Aplikacija za upravljanje glazbenim studijima

Krivičić, Dino

Master's thesis / Diplomski rad

2019

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Rijeka / Sveučilište u Rijeci**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:195:625298

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-28**

Sveučilište u Rijeci
Fakultet informatike
i digitalnih tehnologija

Repository / Repozitorij:

Repository of the University of Rijeka, Faculty of Informatics and Digital Technologies - INFORI Repository

zir.nsk.hr

DIGITALNA KNJIŽNICA

Image not found or type unknown

Sveučilište u Rijeci – Odjel za informatiku

Diplomski studij informatike – nastavnički smjer

Dino Krivičić

Aplikacija za upravljanje glazbenim studijima

Diplomski rad

Mentor: Izv. prof. dr. sc. Sanja Čandrlić

Komentor: dr. sc. Martina Ašenbrener Katić

Rijeka, rujan 2019.

ODJEL ZA INFORMATIKU
SVEUČILIŠTE U RIJECI

Rijeka, 07.06.2019.

Zadatak za diplomski rad

Pristupnik: Dino Krivičić

Naziv diplomskog rada: Aplikacija za upravljanje glazbenim studijima

Naziv diplomskog rada na eng. jeziku: Application for music studio management

Sadržaj zadatka:

Zadatak diplomskog rada je opisati izradu aplikacije za upravljanje glazbenim studijima. Ova aplikacija će sadržavati podatke o glazbenim studijima, producentima i glazbenicima. Procesu izrade aplikacije prethodit će analiza sustava te na temelju toga oblikovanje dijagrama entiteta i veza. U radu će biti prikazani i opisani navedeni modeli te slike osnovnih prozora aplikacije i njenih funkcija.

Mentor:

Izv. prof. dr. sc. Sanja Čandrlić

Voditelj za diplomske radove:

Izv. prof. dr. sc. Ana Meštrović

Komentor:

Dr. sc. Martina Ašenbrener Katić

Zadatak preuzet: 14.06.2019.

(Dino Krivičić)

SADRŽAJ

SAŽETAK 1

1. UVOD 2

2. ANDROID OPERACIJSKI SUSTAV 3

2.1. Povijest 4

2.2. Arhitektura 6

3. RAZVOJNI ALATI 9

3.1. Android studio 9

3.2. Java 10

3.3. MySql 11

3.4. PHP 11

3.5. MIRIS metodologija i metoda entiteti – veze 12

4. RAZVOJ ANDROID APLIKACIJE 15

4.1. Baza podataka 15

4.2. Opis aplikacije 18

4.3. Razvoj aplikacije 19

4.3.1. Android Manifest 20

4.3.2. Aktivnosti registracije i prijave 22

4.3.3. Opis aktivnosti Welcome 26

4.3.4. Opis aktivnosti Settings 26

4.3.5. Opis aktivnosti NewsFeed 29

4.3.6. Opis aktivnosti VisitProfile 32

4.3.7. Opis aktivnosti SendMail 34

4.2.8. Opis aktivnosti Hire 35

5. ZAKLJUČAK 37

LITERATURA 38

POPIS SLIKA 39

POPIS TABLICA 40

PRILOZI 41

SAŽETAK

Tema ovog diplomskog rada je izrada Android aplikacije *Find My Studio*. U prvom djelu rada ukratko je opisana arhitektura operacijskog sustava Android i njegova povijest. Opisani su razvojni alati Android Studio, Java, PHP i MySql. Osim toga, opisani su i osnovni koncepti metode entiteti – veze metodologije MIRIS. Prikazana je izrada same mobilne aplikacije. Rad je fokusiran na izradu aplikacije za operacijski sustav Android, te je to ujedno i najveći dio rada. Kod opisa razvoja aplikacije detaljno je opisana baza podataka. Prikazan je dijagram entiteta i veza te relacijska shema baze podataka. Opisana je aplikacija i sve njene funkcije. Detaljno su razrađene glavne aktivnosti aplikacije, potkrijepljene slikama grafičkog sučelja i linijama programskog koda.

Ključne riječi : android aplikacija, android studio, razvojni alati, baza podataka

1. UVOD

Pametni telefoni bilježe eksponencijalan rast prodaje iz godine u godinu, te postaju sve zastupljeniji u svakodnevnom životu. Mnogima je nezamisliv život bez pametnog telefona. Mobilni telefon, s kojim je bilo moguće slati poruke i uspostaviti pozive, evoluirao je u malo prijenosno računalo s vlastitim operacijskim sustavom i raznim aplikacijama. Tržište mobilnih aplikacija postalo je jedna od najbrže rastućih grana industrije. Ponuda aplikacija je svakim danom sve veća i danas se može naći aplikacija za gotovo sve: posao, razne usluge, vijesti, zabavu, komunikaciju, obrazovanje, itd.

Iako je ponuda aplikacija jako velika, aplikacije ipak još nisu pokrile sve aspekte života i industrije. Jedan od tih aspekata jest glazbena industrija. Iz tog razloga izrađena je aplikacija *Find My Studio*. Ova aplikacija zamišljena je kao platforma koja okuplja ključne ljude glazbene industrije na jednom mjestu. Pomoću ove aplikacije korisnici će moći brže i jednostavnije pronaći potencijalne suradnike za svoj projekt, uspostaviti kontakt s njima te dogovoriti suradnju.

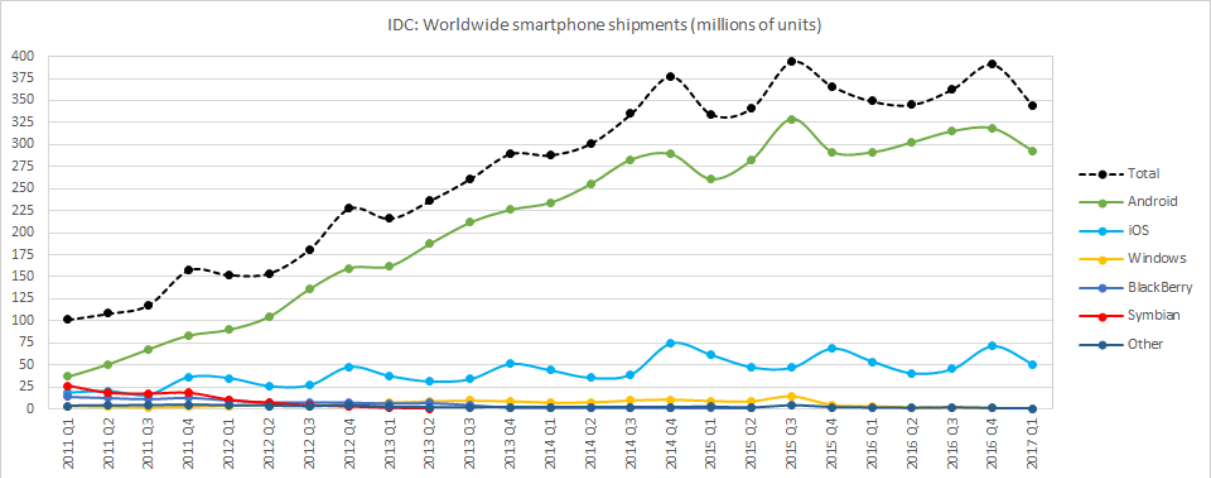
Cilj ovog diplomskog rada je izrada mobilne aplikacije za upravljanje glazbenim studijima za operacijski sustav Android. O cijelom procesu se može pročitati u poglavljima koja slijede. U prvom poglavlju riječ je o samom operacijskom sustavu Android te o njegovoj povijesti i arhitekturi. Drugo poglavlje bavi se razvojnim alatima i tehnologijama koje su bile potrebne za izradu ovakve aplikacije. Zatim slijedi opis baze podataka potrebne za spremanje izrađenih profila, odgovarajući dijagram entiteta i veza te relacijski model baze. U nastavku radi se o analizi sustava i opisu pripreme razvojnog okruženja. U zadnjem dijelu rada opisane su sve funkcije i prikazani glavni prozori *Find My Studio* aplikacije.

2. ANDROID OPERACIJSKI SUSTAV

Android je prva besplatna, potpuno prilagodljiva platforma otvorenog koda za mobilne uređaje. Razvijen je od strane Google-a, a temelji se na Linux kernel jezgri. Dizajniran je za ekrane osjetljive na dodir koje koriste uređaji poput pametnih telefona ili tableta, a sve više se počinje koristiti i kod pametnih satova, Android TV-a i multimedije u automobilima. U početku se Androidu nije predviđala svijetla budućnost upravo zbog činjenice da je temeljen na otvorenom kodu, ali unatoč tome Android je trenutno najraširenija platforma za pametne telefone.

Android je cjelokupna platforma otvorenog koda dizajnirana prvenstveno za mobilne uređaje. Odvaja hardware uređaja od software-a kojeg pokreće te tako omogućuje velikom broju uređaja pokretanje istih aplikacija, što stvara bogati ekosustav i za korisnike i programere. Programerima su dostupni svi alati i framework-ovi za što lakše i brže stvaranje mobilnih aplikacija, čak i ako nemaju fizički telefon pri ruci. Korisnicima je bitno da uređaj radi bez dodatnih podešavanja i istovremeno daje razne mogućnosti personalizacije uređaja. Proizvođačima uređaja Android pruža sve osim specifičnih drivera za određene dijelove hardware-a (Gargenta, 2011, 1-2).

Slika 1. prikazuje udio određenih mobilnih platforma na tržištu kroz vrijeme. Iz slike se može jasno iščitati da Android platforma eksponencijalno raste dok su ostale platforme u opadanju, osim iOS koji drži konstantu.



Slika 1. Prikaz udjela mobilnih operacijskih sustava na tržištu (Wikipedia, 2017)

2.1. Povijest

U listopadu 2003. godine Andy Rubin, Rich Miner, Nick Sears i Chris White osnovali su tvrtku Android Inc. u Kaliforniji. Tvrtka je prvotno bila namijenjena proizvodnji operacijskog sustava za fotoaparate, no zbog nedostatka tržišta preusmjerava svoj fokus na razvoj operacijskog sustava za pametne telefone (Unixman, 2015).

Godine 2005. tvrtku preuzima Google te time ulazi na tržište mobilnih uređaja. Prva verzija Androida pod vodstvom Google-a jest verzija 1.5 pod nazivom Cupcake. Android je postigao uspjeh zahvaljujući HTC uređajima koji ga je među prvima implementirao u svoje mobilne uređaje (Unixman, 2015).

U tu svrhu 2007. godine osnovana je tvrtka Open Handset Alliance (OHA) kojeg čine Google i brojne druge tvrtke uključujući proizvođače uređaja, aplikacija i čipova te pružatelje mobilnih usluga. Nedugo nakon osnivanja, OHA 2008. godine predstavlja Android zajedno sa SDK (eng. Software development kit) verzijom 1.0. U to vrijeme na tržištu još nije bilo uređaja s Android platformom, što je značilo da nije nužno imati uređaj za razvoj aplikacija za Android. HTC uskoro izdaje prvi komercijalni Android uređaj pod imenom G1 (poznat i kao HTC Dream) (Gargenta, 2011, 3- 4).

Android je predstavio šesnaest verzija do danas. Prve dvije verzije Androida (1.0 i 1.1) nisu imale ime, dok sve ostale verzije do danas imaju. Do sad predstavljene verzije su (AndroidAuthority, 2019):

1.5 – Cupcake

1.6. – Donut

2.1. – Enclair

2.2. – Froyo

2.3. – Gingerbread

3.2. – Honeycomb

4.0. – Ice Cream Sandwich

4.1. – Jelly Bean

4.4. – KitKat

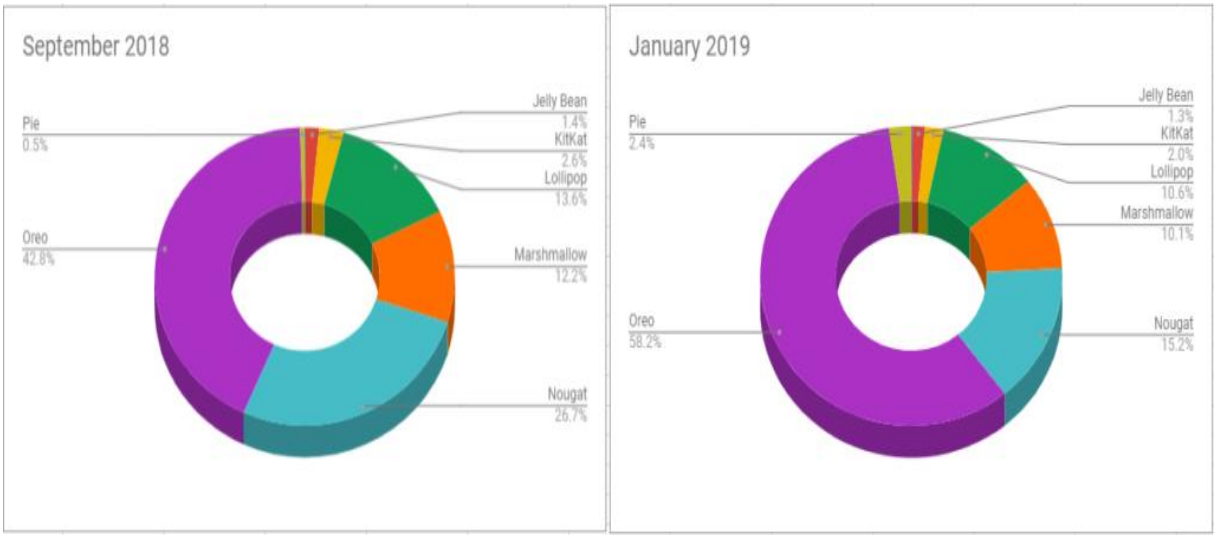
5.0. – Lollipop

6.0. – Marshmallow

7.0. – Nougat

8.0. – Oreo

9.0. – Pie



Slika 2. Prikaz udjela Android verzija na tržištu 2018. i 2019. godine (Mulder, 2019)

Slika 2. prikazuje udio raznih verzija Androida na tržištu u prosincu 2018. i siječnju 2019. Iz grafova može se iščitati da su verzije Oreo i Nougat zauzele najveći dio tržišta krajem 2018. godine s gotovo 70% tržišta. Već početkom sljedeće godine udio verzije Nougat pao je za 10%, dok se verzija Oreo popela do 60% tržišta. Iako najnovija, verzija Pie još uvijek zauzima samo 2.4% tržišta.

2.2. Arhitektura

Android je zasnovan na jezgri Linux 2.6 napisanoj u C/C++ programskom jeziku. S obzirom na to da se radi o otvorenom izvornom kodu, aplikacije putem middleware-a mogu komunicirati i pokretati druge aplikacije kao što su slanje SMS poruka, ostvarivanja poziva, pokretanja kamere i drugih aplikacija.

Arhitektura Androida podijeljena je na nekoliko razina. Na dnu arhitekture nalazi se Linux 2.6 jezgra koja sadrži drivere od kojih je najvažniji driver za međuprocesnu komunikaciju (IPC – Inter process communication). Taj driver služi za izmjenu podataka između različitih procesa ili unutar istog procesa. Još jedan važan driver jest driver za upravljanje napajanjem (Power Management).

Iznad jezgre nalaze se biblioteke pisane u C/C++ programskom jeziku (Android Tutorial Horizon, 2014):

- Surface Manager – biblioteka koja nadzire iscrtavanje grafičkog sučelja
- OpenGL | ES – biblioteka za sklopovsko ubrzavanje 3D prikaza (ako je moguća) te za visoko optimiziranu 3D softversku rasterizaciju
- SGL – 2D biblioteka upotrebljavana za većinu aplikacija
- Media Framework – biblioteka temeljena na OpenCORE koja podržava snimanje i reproduciranje poznatih audio/video formata
- FreeType – biblioteka za iscrtavanje fontova
- SSL (Secure Sockets Layer) - biblioteka za sigurnosnu komunikaciju putem interneta
- SQLite – biblioteka za upravljanje bazama podataka dostupna svim aplikacijama
- WebKit – pokretač za web preglednike
- libc – sistemska C bibliotekaa prilagođena za ugradbene sustave zasnovane na Linux OS-u

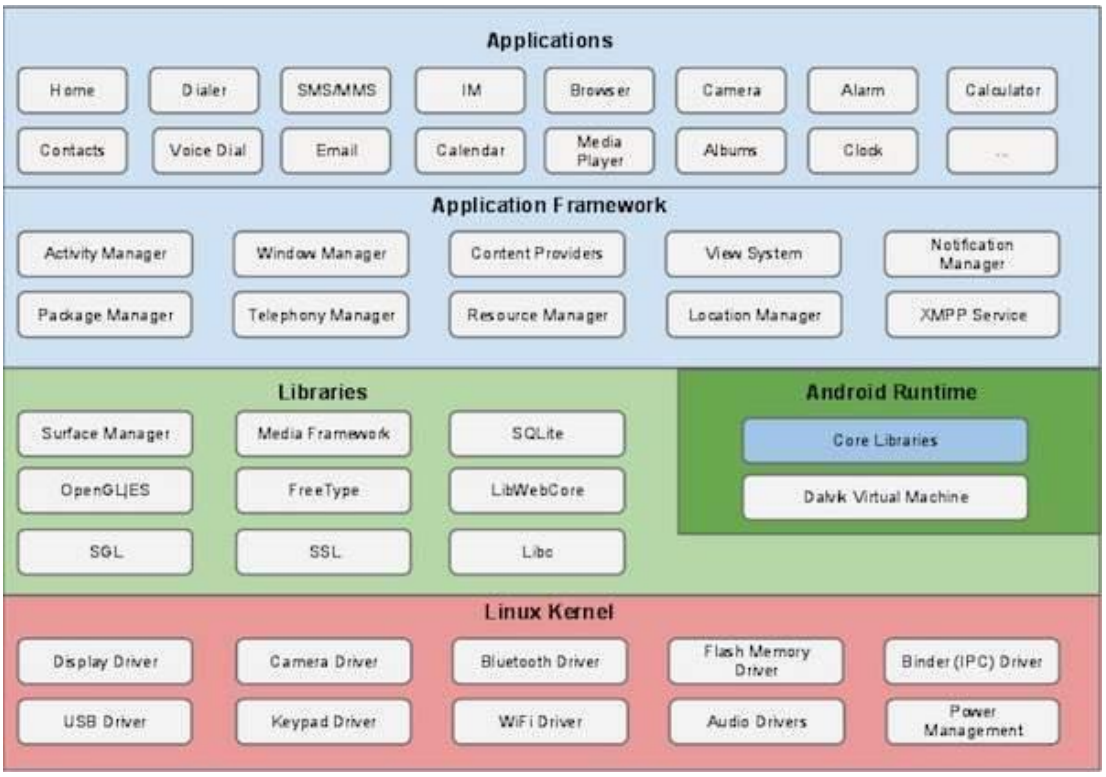
Sljedeći sloj je Android Runtime koji služi za pokretanje aplikacija. Runtime se sastoji od dvije važne komponente: Core libraries i Dalvik Virtual Machine. Core libraries je biblioteka koja sadrži većinu jezgrenih biblioteka programskog jezika Java. Dalvik Virtual Machine pokreće aplikacije kao zasebne procese odnosno instance virtualnog sloja. DVM pretvara Java class datoteke u format .dex kako bi se optimizirale za minimalnu potrošnju memorije.

Sljedeći sloj koji se nadovezuje na Dalvik je aplikacijski okvir koji se sastoji od mehanizama koji pomažu pisanju aplikacija. Aplikacijski okvir dozvoljava upotrebu svih API-ja koji su korišteni za bazne aplikacije. Time se omogućava upravljanje programskim paketima, aktivnostima aplikacija, pozivima, prozorima, resursima, kao i dohvaćanje i korištenje trenutne lokacije korisnika.

Važni elementi aplikacijskog okvira prema Neil Smyth-u, autoru knjige Android Studio 3.4 Development Essentials – Java Edition jesu:

- Activity Manager – upravlja životnim ciklusom aplikacije
- Package Manager – sadrži informacije koje su instalirane na sustav
- Window Manager – služi za upravljanje aplikacijskim prozorima
- Telephony Manager – API koji se koristi pri izradi aplikacija za upravljanje pozivima
- Content Provider – odgovoran za zajedničko korištenje podataka od strane više aplikacije
- Resource Manager – služi za pohranu dijelova aplikacije koji nisu kod (npr. slike)
- View System – sadrži bazu gotovih grafičkih prikaza i alata
- Location Manager – upravlja lokacijskim uslugama
- Notification Manager – upravljanje obavijestima i događajima.

Posljednji sloj arhitekture je aplikacijski sloj, a čine ga korisničke aplikacije. Ovaj sloj je jedini koji je vidljiv krajnjem korisniku. Neke od osnovnih ugrađenih aplikacija koje sadrži jesu e-mail klijent, SMS program, kalendar, web-preglednik ili neku od aplikacija koju je moguće preuzeti s Android Marketa. Na slici 3. prikazana je cjelokupna Android arhitektura.



Slika 3. Android arhitektura (TutoriansPoint, 2019)

3. RAZVOJNI ALATI

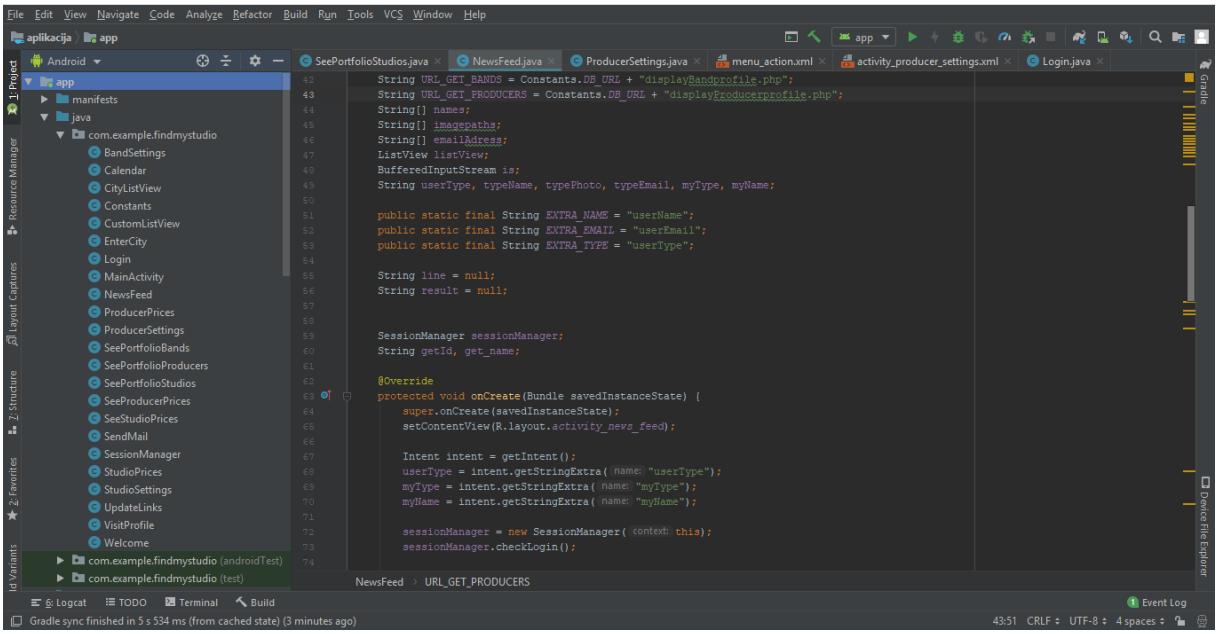
Razvojni alati (eng. Software development kit) definiraju se kao niz alata koji omogućuju kreiranje aplikacija za određene programske pakete, programska okruženja, hardverske platforme, operacijske sustave, igraće konzole i ostale razvojne platforme. Neki razvojni alati nužni su za razvoj aplikacija za pojedine platforme. Na primjer, za razvoj Android aplikacije koristeći programski jezik Java potreban je JDK (eng. Java development kit), dok je za razvoj iOS aplikacija potreban iOS SDK, te za Windows platformu koristi se .NET Framework SDK.

3.1. Android studio

Za razvoj aplikacije korišten je program Android studio. Android studio je prvi puta najavljen 16. svibnja 2013., a prva stabilna verzija objavljena je u prosincu 2014. Dostupan je za preuzimanje na Windows, MacOS i Linux operacijskim sustavima. Nastao je kao zamjena za Eclipse Android Development Tools, te je postao primarno razvojno okruženje za Android aplikacije. Neke od prednosti koje Android studio pruža jesu niz gotovih predložaka, jednostavno vizualno uređivanje korisničkog sučelja, ugrađena podrška za Google Cloud platforme te Android Virtual Device (emulator) za pokretanje aplikacije i otklanjanje neispravnosti. Pomoću AVD – a vrši se testiranje izrađene aplikacije prije upotrebe na stvarnom mobilnom uređaju. Prije nego se emulator može koristiti potrebno ga je kreirati pokretanjem AVD Managera. U AVD Manageru vrši se podešavanje željenih parametara emulatora kao npr. veličina ekrana, kamera, vrsta tipkovnica, količina memorije itd. AVD Manager nije ograničen brojem virtualnih uređaja, što omogućava testiranje aplikacije na virtualnim uređajima raznih veličina ekrana ili hardverskih specifikacija, što je vrlo bitno kod prilagodbe aplikacije za različite uređaje.

Kako je pisan u Java programskom jeziku na kojem se temelji i programiranje za Android, za uspješno funkcioniranje Android Studija nužno je na računalu imati instaliran najnoviji Java Development Kit (JDK) kojeg sam Android Studio preporučuje instalirati prije svoje instalacije (Developers, 2019).

Na slici 4. je prikazan izgled korisničkog sučelja programa Android studio. Korisničko sučelje je podijeljeno u dva dijela, editor za pisanje programskog koda te trake za odabir aktivnosti.



Slika 4. Prikaz korisničkog sučelja Android studija

3.2. Java

Java je objektno orijentirani programski jezik koji je razvijen od tvrtke Sun Microsystems, a glavni inženjeri su James Gosling i Patrick Naughton. Razvoj jave počeo je 1992. godine kao dio projekta Green, a beta verzija objavljena je u studenom 1995. godine. Cilj da jezik bude jednostavan, objektno orijentiran, robustan i siguran, višedretven, dinamičan te da se izvršava s visokim performansama. Jedna od većih prednosti u odnosu na do tada korištene programske jezike je to što se Java programi mogu pokretati na svim operacijskim sustavima za koje je razvijen Java Virtual Machine, dok je programe pisane u drugim programskim jezicima bilo potrebno prilagođavati raznim operacijskim sustavima kako bi se mogli izvoditi (Edureka, 2019).

Java je danas jedan od najkorištenijih programskih jezika, prosječan broj korisnika je preko deset milijuna. Do danas je objavljeno trinaest verzija Java programskog jezika, prva – JDK 1.0 objavljena je u siječnju 1996., a posljednja i trenutna verzija je Java SE 12 objavljena u ožujku 2019.. Standardno izdanje (Java SE) programskog jezika Java sastoji se od skupa osnovnih biblioteka. Java Micro Edition (Java ME) je izdanje koje se koristi u mobilnim uređajima, a Java Enterprise Edition (Java EE) je poslovno izdanje namijenjeno poslužiteljima za mrežne i poslovne aplikacije.

3.3. MySql

MySql je besplatan sustav otvorenog koda za upravljanje bazom podataka razvijen od Švedske tvrtke MySQL AB. Prva verzija objavljena je u svibnju 1995. te je postala čest izbor baze za projekte otvorenog koda. MySql baza optimizirana je kako bi bila brza, vrlo stabilna i ima dobro dokumentirane module i ekstenzije te podršku brojnih programskih jezika kao što su PHP, Java, Python i ostali. MySql baze su relacijskog tipa koji se pokazao kao najbolji način spremanja i pretraživanja velikih količina podataka te u suštini predstavljaju osnovu svakog većeg informacijskog sustava odnosno svakog poslovnog subjekta koji svoje poslovanje bazira na dostupnosti kvalitetnih i brzih informacija. Neke od najvećih svjetskih kompanija, poput PayPala, Twittera i Wikipedije, koriste MySql bazu podataka (JavaPoint, 2019).

Podaci se spremaju u dvodimenzionalne tablice gdje se svaka tablica sastoji od stupaca i redaka. Svaki redak u tablici predstavlja entitet koji sadrži neku informaciju, dok stupci prikazuju atribute entiteta. Tablice u bazi podataka mogu biti povezane relacijama. Postoji više tipova relacija: 1:1, 1:N i N:N. Relacija 1:1 predstavlja vezu u kojoj je jedan zapis iz tablice vezan samo za jedan zapis iz duge tablice. Relacija 1:N, jedan naprema više, predstavlja vezu u kojoj se jedan zapis u tablici može odnositi na jedan ili više zapisa u drugoj tablici. Posljednja relacija je N:N, a označava vezu više naprema više, a znači da jedan ili više entiteta iz tablice mogu biti povezani s jednim ili više entiteta u drugoj tablici.

3.4. PHP

PHP: Hypertext Preprocessor ili jednostavnije PHP je programski jezik namijenjen programiranju dinamičkih internetskih stranica. Prvi ga je razvio Rasmus Lerdorf 1994. godine za brojanje posjeta na svojoj internetskoj stranici, a daljnji razvoj preuzela je The PHP Group. To je slobodni softver koji se ističe širokom podrškom internet protokola i baza podataka kao i po raspoloživosti brojnih programskih biblioteka. Prva verzija bila je PHP/FI (Personal Home Page Tools/Forms Interpreter), a zadnja verzija PHP 7 objavljena je u prosincu 2015. godine (PHP 2019).

Jedan od razloga zbog kojeg je PHP jako dobro prihvaćen u programerskom svijetu jest slična sintaksa programskom jeziku C. Jako velik broj funkcija ima istu sintaksu. Dobar je za

početnike jer nije ograničavajući kao neki drugi programski jezici pošto nema deklaracija tipova varijabli, a istu je varijablu moguće koristiti za pohranu različitih vrsta vrijednosti. Danas se PHP koristi na skoro dvadeset milijuna raznih internetskih stranica prema istraživanju tvrtke Netcraft.

3.5. MIRIS metodologija i metoda entiteti – veze

Danas postoji nekoliko metoda za modeliranje podataka. Jedna od najpoznatijih i najkorištenijih jest metoda entiteta i veza. U ovom radu korištena je metoda entiteta i veza po MIRIS metodologiji.

Metodologija MIRIS (Metodologija za Razvoj Informacijskih Sustava) je skup metoda čiji je cilj projektirati i izgraditi Informacijski sustav (IS). Metodologiju MIRIS razvio je prof. dr. sc. Mile Pavlić, a objavljena je 1995. godine (Pavlić, 2009).

MIRIS koristi tri osnovne metode: metodu za modeliranje podataka, metodu za modeliranje procesa i metodu za modeliranje aplikacija.

Metoda entiteti – veze (EV) (engl. Entity – Relationship model, skraćeno ER) je metoda za modeliranje podataka. Model Entiteti – Veze je grafički prikaz međusobno povezanih grupa podataka promatranog sustava. EV je semantički bogata metoda za modeliranje podataka. Koncepti ove metode su bliski korisniku, pa je shema podataka jednostavna za razumijevanje i komunikaciju korisnika i projektanta (Pavlić, 2011).

Metodu Entiteti-Veze je razvio američki znanstvenik i profesor Dr. Peter Pin-Shan Chen koju je objavio u znanstvenom članku 1976. godine. Peter Chen je na „1st International Conference on Very Large Databases“ 1975. godine predstavio prvi rad na temu EV metode koji se smatra jednim od najutjecajnijih radova u informatičkoj povijesti (Pavlić, 2011).

Dijagram Entiteti – Veze (engl. Entity – Relationship Diagram) je grafički prikaz modela podataka sustava, metodom EV. Kako bi pravilno izgradili dijagram EV definiran je skup pravila kombiniranja koncepata. Pravila detaljnije određuju semantiku koncepata strukture dijagrama EV (Pavlić, 2011).

Osnovni simboli dijagrama Entiteti – Veze prikazani su u tablici 1 (Pavlić, 2011):

Tablica 1. Koncepti dijagrama entiteti – veze

KONCEPT	SIMBOL
TIP ENTITETA	
SLABI TIP ENTITETA	
TIP VEZE	
ATRIBUT	
AGREGACIJA	

Osnovni pojam metode entiteti-veze je Entitet. Entitet (engl. Enitity) je dio stvarnosti što možemo jednoznačno odrediti. Entitet može biti stvaran predmet, događaj, osoba, apstraktni pojam, dokument ili nešto drugo (Pavlić, 2011).

Slabi tip entiteta (engl. weak entity) je tip entiteta koji je na neki način zavisan drugom tipu entiteta, a ta zavisnost se prikazuje specijalnim tipom veze među jakim i slabim tipom entiteta. Na primjer, Pošiljka ovisi o Narudžbi, Grad ovisi o Državi, itd. Slabi tip entiteta crtamo s dvostrukim pravokutnikom. (Pavlić, 2011)

Atribut (engl. Attribute) tipa entiteta je funkcija koja preslikava tip entiteta u tip vrijednosti. Svaki entitet ima niz svojstava (karakteristika, odlika). Atribut je imenovana karakteristika (svojstvo) nekog entiteta. Atribut je funkcija koja pridružuje entitetu neku vrijednost iz tipa vrijednosti. Tip vrijednosti je povezan s tipom entiteta preko atributa. Primjeri atributa mogu biti: boja, datum rođenja, ime radnika, šifra dobavljača, itd. (Pavlić, 2011).

Veza (asocijacija) (engl. Relationship) je pridruživanje između entiteta. Dva tipa entiteta mogu imati više pojedinačnih veza. Veza predstavlja povezanost među entitetima u stvarnosti ili u mislima. Primjer veze: „Otac Ivan ima sina Marka“, veza „ima“ povezuje entitete „Ivan“ i „Marko“. Dva tipa entiteta mogu imati više pojedinačnih veza – tako dolazimo do Tipa veze (Pavlić, 2011).

Brojnost tipa veze je broj koji kaže koliko entiteta pojedinog tipa entiteta sudjeluje u tipu veze s entitetom iz drugog tipa entiteta (Pavlić, 2011).

Neki tipovi brojnosti su (Pavlić, 2011):

- (1,1) : (1,1) – jedan entitet prvog tipa pridružuje se najmanje nula ili najviše jednom entitetu drugog tipa i obrnuto.

- (1,1) : (0,M) – Jedan entitet prvog tipa pridružuje s najmanje nijednom i najviše jednom entitetu drugog tipa, a jedan entitet drugog tipa može se pridružiti najmanje nula i najviše mnogo entiteta prvog tipa.

- (0,M) : (0,M) – jedan entitet prvog tipa može se pridružiti najmanje nula i najviše mnogo entiteta drugog tipa i obrnuto.

4. RAZVOJ ANDROID APLIKACIJE

Analizom sustava glazbene industrije pokazala se potreba za aplikacijom ovakvog tipa. Glazbenici su do sada trebali trošiti jako puno vremena listajući razne internet stranice kako bi pronašli producenta koji odgovara njihovom stilu ili studio s odgovarajućom opremom. U istom su problemu i producenti početnici. Dok još nisu poznati u zajednici, teško pronalaze nove projekte, a i nemaju se gdje predstaviti. Zbog tih razloga rodila se ideja za izradu platforme koja bi okupljala glazbenike, producente i studija na jednom mjestu. Pomoću ove aplikacije povećala bi se vidljivost svih pojedinaca u glazbenoj industriji. Glazbenicima bi, osim vidljivosti, bio riješen problem snimanja novog materijala s profesionalnom osobom, i prije svega osobom čiji su rad imali prilike poslušati i analizirati i sa sigurnošću reći da odgovara njihovom stilu. Producenti bi, s druge strane, imali više upita za suradnju, a i sami bi mogli pronaći glazbenike s kojima žele raditi te ostvariti takvu suradnju. Ova aplikacija okuplja sve relevantne podatke o glazbenicima, producentima i studijima za lakše ostvarivanje suradnje u svačiju korist.

4.1. Baza podataka

Baza podataka aplikacije „Find My Studio“ sastoji se od sedam tablica, a to su *band*, *producer*, *studio*, *city*, *hires_bp*, *hires_bs* i *hires_ps*.

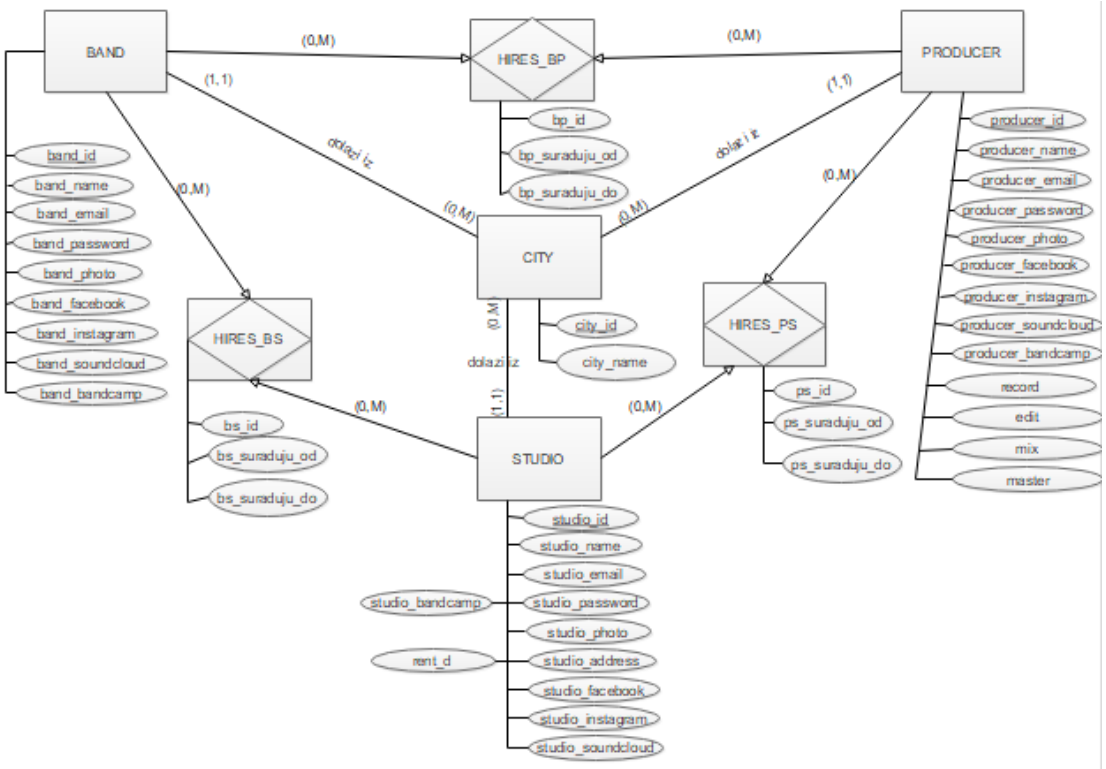
Prva tablica je *band*. U tu tablicu spremaju se svi podaci korisnika s profilom *Band*. Tablica *band* sastoji se od primarnog ključa *id*, vanjskog ključa *id_city* i još osam atributa, a namijenjena je za spremanje podataka o profilima tipa *Band*. Primarni i vanjski ključ su formata INT, atributi *band_name* i *band_email* su formata VARCHAR s maksimalno 50 znakova, dok su ostali atributi formata TEXT. Atribut *band_password* enkriptiran je funkcijom *password_hash()* u PHP djelu koda. Za enkripciju korišten je algoritam *PASSWORD_DEFAULT* koji je zadani algoritam za PHP 5.5.0. Nakon kriptiranja lozinke ona se u bazu podataka sprema kao 60 znamenkasti niz slova, brojeva i ostalih znakova. Ostali atributi su *band_photo*, u koji se sprema URL do korisnikove slike profila, te *band_facebook*, *band_instagram*, *band_soundcloud* i *band_bandcamp* u koje se spremaju poveznice do korisnikovih društvenih mreža ili internet stranica s njihovom glazbom kako bi korisnici koji ih pretražuju mogli dobiti više informacija o njima.

Tablica *producer* vrlo je slična tablici *band*. Razlika je u tome što je tablica *producer* namijenjena spremanju podataka profila tipa *Producer*, te ima četiri dodatna atributa za spremanje cijena njihovih usluga. Ti su atributi *record*, *edit*, *mix* i *master*.
Tablica *studio* namijenjena je spremanju podataka profila tipa *Studio*, te sadrži dodatne atribute *rent_d* za spremanje cijene najma studija po danu, te *studio_address* koji sadrži adresu studija.

Tablica *city* sastoji se od primarnog ključa *id* i jednog atributa *city_name* u koji se sprema ime unesenog grada.

Posljednje tri tablice *hires_bs*, *hires_bp* i *hires_ps* su agregacije. Ove tri tablice sadrže podatke o suradnjama između različitih tipa profila. *Hires_bs* sprema podatke o suradnji između profila tipa *Studio* i *Band*. Ova tablica sadrži atribute *bs_id*, *bs_suraduju_od* (početni datum suradnje) formata DATE, *bs_suraduju_do* (završni datum suradnje) formata DATE, te ključ: *id_studio* i *id_band*. *Hires_bp* sprema podatke o suradnji između profila tipa *Band* i *Producer*. Sastoji se od atributa *bp_id*, *bp_suraduju_od*, *bp_suraduju_do* i ključa: *id_band* i *id_producer*. *Hires_ps* sprema podatke o suradnji između profila tipa *Studio* i *Producer*. Sastoji se od atributa *ps_id*, *ps_suraduju_od*, *ps_suraduju_do* i ključa: *id_studio* i *id_producer*.

Dijagram entiteta i veza baze podataka za aplikaciju „Find My Studio“ napravljen je po MIRIS metodologiji te se može vidjeti na slici 5.



Slika 5. Dijagram entiteta i veza baze podataka FMS

Tablica *band* povezana je s tablicama *hires_bp*, *hires_bs* i *city*. Veza između tablica *band* i *hires_bp*, odnosno *hires_bs* je (0,M) : (1,1) što znači da jedan band može imati više suradnji, ali jedna suradnja se odnosi samo na jedan band. Ista brojnost vrijedi i za tablicu *studio* s tablicama *hires_ps* i *hires_bs*, te za tablicu *producer* s tablicama *hires_ps* i *hires_bp*.

Tablica *city* povezana je s tablicama *producer*, *studio* i *band* brojnošću (0, M) : (1, 1) što znači da se jedan grad može odnositi na više *bandova*, *producenata* ili *studija*, ali jedan *band*, *producer* ili *studio* može biti samo iz jednog grada.

U nastavku je prikazan relacijski model baze podataka.

BAND (band_id, band_name, band_email, band_password, band_photo, band_facebook, band_instagram, band_soundcloud, band_bandcamp, *id_city*)

PRODUCER (producer_id, producer_name, producer_email, producer_password, producer_photo, producer_facebook, producer_instagram, producer_soundcloud, producer_ bandcamp, record, edit, mix, master, *id_city*)

STUDIO (studio_id, studio_name, studio_email, studio_password, studio_photo, studio_adress, studio_facebook, studio_instagram, studio_soundcloud, studio_bandcamp, rent_d, *id_city*)

CITY (city_id, city_name)

HIRES_BP (id_band, id_producer, bp_id, bp_suraduju_od, bp_suraduju_do)

HIRES_PS (id_producer, id_studio, ps_id, ps_suraduju_od, ps_suraduju_do,)

HIRES_BS (id_band, id_studio, bs_id, bs_suraduju_od, bs_suraduju_do,)

4.2. Opis aplikacije

Aplikacija za upravljanje glazbenim studijima – *„Find My Studio“* zamišljena je prvenstveno kao poslovna platforma čiji je fokus na glazbenicima, producentima i glazbenim studijima. Putem ove aplikacije korisnik može naći informacije o drugim korisnicima aplikacije u svrhu poslovnog povezivanja.

Kako bi mogao koristiti aplikaciju, korisnik se mora registrirati, gdje uz unošenje osnovnih podataka (ime, e-mail adresa i lozinka), ima opciju registrirati se kao glazbenik (*Band*), producent (*Producer*) ili glazbeni studio (*Studio*). Svaka od navedene tri opcije ima jedinstvene mogućnosti. Nakon registracije, korisnik se mora prijaviti u aplikaciju, te ga se preusmjerava na početnu aktivnost aplikacije – *„Welcome“*. U ovoj aktivnosti korisnik može izabrati koju vrstu profila korisnika želi pregledati, te klikom na gumb *„Go“* odlazi na sljedeću aktivnost - *„News Feed“* - koja prikazuje sve korisnike odabranog tipa koji koriste aplikaciju. Također, na *„Welcome“* aktivnosti korisnik može promijeniti neke od svojih podataka pritiskom na gumb *„Settings“*.

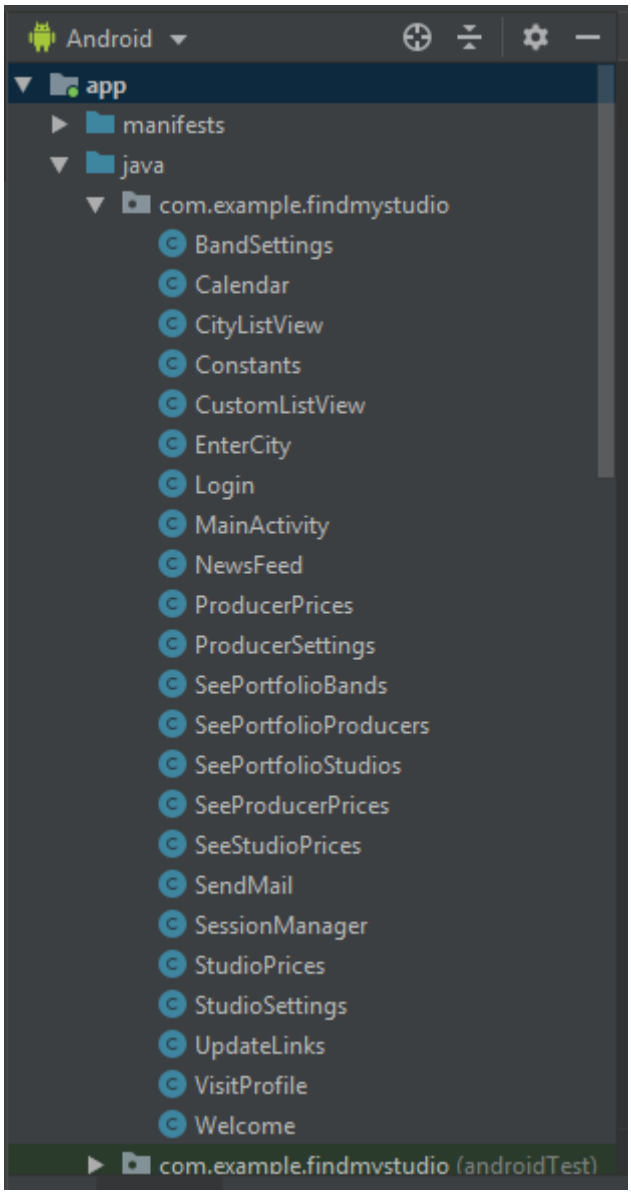
Promjena imena, e-mail adrese, profilne slike, gumb za odjavu i poveznica na druge društvene mreže ili internet stranice s autorskom glazbom zajednička je opcija svih triju profila. Kod *Producer* i *Studio* profila dodatna je opcija *„Edit prices“* gdje producent može postaviti cijene za usluge snimanja (*Record*), uređivanja (*Edit*), miksanja (*Mix*) i masteriranja (*Master*) jedne pjesme, a studio može postaviti cijene za dnevno iznajmljivanje (*Rent*).

Kada korisnik posjećuje druge profile vidljiva mu je slika profila, poveznice na vanjske aplikacije koje može posjetiti, gumb *„Send Mail“* te, ako je posjećeni profil tipa *Producer* ili *Studio*, vidljiv je i gumb *„See Prices“*. Pritiskom na gumb *„Send Mail“* otvara se forma za pisanje e-maila, a pritiskom na *„See Prices“* otvara se popis cijena usluga odabranog profila.

Ako korisnik tipa *Band* posjećuje profile *Studio* ili *Producer* vidljiv mu je i gumb *„Hire“* koji vodi do aktivnosti za zahtjev suradnje. U ovoj aktivnosti korisnik odabire dva datuma, početni i završni dan željene suradnje. Također ovoj aktivnosti može pristupiti i korisnik tipa *Producer* ako želi surađivati s korisnikom tipa *Studio*, te korisnik tipa *Studio* ako želi surađivati s korisnikom tipa *Producer*.

4.3. Razvoj aplikacije

Svaka Android aplikacija sadrži tri glavne mape - *manifests, java* i *res*. U manifests mapi nalazi se AndroidManifest.xml datoteka koja služi za pokretanje cijelog projekta, što je čini najvažnijom datotekom u projektu. U mapi *java* spremljene su sve java datoteke, odnosno pozadinski kod aplikacije. Zadnja datoteka je *res* datoteka (eng. resources) koja je podijeljena na nekoliko podmapa. U *drawable* mapi su smještene slikovne datoteke koje se koriste u aplikaciji. U *layout* mapi nalaze se XML datoteke aktivnosti aplikacije. Te datoteke zapravo predstavljaju ekran koji korisnik vidi. Svaka aplikacija sastoji se od više povezanih aktivnosti, te svaka aktivnost ima svoju XML datoteku u kojoj je zapisano kako pojedina aktivnost izgleda na ekranu uređaja. U tim datotekama zapisano je koji će elementi biti u aktivnosti, kako će biti raspoređeni, veličina fontova, boje i ostali elementi dizajna. U *menu* mapi nalaze se informacije o izbornicima. U *minimap* mapi spremljene su ikone koje se koriste u aplikaciji. Kod kreiranja ikona Android studio sprema istu ikonu u više različitih veličina, te koristi onu koja najbolje odgovara veličini ekrana uređaja. Posljednja mapa je *values*. Ona sadrži *strings.xml*, *styles.xml* i *colors.xml* datoteke. U *strings.xml* datoteci spremljene su string vrijednosti koje se kasnije pozivaju u aplikaciji, dok su *styles.xml* i *colors.xml* zadužene za dizajn aplikacije. Na slici 6. se može vidjeti raspored mapa i datoteka FMS aplikacije u Android studiu.



Slika 6. Prikaz datoteka u Android studiju

4.3.1. Android Manifest

Kao što je ranije spomenuto, svaka Android aplikacija koristi Android Manifest datoteku. U njoj se nalaze ključne informacije o aplikaciji koje su operacijskom sustavu potrebne za pokretanje aplikacije. Ova datoteka mora sadržavati ime paketa aplikacije, pomoću kojeg aplikacija dolazi do napisanih kodova. U Android Manifest datoteci također su napisane sve komponente aplikacije – *activities, services, broadcast receivers* i *content providers*. Svaka od tih komponenti mora imati definirane značajke poput imena njenih java klasa. Dopuštenja koja

aplikacija treba za korištenje zaštićenih dijelova sistema kao na primjer spajanje na Internet ili Bluetooth. U nastavku je Android Manifest kod aplikaciju *Find My Studio*.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.findmystudio">

    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission
android:name="android.permission.READ_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"
/>

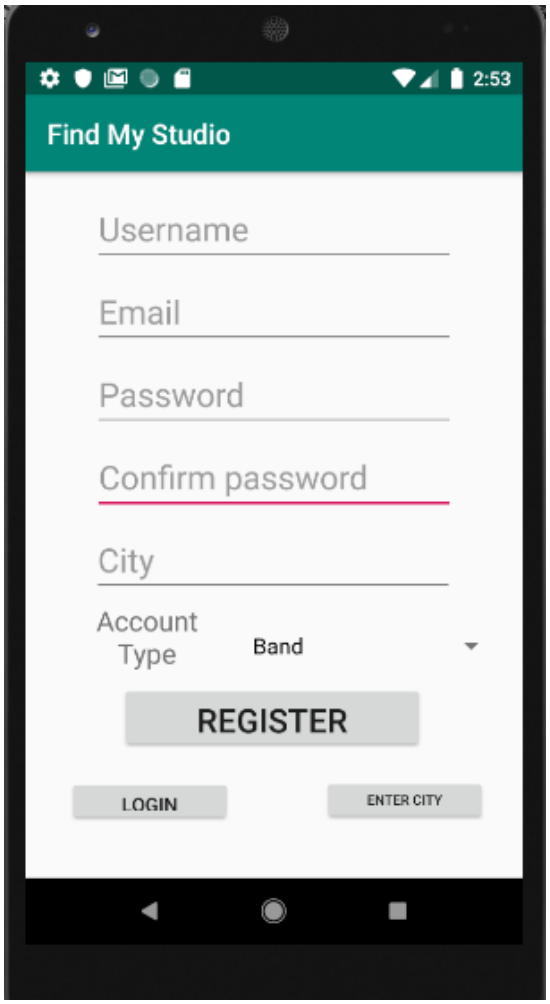
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".EnterCity"></activity>
        <activity android:name=".Calendar" />
        <activity android:name=".SeePortfolioBands" />
        <activity android:name=".SeePortfolioStudios" />
        <activity android:name=".SeePortfolioProducers" />
        <activity android:name=".SendMail" />
        <activity android:name=".SeeProducerPrices" />
        <activity android:name=".SeeStudioPrices" />
        <activity android:name=".VisitProfile" />
        <activity android:name=".UpdateLinks" />
        <activity android:name=".ProducerPrices" />
        <activity android:name=".StudioPrices" />
        <activity android:name=".StudioSettings" />
        <activity android:name=".ProducerSettings" />
        <activity android:name=".BandSettings" />
        <activity android:name=".Welcome" />
        <activity android:name=".NewsFeed" />
        <activity android:name=".Login" />
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER"
/>
            </intent-filter>
        </activity>
    </application>

</manifest>
```

4.3.2. Aktivnosti registracije i prijave

Kod pokretanja aplikacije prva aktivnost koju vidimo jest *MainActivity* odnosno aktivnost registracije u aplikaciju. Ova aktivnost od korisnika zahtjeva unošenje osnovnih podataka za registraciju (ime, e-mail adresa, lozinka, provjera lozinke i grad), te klikom na gumb „Registracija“ sprema zapisane podatke u bazu podataka. Na slici 7. vidi se izgled aktivnosti registracije.



Slika 7. Aktivnost registracije

Java kod za aktivnost registracije priložen je u nastavku. Metoda *Regist* koristi argument *String Type* koji sadrži podatak o tipu profila koji korisnik izrađuje. Aplikacija sprema podatke koje je korisnik unio u stringove koji se šalju POST metodom putem PHP skripti *register_band, register_producer* ili *register_studio*, ovisno o kojem se tipu profila radi, u bazu podataka

pomoću Volley biblioteke. U nastavku je prikazan kod za *Regist* metodu, odnosno za registraciju.

```
private void Regist(String Type) {
    loading.setVisibility(View.VISIBLE);

    final String name = this.name.getText().toString().trim();
    final String email = this.email.getText().toString().trim();
    final String password = this.password.getText().toString().trim();
    final String city_name = city.getText().toString().trim();

        StringRequest stringRequest = new
StringRequest(Request.Method.POST, Type,
                new Response.Listener<String>() {
                    @Override
                    public void onResponse(String response) {
                        try {
                            JSONObject jsonObject = new
JSONObject(response);
                            String success =
jsonObject.getString("success");

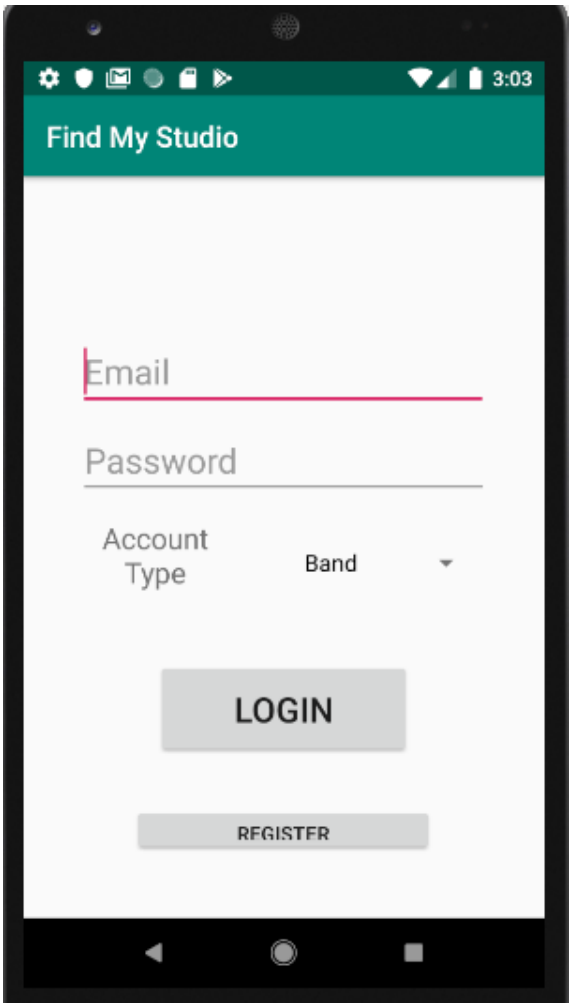
                            if (success.equals("1")) {
                                Toast.makeText(MainActivity.this, "Register
Success!", Toast.LENGTH_SHORT).show();

                                Intent intent = new
Intent(MainActivity.this, Login.class);
                                startActivity(intent);
                            }
                        } catch (JSONException e) {
                            e.printStackTrace();
                            Toast.makeText(MainActivity.this, "User with
this data already exists, or city you entered doesn't exist in database!
Try enter new city first!", Toast.LENGTH_LONG).show();
                            loading.setVisibility(View.GONE);
                            btn_regist.setVisibility(View.VISIBLE);
                        }
                    }
                },
                new Response.ErrorListener() {
                    @Override
                    public void onErrorResponse(VolleyError error) {
                        Toast.makeText(MainActivity.this, "Register Error!"
+ error.toString(), Toast.LENGTH_SHORT).show();
                        loading.setVisibility(View.GONE);
                        btn_regist.setVisibility(View.VISIBLE);
                    }
                }) {
            @Override
            protected Map<String, String> getParams() throws
AuthFailureError {
                Map<String, String> params = new HashMap<>();
                params.put("name", name);
                params.put("email", email);
                params.put("password", password);
                params.put("city", city_name);
```

```
                return params;
            }
        };

        RequestQueue requestQueue = Volley.newRequestQueue(this);
        requestQueue.add(stringRequest);
}
```

Kada se korisnik registrirao, aplikacija ga preusmjerava na sljedeću aktivnost – *LoginActivity*. Kako bi se korisnik prijavio u aplikaciju, treba unijeti svoju e-mail adresu i lozinku koju je prethodno postavio te odabrati tip profila. Na slici 8. vidi se izgled aktivnosti prijave.



Slika 8. Aktivnost prijave

Ova aktivnost koristi sličan kod kao i za registraciju osim, umjesto da sprema upisane podatke u bazu podataka, ovaj kod provjerava postoji li već neki korisnik s unesenim podacima, te ako

postoji sprema njegove podatke u klasu *SessionManager* koja koristi te podatke u sljedećim aktivnostima aplikacije. U nastavku je prikazan kod klase *SessionManager*.

```
public class SessionManager {

    SharedPreferences sharedPreferences;
    public  SharedPreferences.Editor editor;
    public Context context;
    int PRIVATE_MODE = 0;

    private static final String PREF_NAME = "LOGIN";
    private static final String LOGIN = "IS_LOGIN";
    public static final String NAME = "NAME";
    public static final String EMAIL = "EMAIL";
    public static final String ID = "ID";
    public static final String AT = "ACCOUNT_TYPE";


public SessionManager(Context context) {
    this.context = context;
    sharedPreferences = context.getSharedPreferences("LOGIN",
PRIVATE_MODE);
    editor = sharedPreferences.edit();
}

public void createSession(String name, String email, String id){
    editor.putBoolean(LOGIN, true);
    editor.putString(NAME, name);
    editor.putString(EMAIL, email);
    editor.putString(ID, id);
    editor.apply();
}

public  boolean isLogin(){
    return sharedPreferences.getBoolean(LOGIN, false);
}

public void checkLogin(){
    if (!this.isLogin()){
        Intent i = new Intent(context, LoginActivity.class);
        context.startActivity(i);
    }
}
public void logout(){

    editor.clear();
    editor.commit();
    Intent i = new Intent(context, Login.class);
    context.startActivity(i);
    ((BandSettings) context).finish();
}
```

```
public void logoutP(){

    editor.clear();
    editor.commit();
    Intent i = new Intent(context, Login.class);
    context.startActivity(i);
    ((ProducerSettings) context).finish();
}

public void logoutS(){

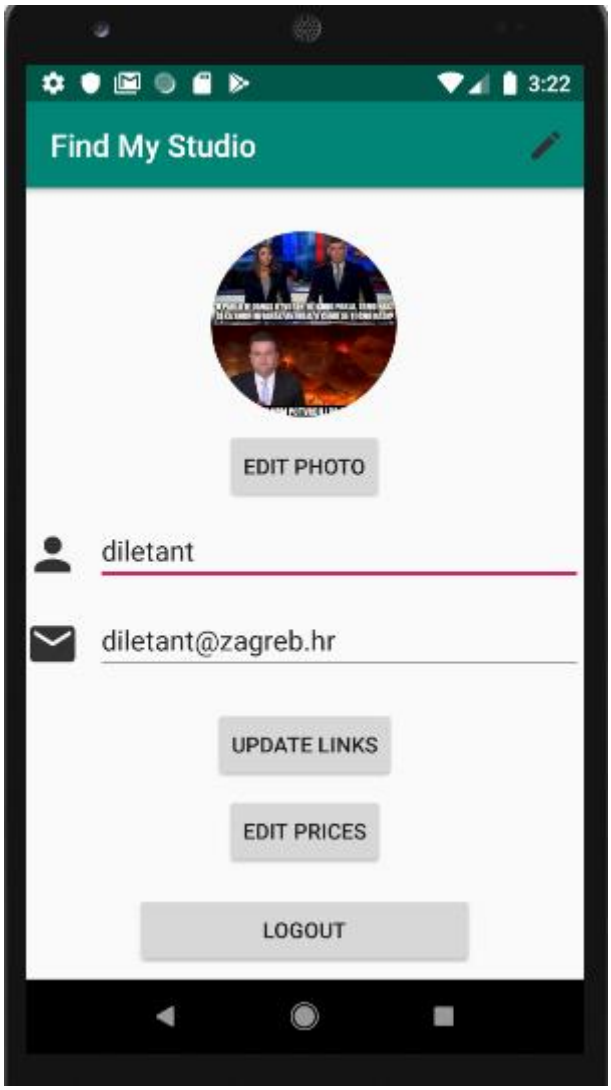
    editor.clear();
    editor.commit();
    Intent i = new Intent(context, Login.class);
    context.startActivity(i);
    ((StudioSettings) context).finish();
}
```

4.3.3. Opis aktivnosti Welcome

Nakon što se korisnik prijavio u aplikaciju prikazuje se aktivnost *Welcome*. Glavni elementi ove aktivnosti su slika profila i ime prijavljenog korisnika, gumb *Settings* te izbornik za pretraživanje profila drugih korisnika po kategorijama tipova profila. Klikom na gumb *GO* otvara se aktivnost *News Feed*, u kojoj su izlistani svi korisnici traženog tipa profila. Ako korisnik želi promijeniti neke od svojih podataka ili izgraditi svoj profil do kraja, to će učiniti klikom na gumb *Settings*.

4.3.4. Opis aktivnosti Settings

Kada je korisnik kliknuo na gumb *Settings* pokreće se nova aktivnost ovisno s kojim tipom računa je korisnik prijavljen. Zajedničko svim računima jesu elementi promjene slike profila, promjena imena, e-mail adrese te gumb *Update Links*. Gumb koji se nalazi samo na profilima *Studio* i *Producer* jest gumb *Edit Prices*. Klikom na gumb *Edit Photo* korisnik može zamijeniti svoju sliku profila nekom drugom slikom iz memorije svog uređaja. Kada to učini stara slika briše se iz baze podataka te je zamijenjena novom slikom. Na slici 9. prikazan je izgled ove aktivnosti ako je profil tipa *Producer*.

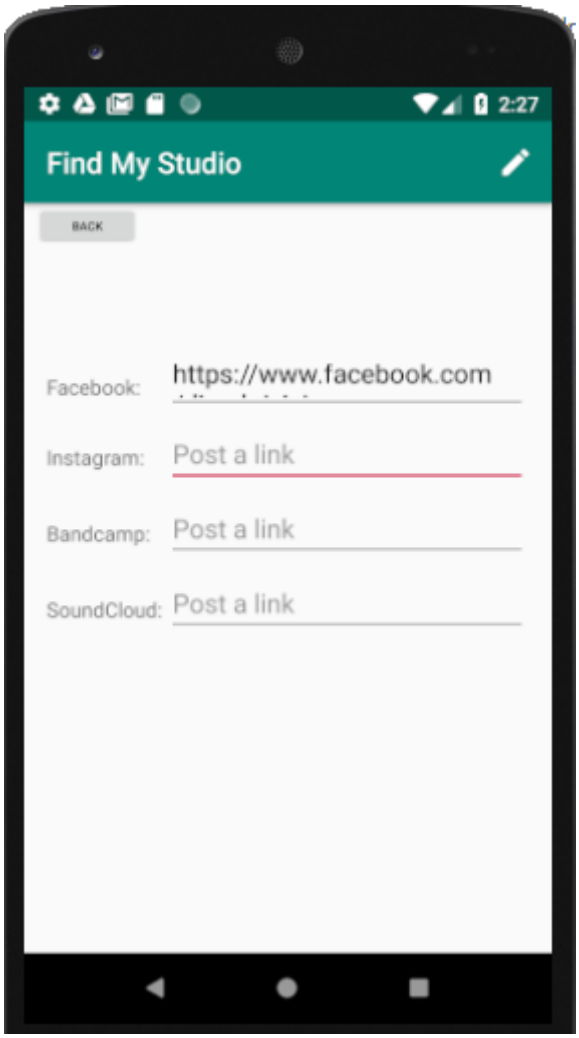


Slika 9. Aktivnost Settings

Ako korisnik želi postaviti poveznice na vanjske internetske stranice to može napraviti klikom na gumb *Update links*. Klikom na ovaj gumb pokreće se aktivnost *UpdateLinks* u kojoj postoji mjesto za upisivanje četiri poveznice namijenjene za društvene mreže ili put do glazbe korisnika. Tekst koji je korisnik upisao u predviđeno polje pretvoreno je u poveznicu pomoću *Linkify()* metode. U nastavku je prikazan dio koda za pretvaranje teksta u poveznicu.

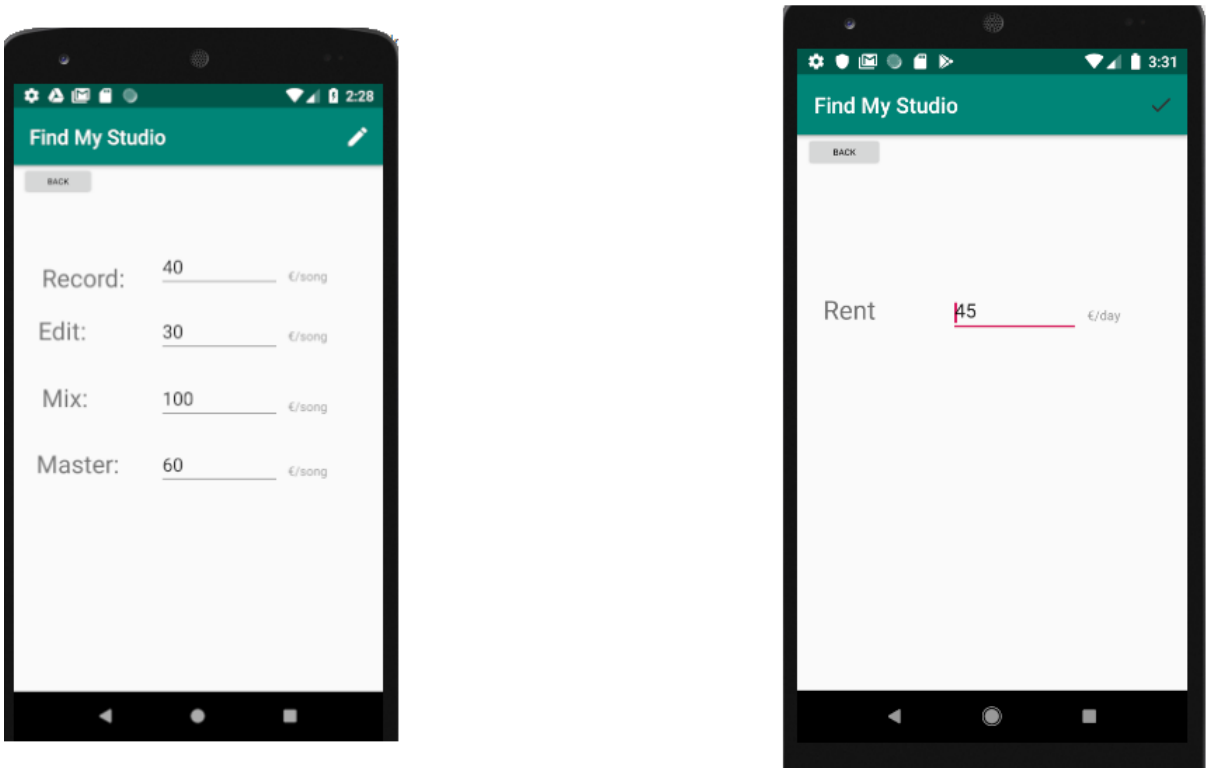
```
if(strFB.isEmpty())
    continue;
else {
    fb.setText(strFB);
    Linkify.addLinks(fb, Linkify.WEB_URLS);
}
```

Izgled aktivnosti *Update links* može se vidjeti na slici 10.



Slika 10. Aktivnost UpdateLinks

Korisnici profila *Studio* i *Producer* mogu u aplikaciju upisati svoje cijene pojedinih usluga klikom na gumb *Edit Prices*. Tako se na profilu *Producer* nalazi forma za upisivanje cijena za snimanje, uređivanje, miksanje i masteriranje za jednu pjesmu, a na profilu *Studio* može se upisati cijena najma studija za jedan dan. Klikom na ikonu olovke u gornjem desnom kutu aplikacije otvara se mogućnost pisanja u polja, te na njeno mjesto dolazi ikona kvačica koju je potrebno kliknuti pri završetku da bi se promjene spremile. Na slici 11. vidljive su spomenute aktivnosti.



Slika 11. Aktivnost ProducerPrices (lijevo) i StudioPrices (desno)

4.3.5. Opis aktivnosti NewsFeed

Aktivnost *NewsFeed* pokreće se kada korisnik u aktivnosti *Welcome* odabere tip profila koji pretražuje i klikne na gumb *GO*. U ovoj aktivnosti prikazuju se svi profili odabranog tipa u obliku liste.

Za popunjavanje liste korisnika aplikacije zadužena je klasa *CustomListView*. Ova klasa koristi *users_listview_detail.xml* datoteku u kojoj su definirani elementi liste, a to su slika profila i ime korisnika. Kada se u *NewsFeed* klasi pozove metoda `CollectProfiles(String URL, String name, String photo, String email)` prikupljeni podaci prikazuju se na ekranu u obliku liste. U nastavku je prikazan kod za *CollectProfiles* metodu.

```
private void CollectProfiles(String URL, String name, String photo, String
email) {

    try {
        URL url = new URL(URL);
        HttpURLConnection con = (HttpURLConnection)url.openConnection();
        con.setRequestMethod("GET");
        is = new BufferedInputStream(con.getInputStream());

    }catch (Exception ex){
        ex.printStackTrace();
```

```
    }

    try{
        BufferedReader br = new BufferedReader(new InputStreamReader(is));
        StringBuilder sb = new StringBuilder();

        while ((line = br.readLine()) != null) {

            sb.append(line+"\n");
        }
        is.close();
        result = sb.toString();

    }catch (Exception ex) {
        ex.printStackTrace();
    }

    try{
        JSONArray ja = new JSONArray(result);
        JSONObject jo = null;
        names = new String [ja.length()];
        imagepaths = new String[ja.length()];
        emailAdress = new String[ja.length()];

        for(int i=0; i<=ja.length(); i++){

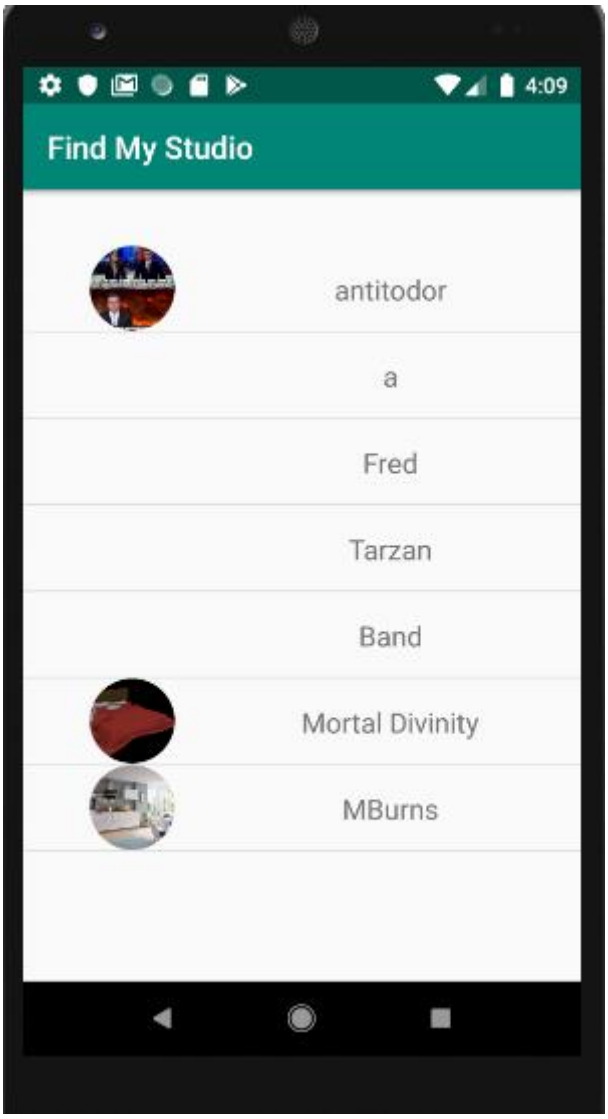
            jo=ja.getJSONObject(i);
            names[i] = jo.getString(name);
            imagepaths[i] = jo.getString(photo);
            emailAdress[i] = jo.getString(email);
        }
    }catch (Exception ex){
        ex.printStackTrace();
    }
}
```

Korisnik može posjetiti profil svih korisnika aplikacije tako da klikne na željeni profil u listi. Klikom na profil otvara se nova aktivnost u koju su proslijeđeni podaci odabranog profila. Ti podaci služit će za komuniciranje s bazom podataka i upisivanje točnih vrijednosti u aktivnost *VisitProfile*. U nastavku je prikazan kod za omogućavanje klika na objekt u listi traženih profila.

```
listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position,
long id) {
        Intent intent = new Intent(NewsFeed.this, VisitProfile.class);
        intent.putExtra(EXTRA_NAME, names[position]);
        intent.putExtra(EXTRA_EMAIL, emailAdress[position]);
        intent.putExtra(EXTRA_TYPE, userType);
        intent.putExtra("myType", myType);
        intent.putExtra("myName", myName);

        startActivity(intent);
    }
});
```

Izgled aktivnosti *NewsFeed* može se vidjeti na slici 12.



Slika 12. Aktivnost NewsFeed

4.3.6. Opis aktivnosti VisitProfile

Ako korisnik želi posjetiti neki od profila registriranih u aplikaciji, to može učiniti klikom na neki od ponuđenih profila u *NewsFeed* aktivnosti. Klikom na profil otvara se aktivnost *VisitProfile*. Kao što je prije spomenuto, klikom na profil u novu aktivnost se prosljeđuju podaci koji su potrebni za pravilan rad aplikacije, što je prikazano u kodu koji slijedi. U ovom slučaju to su ime korisnika, tip njegovog računa te e-mail adresa.

```
Intent intent = getIntent();
userName = intent.getStringExtra(EXTRA_NAME);
userType = intent.getStringExtra(EXTRA_TYPE);
userEmail = intent.getStringExtra(EXTRA_EMAIL);
```

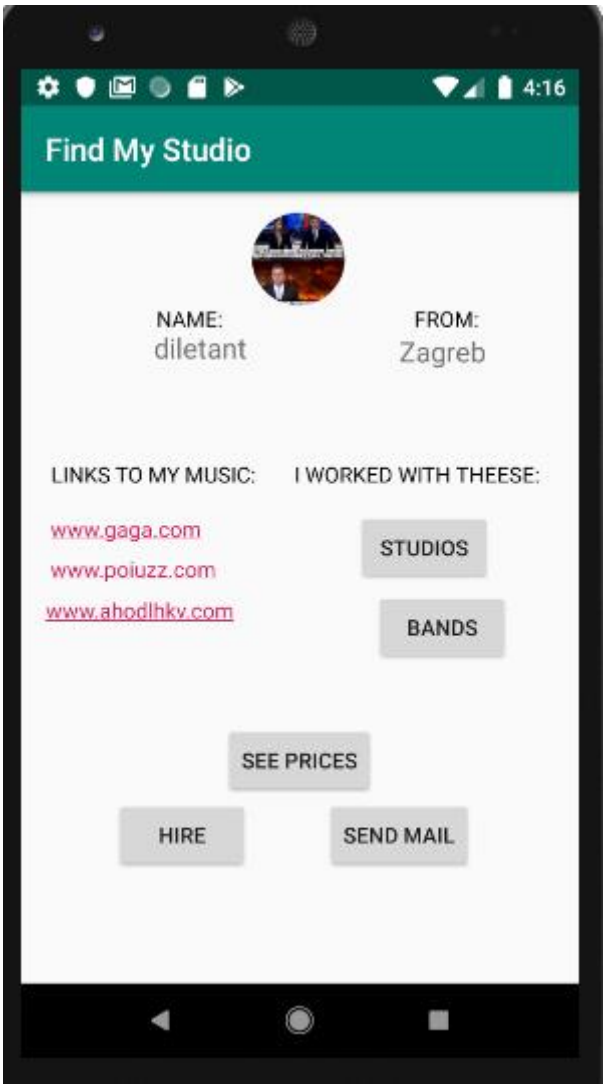
Pomoću imena korisnika traže se svi njegovi podaci iz baze podataka. Tip računa koristi za pravilno prikazivanje cijena odnosno profil bez cijena. Ako je posjećeni profil tipa *Band* ne prikazuju se gumbovi *See Prices* i *Hire*. Ista dva gumba ne prikazuju se ako se posjećuje isti tip profila kao kod korisnika. Svaki posjećeni profil ima portfolio sekciju, ali ona varira ovisno o tipu profila. Kod profila tipa *Band* mogu se vidjeti korisnikove suradnje s korisnicima profila tipa *Producer* i *Studio*. Kod tipa *Producer* mogu se vidjeti suradnje s korisnicima profila tipa *Band* i *Studio*. Kod tipa *Studio* mogu se vidjeti suradnje s korisnicima profila tipa *Band* i *Producer*. E-mail adresa bit će proslijeđena dalje u aktivnost *SendMail*, koja omogućava korisniku komunikaciju s korisnikom posjećenog profila. U kodu koji slijedi prikazan je uvjet kojim se opcija *SeePrices* uklanja ako korisnik posjećuje profil tipa *Band* te ako posjećuje isti tip profila kojim je i sam korisnik prijavljen.

```
if(userType.equals("Band")){
    View b = findViewById(R.id.btn_seePrices);
    b.setVisibility(View.GONE);

    View a = findViewById(R.id.btn_hire);
    a.setVisibility(View.GONE);
}

if(myType.equals(userType)) {
    View b = findViewById(R.id.btn_hire);
    b.setVisibility(View.GONE);
    View a = findViewById(R.id.btn_seePrices);
    a.setVisibility(View.GONE);
}
```

Na slici 13. prikazan je izgled aktivnosti *VisitProfile* za tip računa *Producer*.



Slika 13. Aktivnost VisitProfile

Klikom na gumb *See Prices* otvara se aktivnost koja prikazuje cijene pregledanog profila. Izgled te aktivnosti sličan je onom kod postavljanja cijena, ali korisnik koji pregledava cijene ne može pisati ili mijenjati podatke.

4.3.7. Opis aktivnosti SendMail

Ako korisnik klikne na gumb *Send Mail* otvara se nova aktivnost. U tu aktivnost prosljeđuju se podaci e-mail adrese korisnika čiji se profil pregledava.

```
sendMail.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(VisitProfile.this, SendMail.class);
        intent.putExtra("userEmail", userEmail);
        startActivity(intent);}});
```

Aktivnost *SendMail* sastoji se od tri mjesta za pisanje i njihovih naslova. To su *„To:“* gdje se je upisana e-mail adresa primatelja koja se po potrebi može i mijenjati, *„Subject:“* gdje se upisuje naslov poruke, te *„Message:“* gdje korisnik može napisati tekstualnu poruku.

Kada je korisnik napisao poruku potrebno je stisnuti gumb *Send*. Pritiskom na spomenuti gumb aplikacija traži od korisnika da odabere s koje će usluge elektronske pošte poslati poruku, te cijeli sadržaj poruke kopira u tu uslugu npr. *Gmail*.

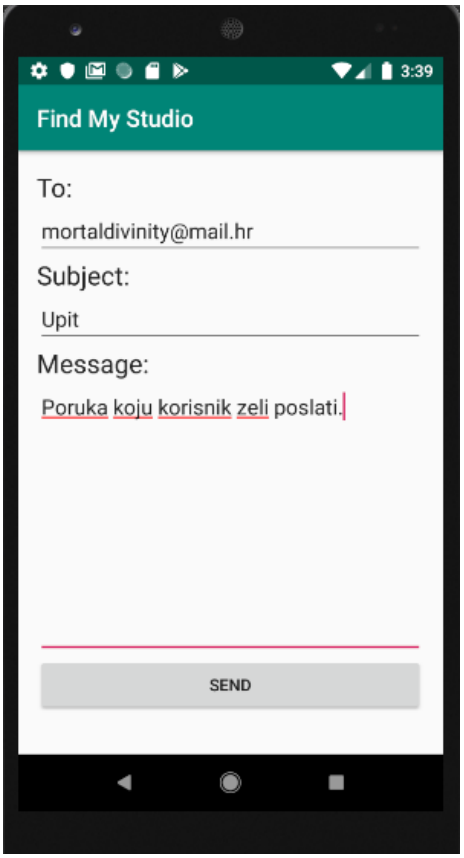
```
private void sendMail(){

    String recipient = to.getText().toString();
    String[] recipients = recipient.split(",");
    String Subject = subject.getText().toString();
    String Message = message.getText().toString();

    Intent intent = new Intent(Intent.ACTION_SEND);
    intent.putExtra(Intent.EXTRA_EMAIL, recipients);
    intent.putExtra(Intent.EXTRA_SUBJECT, Subject);
    intent.putExtra(Intent.EXTRA_TEXT, Message);

    intent.setType("message/rfc822");
    startActivity(Intent.createChooser(intent, "Choose an email client"));
}
```

Izgled aktivnosti SendMail može se vidjeti na slici 14.



Slika 14. Aktivnost SendMail

4.2.8. Opis aktivnosti Hire

Kada korisnik pritisne gumb *Hire* otvara se aktivnost *Calendar*. U ovoj aktivnosti nalaze se dva tekstualna polja u kojima korisnik može izabrati dva datuma, početni i završni datum suradnje.

Klikom na tekstualno polje otvara se kalendar gdje korisnik može izabrati željene datume. Kada se kalendar otvori, izbor je postavljen na trenutni datum. U ovoj metodi postavljeno je nekoliko uvjeta. Prvi uvjet je da početak suradnje ne može biti raniji datum od trenutnog: Drugi uvjet je da kraj suradnje ne može biti prije odabranog početnog datuma suradnje. Ti su uvjeti postavljeni u PHP skripti čiji dio slijedi u nastavku.

```
if((strtotime($ts) > strtotime($timestamp)) && (strtotime($timestamp) >= strtotime($today))){

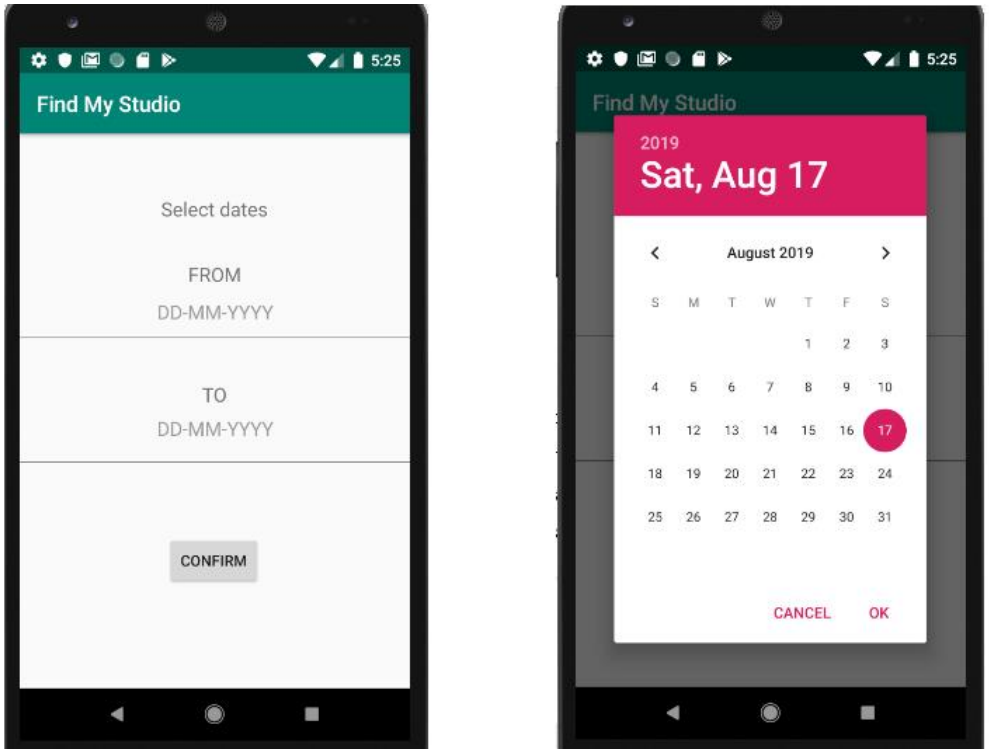
            $sql = "INSERT INTO hires_ps (ps_suraduju_od, ps_suraduju_do, id_producer, id_studio) VALUES ('$timestamp', '$ts', (SELECT id FROM producer
```

```
WHERE producer_name = '$name'), (SELECT id FROM studio WHERE studio_name =
'$myname'))";
```

U nastavku može se vidjeti kod klase *Calendar*, te slika 15 u kojoj se vidi izgled iste.

```
mDateOneEditText.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        mCurrentDate = java.util.Calendar.getInstance();
        int year = mCurrentDate.get(java.util.Calendar.YEAR);
        int month = mCurrentDate.get(java.util.Calendar.MONTH);
        int day = mCurrentDate.get(java.util.Calendar.DAY_OF_MONTH);

        DatePickerDialog mDatePicker = new DatePickerDialog(Calendar.this,
new DatePickerDialog.OnDateSetListener() {
            @Override
            public void onDateSet(DatePicker view, int year, int month, int
dayOfMonth) {
                mDateOneEditText.setText(dayOfMonth + "-" + (month+1) + "-"
+ year);
                mCurrentDate.set(year, month, dayOfMonth);
            }
        }, year, month, day);
        mDatePicker.show();
    }
});
```



Slika 15. Aktivnost Calendar

5. ZAKLJUČAK

U diplomskom radu je izrađena aplikacija *Find My Studio*, usluga za povezivanje i ostvarivanje suradnje između glazbenika, producenata i studija na operacijskom sustavu Android. Tijekom razvoja aplikacije usvojena su znanja objektno orijentiranog programiranja, rad na Android Studio platformi, te korištenja različitih programskih alata i tehnologija. Iako je zadatak bio zahtjevan, bio je jako poučan i zanimljiv. Zahvaljujući velikoj podršci na internetskim forumima poput *stackoverflow.com* omogućen je uvid u već postojeće aplikacije i primjere koda koji su uvelike pomogli razviti ovu aplikaciju.

Android studio pokazao se kao snažan i jednostavan alat za razvoj mobilnih aplikacija za operacijski sustav Android. Uz aktivna poboljšanja razvojne okoline za pisanje aplikacija i programskih biblioteka, Android predstavlja operacijski sustav koji obećava ostaviti značajni trag u svijetu razvoja programske podrške.

Razvoj *Find My Studio* aplikacije doveo je do razvoja novih ideja koje će značajno poboljšati aplikaciju u budućnosti. U aplikaciju bi se moglo, osim producenata, studija i glazbenika, uvesti i ostatak glazbene industrije kao što su izdavačke kuće, booking agenti, pa i ostali umjetnici poput grafičkih dizajnera za omote albuma ili izrade promotivnog materijala.

LITERATURA

Android (2014) Android Tutorial Horizon <https://android.tutorialhorizon.com/android-system-architecture/> Pristupljeno 3. kolovoza 2019.

AndroidAuthority (2019) <https://www.androidauthority.com/history-android-os-name-789433/> Pristupljeno 3. kolovoza 2019.

Developers (2019) <https://developer.android.com/studio/intro/studio-config> Pristupljeno 4.kolovoza 2019.

Edureka (2019), <https://www.edureka.co/blog/what-is-java/> Pristupljeno 4. kolovoza 2019.

Friesen, J. (2014) Learn Java for Android Development (third edition), Apress

Gargenta, M. (2011) Learning Android, Sebastopol: O'Reilly Media, Inc.

JavaPoint (2019) <https://www.javatpoint.com/mysql-history> Pristupljeno 4. kolovoza 2019.

Mulder (2019) <https://www.londonbuspal.co.uk/blog/wp-content/uploads/2019/01/sidebyside-1024x318.png> Pristupljeno 7. kolovoza 2019.

Pavlić, M. (2011) Oblikovanje baza podataka, Udžbenici sveučilišta u Rijeci

Pavlić, M. (2009) Informacijski sustavi, Odjel za informatiku Sveučilišta u Rijeci

PHP (2019), <https://www.php.net/manual/en/history.php.php> Pristupljeno 4. kolovoza 2019.

Smyth, N. (2019) Android Studio 3.4 Development Essentials Java Edition

TutorialsPoint (2019) <https://www.tutorialspoint.com/android/images/architecture.jpg> Pristupljeno 7. kolovoza 2019.

Unixman (2015) <https://www.unixmen.com/the-history-of-android/> Pristupljeno 3. kolovoza 2019.

POPIS SLIKA

Slika 1. Prikaz udjela mobilnih operacijskih sustava na tržištu (Wikipedia, 2017) 3
Slika 2. Prikaz udjela Android verzija na tržištu 2018. i 2019. godine (Mulder, 2019) 5
Slika 3. Android arhitektura (TutoriansPoint, 2019) 8
Slika 4. Prikaz korisničkog sučelja Android studija 10
Slika 5. Dijagram entiteta i veza baze podataka FMS 16
Slika 6. Prikaz datoteka u Android studiju 20
Slika 7. Aktivnost registracije 22
Slika 8. Aktivnost prijave 24
Slika 9. Aktivnost Settings 27
Slika 10. Aktivnost UpdateLinks 28
Slika 11. Aktivnost ProducerPrices (lijevo) i StudioPrices (desno) 29
Slika 12. Aktivnost NewsFeed 31
Slika 13. Aktivnost VisitProfile 33
Slika 14. Aktivnost SendMail 35
Slika 15. Aktivnost Calendar 36

POPIS TABLICA

Tablica 1. Koncepti dijagrama entiteti – veze……13

PRILOZI

1. CD – ROM sa izrađenom Android aplikacijom