Elektrostrojarska škola Varaždin  
Hallerova aleja 5, 42000, Varaždin

Tema:

**IZRADA ANDROID APLIKACIJE ZA SMARTPHONE POMOĆU ANDROID STUDIO ALATA**

ZAVRŠNI RAD

Mentor: Dubravko Tomasović Učenik: Valentin Mravlinčić

Smjer: tehničar za računalstvo

Šk. god: 2021./2022.

Varaždin, svibanj 2022.

**Sadržaj**

**1. Uvod1**

1.1 Trend mobilnih aplikacija1

1.2 Android operacijski sustav1

1.3 Pojam mobilnih aplikacija2

1.4 Android Studio alat3

1.5 Java programski jezik4

1.6 Extensible Markup Language (XML)4

1.7 Firebase4

**2. Android aplikacija Kave5**

2.1 Ideja za izradu5

2.2. Sastav aplikacije Kave7

2.2.1 Zaslon za prijavu i registraciju 8

2.2.2 Glavni zaslon 11

2.2.3 Zaslon košarica 15

**3. Zaključak17**

**Literatura18**

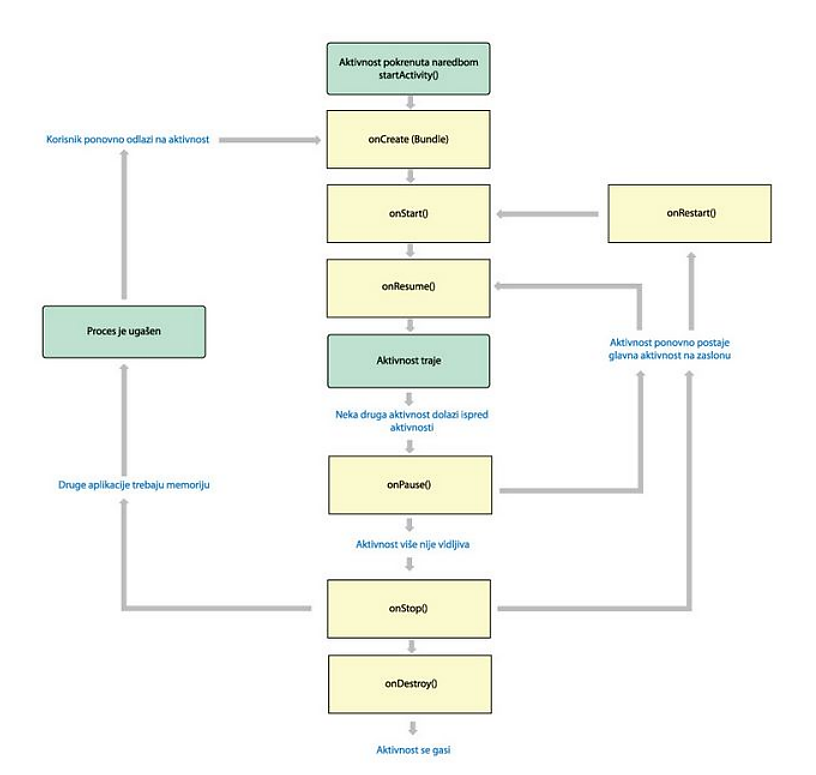
**1. Uvod**

**1.1 Trend mobilnih aplikacija**

U današnje vrijeme mobilne aplikacije postale su neophodne u našem životu. Prije su imale samo osnovne funkcije poput slanja poruka, a danas su se one znatno povećale. Mobilne aplikacije koriste gotovo sve osobe bez obzira na životnu dob. Trend mobilne tehnologije u konstantnom je rastu od njezinog začetka 1980-ih godina. Da bismo razumjeli temu ovog završnog rada, moramo znati osnovne pojmove vezane uz mobilne aplikacije.

**1.2 Android operacijski sustav**

Pojam „Android“ označava operacijski sustav temeljen na Linux jezgri koji je namijenjen izvršavanju na mobilnim uređajima. Android je nastao 2003. godine sa svrhom da omogući korištenje funkcija osobnog računala na mobilnim uređajima, a od tada do danas, najkorišteniji je mobilni operacijski sustav. Bitno je napomenuti da je Android operacijski sustav otvorenog koda, što znači da svatko može vidjeti i uređivati izvorni kod Androida. Android aplikacije razvijaju se u objektno orijentiranim programskim jezicima, a najpoznatiji jest Java. U ovom operacijskom sustavu svaka aplikacija ima svoj životni ciklus, a događaji tog ciklusa su sljedeći: onCreate(Bundle), onStart(), onPause(), onResume(), onStop(), onRestart() i onDestroy(). Tijek odvijanja tih događaja jasno je prikazan na sljedećem dijagramu tijeka aktivnosti.



Sl. 1 Životni ciklus aktivnosti

**1.3** **Pojam mobilnih aplikacija**

Mobilne aplikacije možemo predočiti kao programe na osobnim računalima koji su prilagođeni za mobilne uređaje. Cilj aplikacija je omogućiti i olakšati korisnicima izvršavanje i automatizaciju raznih zadataka. Karakteristika Android aplikacija je ravnopravnost, što znači da ne postoji razlika između osnovnih jezgrenih aplikacija i aplikacija koje korisnik instalira. Još jedna bitna karakteristika je automatsko upravljanje životnim ciklusom aplikacije. To podrazumijeva automatsko pokretanje, izvršavanje i gašenje procesa aplikacije prilikom pokretanja drugih, bez intervencije korisnika. Android uređaji javljaju se u različitim veličinama; pametni telefoni, tableti, čitači e-knjiga, laptopi, pametni televizori… Trenutni broj dostupnih aplikacija za preuzimanje na Trgovini Play je nešto više od tri milijuna.



Sl. 2 Inačice Android operacijskog sustava do danas

**1.4 Android Studio alat**

Android Studio alat je programsko okruženje koje se koristi za razvoj Android aplikacija u Java ili Kotlin programskim jezicima. Prva stabilna verzija objavljena je u prosincu 2014. godine s oznakom verzije 1.0. Android Studio temelji se na JetBrain's IntelliJ IDEA okruženju i predstavlja primarno razvojno okruženje za razvoj mobilnih Android aplikacija nakon što je zamijenio Eclipse. Android Studio omogućuje nam kreiranje korisničkog sučelja pomoću grafičkog sučelja, bez potrebe pisanja koda. Također, vrlo lako se povezuje s Google Cloud platformom, uz Android Virtual Device imamo mogućnost pokretanje aplikacije na virtualnom uređaju i podršku za Gradle build sustav. Trenutna verzija Android Studia je 4.2, a minimalne specifikacije koje naše računalo mora imati da bi pokrenuli Android Studio su: 64-bitni operacijski sustav, 8GB RAM-a, 8GB memorije za pohranu i zaslon s minimalnom rezolucijom od 1280x800 piksela.

Slika na kojoj se prikazuje tekst, snimka zaslona, monitor, računalo

Opis je automatski generiran

Sl. 3 Sučelje Android Studia

**1.5 Java programski jezik**

Tvrtka Sun Microsystems početkom 1990-ih razvila je Javu; objektno orijentirani programski jezik. Java je bio jedan od najkorištenijih programskih jezika zbog jednostavne sintakse i kontrole memorije, ali i sličnosti zbog toga jer je baziran na C++-u. Danas Javu polako zamjenjuju Kotlin i Dart programski jezici zbog brzine izvršavanje i novih mogućnosti.

**1.6 Extensible Markup Language (XML)**

XML predstavlja jezik za označavanje podataka, vrlo je sličan HTML-u. Glavni cilj mu je bio da bude jednostavan za čitanje i razumijevanje ljudima. XML u Android Studiu ima primjenu kod stvaranja vizualnog izgleda aplikacije to jest pomoću njega definiramo korisničko sučelje. XML definira sljedeće stavke: boje u aplikaciji, pozadine, vrstu teksta, gumbove, izgled i veličinu fonta… Svaki sadržaj započinje < i završava /> oznakom.

**1.7 Firebase**

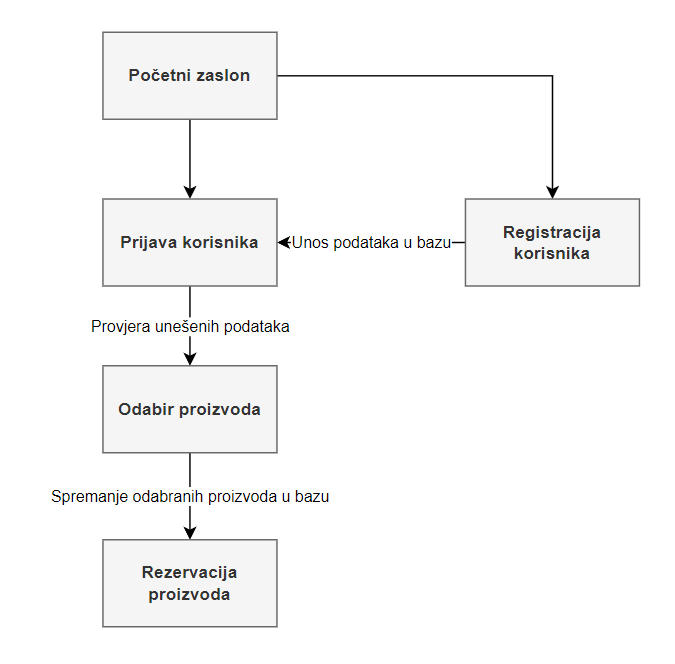
Firebase je platforma koju je 2014. godine otkupio Google, a danas nudi razne usluge vezane uz razvoj aplikacija, kao što su: Firebase Analytics, Firebase Realtime Database, Firebase Storage, Firebase Hosting… U ovom završnom radu koristio sam Firebase Realtime Database - bazu podataka koja se ažurira i sinkronizira u realnom vremenu na svim uređajima. Firebase se u aplikaciju dodaje vrlo jednostavno jer je on već integrirani alat u Android Studiu. Jedini uvjet da bismo mogli koristiti Firebase jest da imamo Google korisnički račun.

**2. Android aplikacija *Kave***

**2.1 Ideja za izradu**

Kako u današnje vrijeme gotovo svi koriste pametne telefone, stvorila se ideja o izradi aplikacije koja će biti korisna, a kroz samu izradu proširili bi svoje znanje i utvrdili postojeće što se tiče područja izrade mobilnih aplikacija. Glavni cilj ove aplikacije jest omogućiti korisnicima narudžbu određenog proizvoda bilo kada i iz bilo kojeg mjesta. Takva metoda naručivanja odgovara mnogim kupcima jer samim naručivanjem unaprijed štede mnogo slobodnog vremena i ne moraju čekati svoju narudžbu u gužvi.

Prvi korak u aplikaciji jest prijava putem e-mail adrese i lozinke ili registracija korisnika ako još nema izrađeni profil. Prilikom registracije potrebno je unijeti ime i prezime, broj godina, e-mail adresu i lozinku. Nakon prijave, otvara se izbornik sa svim dostupnim proizvodima i informacije o svakomu. Posljednji korak je provjera sadržaja u košarici zajedno s ukupnom cijenom te rezervacija sadržaja košarice. Slijedni prikaz odvijanja ove aplikacije prikazan je dijagramom na sljedećoj slici.



Sl. 4 Slijedni prikaz aplikacije

**2.2. Sastav aplikacije *Kave***

Slika na kojoj se prikazuje tekst

Opis je automatski generiran

Sl. 5 Sadržaj aplikacije

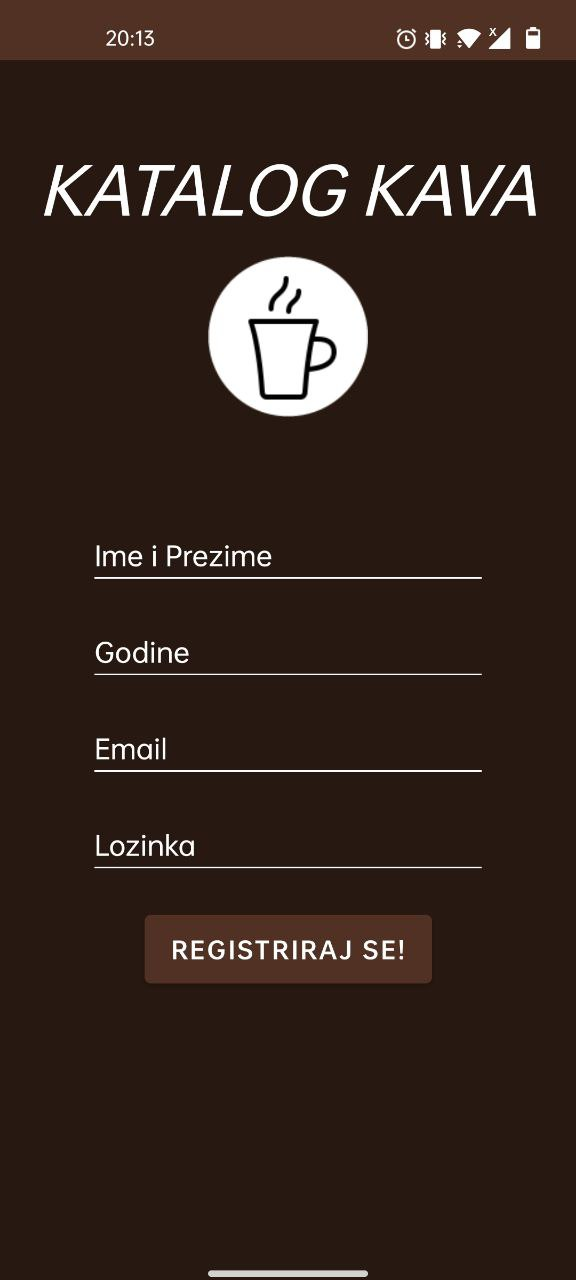
U mapi „manifest“ nalazi se AndroidManifest.xml datoteka koja propisuje sve dozvole, aktivnosti, ikonu i verziju aplikacije te samu struktura aplikacije. Mapa „java“ sadrži deset klasa (adaptere, modele i glavne aktivnosti) te tri sučelja za tzv. „listenere“ tj. osluškivače. Sve datoteke iz „java“ mape napisane su u Javi. Posljednja bitna mapa jest „res“ u kojoj se nalaze datoteke napisane u XML programskom jeziku koje određuju izgled korisničkog sučelja i rasporede ostalih elemenata u aplikaciji.

**2.2.1 Zaslon za prijavu i registraciju**



Sl. 6 Zaslon za prijavu u aplikaciju

Prvi zaslon koji se otvori nakon pokretanja aplikacije je zaslon (activity\_login.xml) na kojem se korisnik mora prijaviti pomoću podataka koje je unio prilikom kreiranja računa ili ako ga još nije kreirao, mora se registrirati pritiskom na tipku *REGISTRACIJA* koji će ga odvesti na sljedeći zaslon.



Sl. 7 Zaslon za registraciju

Nakon što se otvori zaslon za registraciju (activity\_register.xml) korisnik mora ispuniti sva polja s pravilnim podacima kako bi se uspješno registrirao. Forma za podatke korisnika definirana je u User.java modelu. Na ovom zaslonu provjerava se i ispravnost unesenih podataka, pa tako ako ostavimo prazno polje, pojavit će nam se greška s informacijom. Također će nam se pojaviti greška ako polje namijenjeno unosu e-mail adrese ispunimo s nepravilnom formom i ako polje za unos lozinke ima manje od 6 znakova. Posljednja provjera prije registracije jest provjera postoji li korisnički račun s istom e-mail adresom, no ako su svi uvjeti zadovoljeni, dobivamo informaciju o uspješnoj registraciji nakon pritiska na tipku *REGISTRIRAJ SE!* te se možemo vratiti na zaslon za prijavu. Ta provjera obavlja se u RegisterUser.java datoteci, sljedećim programskim kodom:

private void registerUser() {

//pretvorba u string varijable za pretrazivanje

String email=editTextEmail.getText().toString().trim();

String password=editTextPassword.getText().toString().trim();

String fullName=editTextFullName.getText().toString().trim();

String age=editTextAge.getText().toString().trim();

//provjera polja za ime i prezime

if(fullName.isEmpty()){

editTextFullName.setError("Ime i prezime je obavezno!");

editTextFullName.requestFocus();

return;}

//provjera polja za godine

if(age.isEmpty()){

editTextAge.setError("Godine su obavezne!");

editTextAge.requestFocus();

return;}

//provjera polja za email

if(email.isEmpty()){

editTextEmail.setError("Email je obavezan!");

editTextEmail.requestFocus();

return;}

if(!Patterns.EMAIL\_ADDRESS.matcher(email).matches()){

editTextEmail.setError("Unesite ispravnu email adresu!");

editTextEmail.requestFocus();

return; }

//provjera polja za lozinku

if(password.isEmpty()){

editTextPassword.setError("Lozinka je obavezna!");

editTextPassword.requestFocus();

return;}

if(password.length() < 6){

editTextPassword.setError("Lozinka mora sadržavati najmanje 6 znakova!");

editTextPassword.requestFocus();

return; }

U pozadini aplikacije u tom trenutku podaci o korisniku se zapisuju u bazu podataka pomoću mAuth metode kako bi kasnije mogli provjeriti autentičnost korisnika:

mAuth.createUserWithEmailAndPassword(email, password)

Prilikom prijave, nakon što korisnik unese podatke i pritisne gumb *PRIJAVA*, podaci se uspoređuju s onima u bazi podataka, također pomoću mAuth metode, te ako se e-mail adresa podudara s lozinkom iz baze podataka, uspješno će se otvoriti glavni izbornik aplikacije. U programskom kodu to izgleda ovako:

mAuth.signInWithEmailAndPassword(email, password).addOnCompleteListener(new OnCompleteListener<AuthResult>()

**2.2.2 Glavni zaslon**



Sl. 8 Glavni zaslon za biranje kava

Kada se otvori glavni zaslon aplikacije (activity\_main.xml), možemo vidjeti izbor dostupnih proizvoda u aplikaciji, u ovom slučaju su to kave. Svaka kava ima svoj opis; ime, količinu, sadržaj i cijenu. Ova aktivnost većim dijelom definirana je u MainActivity.java i MyDrinkAdapter.java datoteci, a kartica s rasporedom definirana je u DrinkModel.java datoteci pomoću gettera i settera:

public DrinkModel() {

}

public String getKey() {

return key;

}

public void setKey(String key) {

this.key = key;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getImage() {

return image;

}

public void setImage(String image) {

this.image = image;

}

public String getPrice() {

return price;

}

public void setPrice(String price) {

this.price = price;

}

public String getmL() {

return mL;

}

public void setmL(String mL) {

this.mL = mL;

}

public String getDescription() {

return description;

}

public void setDescription(String description) {

this.description = description;

}}

Kavu dodajemo u košaricu pritiskom na bilo koje mjesto kartice za tu kavu. Ta funkcija implementirana je sljedećim programskim kodom:

private void addToCart(DrinkModel drinkModel) {

//dolazak do direktorija u bazi podataka

DatabaseReference userCart = FirebaseDatabase

.getInstance().getReference("Cart")

.child("UNIQUE\_USER\_ID");

userCart.child(drinkModel.getKey())

.addListenerForSingleValueEvent(new ValueEventListener() {

@Override

public void onDataChange(@NonNull DataSnapshot snapshot) {

if(snapshot.exists()){

//ako je stvar u kosarici azuriramo kolicinu i ukupnu cijenu

CartModel cartModel = snapshot.getValue(CartModel.class);

cartModel.setQuantity(cartModel.getQuantity()+1);

Map<String,Object> updateData = new HashMap<>();

updateData.put("quantity", cartModel.getQuantity());

updateData.put("totalPrice", cartModel.getQuantity()\*Float.parseFloat(cartModel.getPrice()));

userCart.child(drinkModel.getKey())

.updateChildren(updateData)

.addOnSuccessListener(aVoid -> {

iCartLoadListener.onCartLoadFailed("Uspješno dodano u košaricu.");

})

.addOnFailureListener(e -> iCartLoadListener.onCartLoadFailed(e.getMessage()));

}

else

//ako stvar nije u kosarici dodajemo ju

{

CartModel cartModel = new CartModel();

cartModel.setName(drinkModel.getName());

cartModel.setImage(drinkModel.getImage());

cartModel.setKey(drinkModel.getKey());

cartModel.setPrice(drinkModel.getPrice());

cartModel.setQuantity(1);

cartModel.setTotalPrice(Float.parseFloat(drinkModel.getPrice()));

userCart.child(drinkModel.getKey())

.setValue(cartModel)

.addOnSuccessListener(aVoid -> {

iCartLoadListener.onCartLoadFailed("Uspješno dodano u košaricu.");

})

.addOnFailureListener(e -> iCartLoadListener.onCartLoadFailed(e.getMessage()));}

EventBus.getDefault().postSticky(new MyUpdateCartEvent());

}

@Override

public void onCancelled(@NonNull DatabaseError error) {

iCartLoadListener.onCartLoadFailed(error.getMessage()); }}); }

Nakon dodavanja kave, uvijek će se pojaviti obavijest je li uspješno dodana u košaricu. U traci na vrhu zaslona imamo gumb za odjavu iz aplikacije *ODJAVA* koji će nas vratiti na zaslon za prijavu, tekst s našim imenom i gumb za ulazak u košaricu. Sve informacije o kavama nalaze se u bazi podataka, koji se dohvaćaju, raspoređuju (layout\_drink\_item.xml) te prikazuju u karticama na glavnom zaslonu za svaku kavu posebno.

**2.2.3 Zaslon košarica**



Sl. 9 Zaslon košarica

Ako smo pritisnuli ikonu za košaricu na glavnom zaslonu, otvorit će nam se ovaj zaslon sa sadržajem košarice (activity\_cart.xml) u kojemu su kave raspoređene u kartice, kao na prošlom zaslonu (layout\_cart\_item.xml). Zaslon za košaricu definiran je u CartActivity.java, MyCartAdapter.java i CartMode.java datotekama. U ovoj kartici osim tekstnih informacija imamo tri gumba; gumb + i – za povećanje ili smanjivanje količine iste kave te gumb s ikonom kante za smeće za potpuno uklanjanje te kave iz košarice bez obzira na količinu. U traci na vrhu zaslona nalazi se zbroj ukupnog sadržaja košarice. Primjer izvedbe sume svih sadržaja u košarici:

@Override

public void onCartLoadSuccess(List<CartModel> cartModelList) {

double sum = 0;

//kada imamo stavke u kosarici, uzima se cijena i izracunava ukupna prema zadanoj cijeni iz CartModel.java datoteke

for(CartModel cartModel:cartModelList){

sum += cartModel.getTotalPrice(); }

//postavljanje ukupne cijene u txtTotal varijablu

txtTotal.setText(new StringBuilder().append(sum));

MyCartAdapter adapter = new MyCartAdapter(this, cartModelList);

recyclerCart.setAdapter(adapter); }

Ispod odabranih kava, nalazi se gumb *REZERVIRAJ* koji se nakon pritiska promijeni u *OTKAŽI REZERVACIJU* te počinje odbrojavati vrijeme do kada je rezervacija našeg sadržaje iz košarice valjana. Ukoliko želimo otkazati rezervaciju iz bilo kojeg razloga, to možemo učiniti jednim klikom na gumb *OTKAŽI REZERVACIJU*.

**Zaključak**

U današnje vrijeme velik broj ljudi koristi pametne mobilne uređaje kako bi si olakšali svakodnevne rutine i poslove. S tim ciljem, dobivena je ideja i potom izrađena aplikacija koja olakšava naručivanje proizvoda kako bi uštedjeli svoje slobodno vrijeme. Za izradu aplikacije bilo je potrebno osnovno znanje sintakse jezika potrebnih za izradu aplikacije u Android Studiu te upoznavanje samog tog programskog okruženja. Uz Android Studio korištena je Firebase baza podataka za pohranu podataka o korisnicima i proizvodima. Osim toga, bilo je potrebno shvatiti način programiranja kod objektno orijentiranog programskog jezika Java. Ovakva aplikacija korisna je za mala poduzeća koja nude svoje proizvode i usluge. U budućnosti, na aplikaciji mogle bi se implementirati dodatne funkcije poput sučelja za administratore, slanje potvrde o narudžbi na mail, plaćanje putem aplikacije…

**Literatura**

1. Java dokumentacija,

<https://docs.oracle.com/en/java/>, 22.3.2022

1. Firebase dokumentacija,

<https://firebase.google.com/docs>, 11.4.2022.

[3] Ivan Škorić, Osnove izrade aplikacija za operativni sustav Android,

<http://www.mathos.unios.hr/~mdjumic/uploads/diplomski/%C5%A0KO03.pdf>, 2012.