje Chat stranice koja omogućava razmjenu poruka između dva korisnika aplikacije, a druga opcija je brisanje korisnika iz liste prijatelja ukoliko se on nalazi u njoj.





Slika 6.4. Chat stranica

U gornjem dijelu Chat stranice nalazi se ime i prezime korisnika kojem se šalje poruka. Ikona u gornjem desnom uglu omogućava otvaranje profila korisnika. Polje za unos teksta i dugme za slanje poruke se nalaze u donjem dijelu Chat stranice. Između navedena dva dijela se nalazi prostor za ispis poruka. Kao što je prikazano na predhodnoj slici, poruka se sastoji od profilne slike korisnika koji šalje poruku, teksta poruke i vremena slanja poruke. Ispod svake poruke se nalazi informacija koja govori da li je korisnik, kojem je poslana poruka, pročitao istu. Uslov za razmjenu poruka između dva korisnika je da korisnik prihvati zahtjev za prijateljstvo koji mu je poslao drugi korisnik. U nastavku je tabela sa kodom za slanje poruka:

```

FloatingActionButton send = (FloatingActionButton) findViewById(R.id.send);
final EditText input = (EditText) findViewById(R.id.input);
final DatabaseReference baza =
    FirebaseDatabase.getInstance().getReference().child("Chat").child(idChat);

send.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        //Unos poruke u bazu
        baza.child("Messages").push().setValue(new ChatMessage(input.getText()
            .toString(), idPrijavljenogKorisnika, "false"));

        baza.child(senderID).child("Seen").setValue("false");

    }
});

```

Tabela 6.5. Kod za slanje poruke

## Tabela sa kodom za ispisivanje poruka:

```
RecyclerView chat = (RecyclerView) findViewById(R.id.chat);
LinearLayoutManager l = new LinearLayoutManager(getApplicationContext());
chat.setLayoutManager(l);

final FirebaseRecyclerAdapter<ChatMessage, chatViewHolder>
chatFirebaseRecyclerAdapter = new FirebaseRecyclerAdapter<ChatMessage,
chatViewHolder>(ChatMessage.class, R.layout.list_item, chatViewHolder.class,
FirebaseDatabase.getInstance().getReference().child("Chat").child(idChat).child("Messages")) {
    @Override
    protected void populateViewHolder(final chatViewHolder viewHolder, final
ChatMessage model, final int position) {

        viewHolder.postaviTekstPoruke(model.getMessageText());
        viewHolder.postaviSeen(model.getSeen());

        String trenutnoVrijeme = DateFormat.format("dd-MM-yyyy", new Date()
            .getTime()).toString();
        String vrijemePoruke = DateFormat.format("dd-MM-yyyy",
            model.getMessageTime()).toString();
        String trenutnoVrijeme2 = DateFormat.format("yyyy", new Date()
            .getTime()).toString();
        String vrijemePoruke2 = DateFormat.format("yyyy",
            model.getMessageTime()).toString();

        if(trenutnoVrijeme.equals(vrijemePoruke)){
            CharSequence datum = DateFormat.format("HH:mm", model.getMessageTime());
            viewHolder.postaviDatumPoruke(datum);
        }
        else if(trenutnoVrijeme2.equals(vrijemePoruke2)){
            CharSequence datum = DateFormat.format("dd MMM, HH:mm",
                model.getMessageTime());
            viewHolder.postaviDatumPoruke(datum);
        }
        else {
            CharSequence datum = DateFormat.format("dd MMM yyyy, HH:mm",
                model.getMessageTime());
            viewHolder.postaviDatumPoruke(datum);
        }

        bazaKorisnik.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot dataSnapshot) {
                String slika = dataSnapshot.child(model.getSenderId())
                    .child("Picture").getValue().toString();
                viewHolder.postaviSliku(getApplicationContext(), slika);
            }
        });
    }
};
chat.setAdapter(chatFirebaseRecyclerAdapter);

public static class chatViewHolder extends RecyclerView.ViewHolder{

    View view;
    TextView tekstPoruke;
    TextView datumPoruke;
    TextView seen;
    CircleImageView profilna;

    public chatViewHolder(View itemView){
```

```

        super(itemView);
        view = itemView;
        tekstPoruke = (TextView) view.findViewById(R.id.tekstPoruke);
        seen = (TextView) view.findViewById(R.id.seenPoruke);
        datumPoruke = (TextView) view.findViewById(R.id.datumPoruke);
        profilna = (CircleImageView) view.findViewById(R.id.profilna_user);
    }

    public void postaviTekstPoruke(String s){
        tekstPoruke.setText(s);
    }

    public void postaviDatumPoruke(CharSequence s){
        datumPoruke.setText(s);
    }

    public void postaviSeen(String s){
        if(s.equals("true")){
            seen.setText("Seen");
        }
        else{
            seen.setText("Sent");
        }
    }

    public void postaviSliku(Context ctx, String slika){
        Picasso.with(ctx).load(slika)
                .resize(150,150)
                .centerCrop()
                .into(profilna);
    }
}

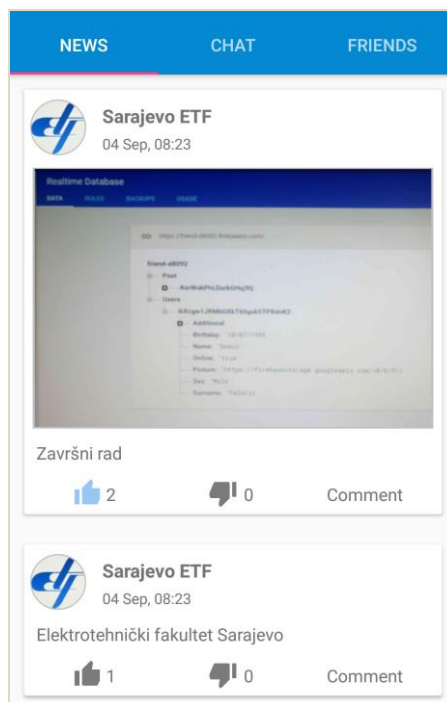
```

*Tabela 6.6. Kod za ispisivanje poruka*

### 6.3. Glavna stranica

Glavna stranica nudi sljedeće mogućnosti:

- Izbor tab stranice
- Otvaranje stranice za objavljivanje statusa
- Otvaranje Online stranice
- Otvaranje stranice za pretragu
- Izbor dodatnih opcija



Slika 6.3. Glavna stranica

Primjer implementacije opcije za pretragu i opcije odjave korisnika na meniju:

```
<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    tools:context="com.example.friend.friendv2.MainActivity">

    <item
        android:id="@+id/action_search"
        android:icon="@mipmap/ic_search_white_24dp"
        android:title="Search"
        app:showAsAction="always" />

    <item
        android:id="@+id/action_logout"
        android:orderInCategory="100"
        android:title="Log Out"
        app:showAsAction="never" />

</menu>
```

Tabela 6.4. Definisanje izgleda menija

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.menu_main, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    int id = item.getItemId();
    Intent intent;
```

```

switch (id) {
    case R.id.action_search:
        intent = new Intent(MainActivity.this, SearchActivity.class);
        startActivity(intent);
        break;
    case R.id.action_logout:
        FirebaseDatabase.getInstance().goOffline();
        FirebaseAuth.getInstance().signOut();
        intent = new Intent(MainActivity.this, LoginActivity.class);
        startActivity(intent);
        finish();
        break;
    default:
}

return super.onOptionsItemSelected(item);
}

```

*Tabela 6.5. Kod menija*

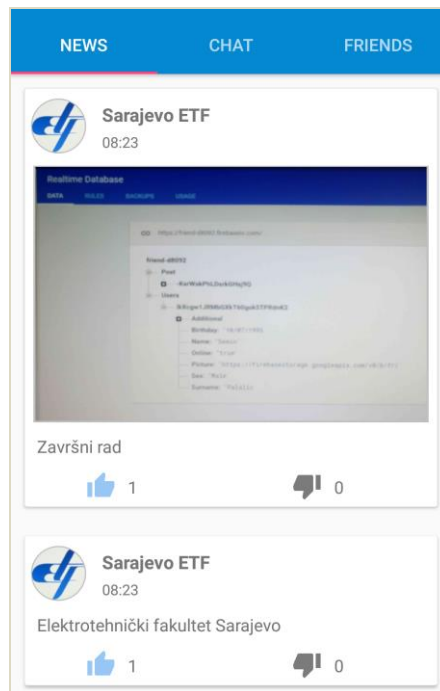
### 6.3.1. Izbor tab stranice

Korisnik ima mogućnost izbora između tri tab stranice:

- News
- Chat
- Friends

#### 6.3.1.1. News

News stranica omogućava pregled statusa svih korisnika poredanih od najnovijih do najstarijih. Jedan status sastoji se od profilne slike korisnika koji je objavio status, njegovog imena i prezimena, vremena objave statusa, slike, opisa statusa, ikone za like i dislike i komentara vezanih za status. Status može biti i bez slike. To se ostavlja na izbor korisniku koji objavljuje status. Kod za prikazivanje statusa je sličan kodu za ispisivanje poruka.



Slika 6.3.1.1. News stranica

U nastavku je prikazan kod za implementaciju click događaja za ikonu like, koji je gotovo identičan kao i kod za dislike:

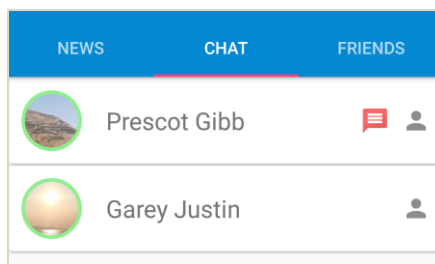
```
//Ovaj dio koda se nalazi u funkciji populateViewHolder
bazaLikeDislike = FirebaseDatabase.getInstance().getReference().child("Post");
final String id = FirebaseAuth.getInstance().getCurrentUser().getUid();
viewHolder.post_like.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        like = true;
        bazaLikeDislike.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot dataSnapshot) {
                if (like) {
                    if (dataSnapshot.child(getRef(position).getKey()).child("Like")
                        .hasChild(id)) {
                        bazaLikeDislike.child(getRef(position).getKey())
                            .child("Like").child(id).removeValue();
                        like = false;
                    } else {
                        bazaLikeDislike.child(getRef(position).getKey())
                            .child("Like").child(id).setValue("Like");
                        like = false;
                    }
                }
                bazaLikeDislike.child(getRef(position).getKey())
                    .child("Dislike").child(id).removeValue();
            }
        });
    }
});
```

```
}  
});
```

*Tabela 6.3.1. Kod za implementaciju click događaja*

### 6.3.1.2. Chat

Chat stranica služi za prikazivanje korisnika s kojim se prijavljeni korisnik dopisivao, to jest s kojim korisnicima je razmjenjivao poruke.

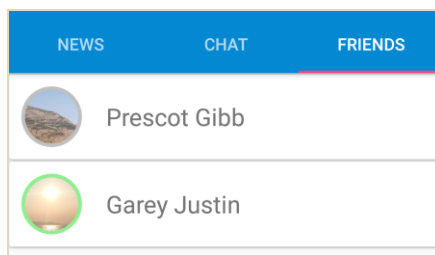


*Slika 6.3.1.2. Chat stranica*

Pritiskom na ime ili profilnu sliku nekog od korisnika, prijavljenom korisniku se otvara Chat stranica. Otvaranje profila korisnika je omogućeno pritiskom na ikonu koja se nalazi na desnom dijelu ekrana. Crvena ikona, koja se nalazi uz ime korisnika, označava da prijavljeni korisnik ima novu poruku od korisnika uz čije ime stoji ta ikona. Profilna slika nekog od korisnika uokvirena zelenom bojom znači da je taj korisnik dostupan, to jest da trenutno koristi aplikaciju. Korisnik nije dostupan ukoliko je profilna slika uokvirena sivom bojom.

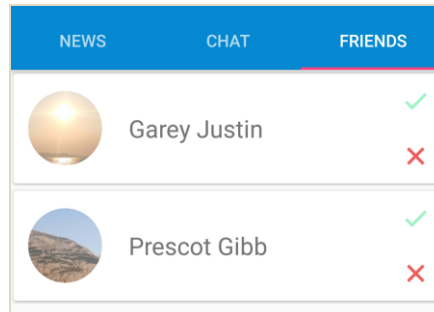
### 6.3.1.3. Friends

Prikaz svih korisnika koji se nalaze u listi prijatelja prijavljenog korisnika omogućava Friends stranica.



*Slika 6.3.1.3. Friends stranica*

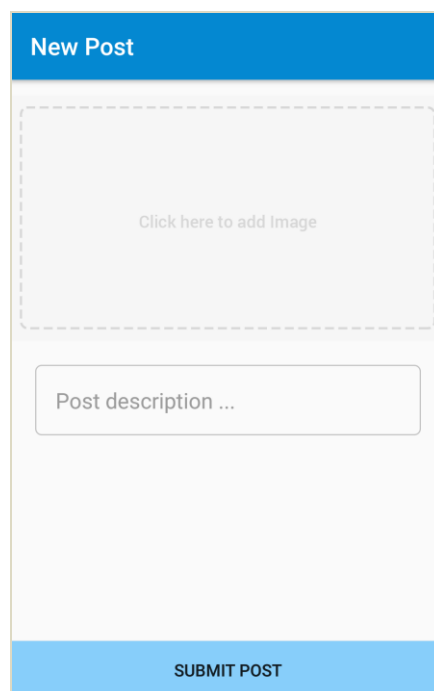
Kao i kod Chat stranice, na isti način se može vidjeti da li je korisnik dostupan. Pritiskom na profilnu sliku ili ime korisnika, za razliku od Chat stranice, otvara se profil korisnika.



*Slika 6.3.1.4. Zahtjevi za prijateljstvo*

Na Friends stranici se prikazuju i zahtjevi za prijateljstvo. Prijavljeni korisnik ima mogućnost da prihvati ili da odbije zahtjev.

### 6.3.2. Objavljivanje statusa



*Slika 6.3.2 Stranica za objavljivanje statusa*

Stranica za objavljivanje statusa sadrži prostor za sliku i polje za unos teksta koji služi za opis statusa. Slika nije neophodna za razliku od opisa statusa. Polje za sliku se ostavlja



prazno ukoliko korisnik želi status bez slike. Proces se završava pritiskom na dugme „Submit post“. U sljedećoj tabeli je primjer koda za učitavanje slike:

```
private static final int zahtjev = 1;
private Uri slikaUri = null;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    ImageButton addImage = (ImageButton) findViewById(R.id.addImage);
    addImage.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent intent = new Intent(Intent.ACTION_GET_CONTENT);
            intent.setType("image/*");
            startActivityForResult(intent, zahtjev);
        }
    });
}

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);





    if(requestCode == zahtjev && resultCode == RESULT_OK) {
        slikaUri = data.getData();
        CropImage.activity(slikaUri)
            .setGuidelines(CropImageView.Guidelines.ON)
            .setOutputCompressQuality(30)
            .start(this);
    }

    if (requestCode == CropImage.CROP_IMAGE_ACTIVITY_REQUEST_CODE) {
        CropImage.ActivityResult result = CropImage.getActivityResult(data);
        if (resultCode == RESULT_OK) {
            slikaUri = result.getUri();
            addImage.setImageURI(slikaUri);
        } else if (resultCode == CropImage.CROP_IMAGE_ACTIVITY_RESULT_ERROR_CODE) {
            Exception error = result.getError();
        }
    }
}
```

*Tabela 6.3.2.1 Kod za učitavanje slike*

### 6.3.3. Dostupni korisnici

Online stranica sadrži listu trenutno dostupnih korisnika. Pritiskom na profilnu sliku ili ime nekog od korisnika otvara se Chat stranica. Korisnička stranica korisnika se otvara pritiskom na jednu od ikona na desnoj strani ekrana.

Online friends		
	Garey Justin	
	Prescot Gibb	

*Slika 6.3.3. Online stranica*

Kod koji omogućava ispis dostupnih korisnika je prikazan u sljedećoj tabeli:

```
//Ovaj dio koda se nalazi u funkciji populateViewHolder
DatabaseReference bazaOnline =
    FirebaseDatabase.getInstance().getReference().child("Users");
bazaOnline.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot dataSnapshot) {
        if (dataSnapshot.child(getRef(position).getKey()).child("Online").getValue()
            != null) {
            String online = dataSnapshot.child(getRef(position).getKey())
                .child("Online").getValue().toString();

        }

        if (online.equals("true")) {
            viewHolder.postaviVidljivost(true);
            viewHolder.postaviBoju(online);
        }
        else {
            viewHolder.postaviVidljivost(false);
        }
    }
});

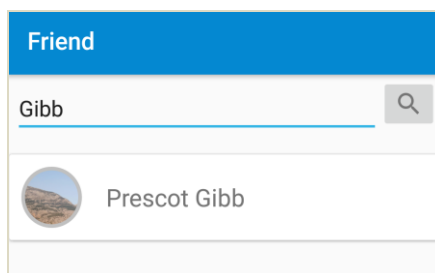
//funkcije postaviBoju i postaviVidljivost se nalaze u klasi onlineViewHolder
public void postaviBoju(String s) {
    CircleImageView profilna;
    profilna = (CircleImageView) view.findViewById(R.id.profilna_user);
    if (s != null) {
        if (s.equals("true")) {
            profilna.setBorderColor(Color.parseColor("#90EE90"));

        } else {
            userRedLinearLayout.setVisibility(View.GONE);
        }
    }
}

public void postaviVidljivost(boolean b) {
    LinearLayout linearLayout;
    linearLayout = (LinearLayout) view.findViewById(R.id.userRedLinearLayout);
    if (b) {
        linearLayout.setVisibility(View.VISIBLE);
    }
    else {
        linearLayout.setVisibility(View.GONE);
    }
}
```

*Tabela 6.3.3.1. Kod za prikaz dostupnih korisnika*

#### 6.3.4. Pretraga korisnika

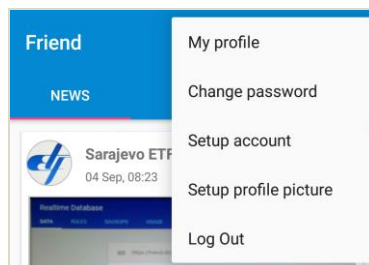


*Slika 6.3.4. Stranica za pretragu*

Pretraga korisnika je omogućena na stranici za pretragu. U gornjem desnom uglu stranice za pretragu se nalazi dugme za pretragu. Pritiskom na dugme počinje proces pretrage po tekstu koji se predhodno unio u polje za tekst.

#### 6.3.5. Izbor dodatnih opcija

Nakon pritiska na ikonu, koja se nalazi u gornjem desnom uglu glavne stranice, otvara se meni koji sadrži dodatne opcije. Meni se sastoji od pet opcija.



*Slika 6.3.5. Dodatne opcije*

Prva opcija je „My profile“. Pritiskom na prvu opciju otvara se korisnička stranica trenutno prijavljenog korisnika. Druga opcija „Change password“ služi za promjenu šifre. Treća opcija „Setup account“ omogućava otvaranje stranice za dodatne informacije korisnika gdje se vrši izmjena unesenih podataka pri registraciji korisnika. Opcija „Setup profile picture“ omogućava ponovno postavljanje profilne slike. Za odjavu korisnika sa aplikacije koristi se četvrta opcija „Log Out“.

## 7. Zaključak

*Zašto Android studio?* Na osnovu podatka da 76.6% pametnih telefona koriste Android operativne sisteme, može se zaključiti da je za razvoj aplikacija najbolje koristiti razvojna okruženje koja podržavaju Android platforme. Od razvojnih okruženja koja podržavaju Android platforme posebno se izdvaja Android studio. Njegova najveća prednost je da se svo programiranje vrši u njemu. Metodom „drag-and-drop“ omogućava veoma jednostavno dizajniranje grafičkog interfejsa. Potpuno je besplatan i svi ga mogu downloadati i modificirati.

*Zašto Firebase?* Firebase nudi mnoštvo dobrih usluga za razvoj web i mobilnih aplikacija. Firebase baza podataka pruža odličnu podršku za razvoj društvenih aplikacija gdje su potrebna ažuriranja podataka u realnom vremenu. Zahvaljujući Firebase bazi podataka u aplikaciji je implementirano dopisivanje u realnom vremenu, objavljivanje statusa i pohrana korisničkih informacija. Friend aplikacija koristi još Firebase autentifikaciju i cloud pohranu. Firebase usluge je veoma jednostavno implementirati u aplikaciji. Firebase omogućava programeru sve što je potrebno za jednu aplikaciju i potpuno je besplatan.

*Zašto koristiti Friend društvenu aplikaciju?* Aplikacija je relativno jednostavna za korištenje zahvaljujući jednostavnom interfejsu koji posjeduje. Jednostavna registracija omogućava korisniku da se registruje u par sekundi. Friend posjeduje chat u realnom vremenu koji omogućava korisniku da ostane u kontaktu sa svojom porodicom i prijateljima. Također, pomaže korisniku da prati najnovije vijesti i događaje. Korisnik ima mogućnost dijeljenja slika i svojih omiljenih trenutaka. Aplikacija nudi mogućnost provjere dostupnosti korisnika. Svaki korisnik posjeduje vlastiti korisnički profil. Za pohranu podataka koristi se Firebase koji garantira privatnost i sigurnost korisničkih podataka. Pri svakoj transakciji podataka Firebase zahtjeva 2048-bitnu enkripciju podataka. Friend aplikacija je pogodna na svim telefonima koji koriste Android operativne sisteme i nakon instalacije zauzima samo 2,5 MB memorije na uređaju.

## 8. Reference

### 8.1. Literatura

1. Yakov Fain – Java 8 programiranje
2. Jonathan LeBlanc – Programming Social Applications
3. <https://developer.android.com/studio/index.html>
4. <https://www.statista.com/>
5. <https://firebase.google.com/>
6. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3194102/>
7. <https://www.youtube.com/channel/UCI6DxakCjDR5AfRwWhWNbMg>
8. <https://material.io/icons/>
9. <http://square.github.io/picasso/>
10. <https://github.com/ArthurHub/Android-Image-Cropper>

### 8.2. Slike

1. Slika 5.1. – screenshot sa <https://console.firebase.google.com/>
2. Slika 5.2.1. – screenshot sa <https://console.firebase.google.com/>
3. Slika 5.2.2. – screenshot sa <https://console.firebase.google.com/>
4. Slika 5.2.3. – screenshot sa <https://console.firebase.google.com/>
5. Slika 5.2.4. – screenshot sa <https://console.firebase.google.com/>
6. Slika 5.3. – screenshot sa <https://console.firebase.google.com/>

Ostale slike su kreirane od strane autora.