Izrada mobilne aplikacije za vođenje evidencije prisutnosti na nastavi

Nenadić, Petar

Master's thesis / Diplomski rad

2022

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:648327

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-05-14**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Sveučilišni studij

IZRADA MOBILNE APLIKACIJE ZA VOĐENJE EVIDENCIJE PRISUTNOSTI NA NASTAVI

Diplomski rad

Petar Nenadić

Osijek, 2022.

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Obrazac D1: Obrazac za imenovanje Povjerenstva za diplomski ispit

Osijek, 07.12.2022.

Odboru za završne i diplomske ispite

Imenovanje Povjerenstva za diplomski ispit

Ime i prezime Pristupnika:	Petar Nenadić
Studij, smjer:	Diplomski sveučilišni studij Računarstvo
Mat. br. Pristupnika, godina upisa:	D-1147R, 13.10.2020.
OIB studenta:	37876856856
Mentor:	Izv. prof. dr. sc. Damir Blažević
Sumentor:	Izv.prof.dr.sc. Tomislav Keser
Sumentor iz tvrtke:	
Predsjednik Povjerenstva:	Izv. prof. dr. sc. Ivan Aleksi
Član Povjerenstva 1:	Izv. prof. dr. sc. Damir Blažević
Član Povjerenstva 2:	Izv. prof. dr. sc. Ivica Lukić
Naslov diplomskog rada:	Izrada mobilne aplikacije za vođenje evidencije prisutnosti na nastavi
Znanstvena grana diplomskog rada:	**Programsko inženjerstvo (zn. polje računarstvo)**
Zadatak diplomskog rada:	Izraditi mobilnu aplikacija na klijent server arhitekturi koja će korištenjem suvremenih komunikacijskih tehnologija omogućiti evidenciju prisutnosti studenata na nastavi uz korištenje smartphone uređaja. (Tema rezervirana za: Petar Nenadić)
Prijedlog ocjene pismenog dijela ispita (diplomskog rada):	Izvrstan (5)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 3 bod/boda Jasnoća pismenog izražavanja: 2 bod/boda Razina samostalnosti: 3 razina
Datum prijedloga ocjene od strane mentora:	07.12.2022.

Potvrda mentora o predaji konačne verzije rada:	*Mentor elektronički potpisao predaju konačne verzije.*
	Datum:

SADRŽAJ

1. UVOD 1

1.1. Zadatak diplomskog rada 1

2. Pregled sličnih aplikacija 2

2.1. Presli – Attendance Manager 2

2.2. Alora – Attendance Tracker App 3

2.3. Self Attendance 4

2.4. QR Attendance Control 4

3. PREGLED UPOTRIJEBLJENIH TEHNOLOGIJA 6

3.1. Android Studio 6

3.2. Kotlin 6

3.3. XML 7

3.4. Firebase 8

3.5. Senzori i kamera 9

3.6. AES algoritam za enkripciju 10

4. RAZVOJ APLIKACIJE 11

4.1. Korisnički zahtjevi 11

4.2. Dijagram toka 12

4.3. Arhitektura sustava 14

5. IMPLEMENTACIJA PROGRAMSKOG RJEŠENJA 16

5.1. Dodavanje ovisnosti 16

5.2. Struktura projekta 16

5.3. Aplikacija za studente 20

5.4. Aplikacija za nastavnike 29

5.5. Testiranje aplikacija 39

6. Zaključak 40

LITERATURA 41

SAŽETAK **43**

ABSTRACT **44**

ŽIVOTOPIS **45**

PRILOZI **46**

1. UVOD

Evidentiranje prisutnosti na nastavi često zahtijeva puno vremena i sklono je ljudskim greškama i čestim bilježenjem netočnih podataka. Također, praćenje prisutnosti po studentu može biti zamoran proces zato što zahtijeva neku vrstu priručnika za praćenje dana prisutnosti. Pojavom pametnih uređaja, koji olakšavaju obavljanje nekih poslova, ovaj proces se može u potpunosti automatizirati korištenjem odgovarajuće tehnologije.

Glavni cilj ovog diplomskog rada je razviti aplikaciju za Android operacijski sustav koja će ispuniti današnje standarde i zahtjeve korisnika. Aplikacija omogućuje studentima da se prijavljuju na predavanje putem QR koda koji je prikazan na mobitelu nastavnika. Za izradu ovog rada koriste se znanja stečena tijekom studiranja na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija u Osijeku.

U drugom poglavlju navedene su aplikacije slične ovoj i opisane njihove prednosti i nedostaci. Treće poglavlje sadrži opis tehnologija korištenih za izradu aplikacije. Četvrto poglavlje sadrži opisane korisničke zahtjeve na aplikaciju, dijagram toka, te arhitekturu sustava. Peto poglavlje sadrži detaljno opisane upute za rad s aplikacijama.

1.1. Zadatak diplomskog rada

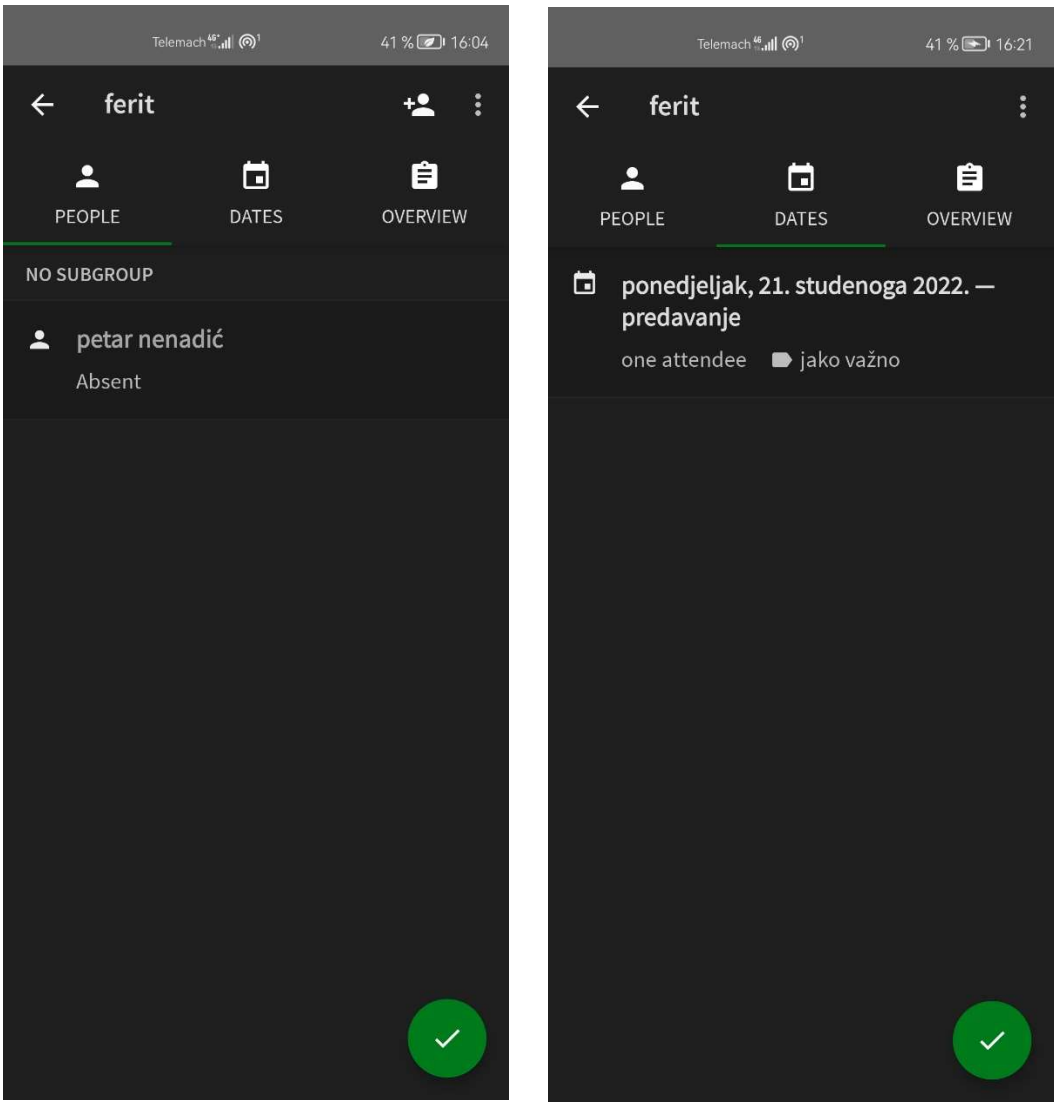
Izraditi mobilnu aplikaciju na klijent-server arhitekturi koja će korištenjem suvremenih komunikacijskih tehnologija omogućiti evidenciju prisutnosti studenata na nastavi uz korištenje smartphone uređaja.

2. Pregled sličnih aplikacija

U ovom poglavlju su navedene aplikacije koje su po funkcionalnostima slične aplikaciji izrađenoj za potrebe ovog rada.

2.1. Presli – Attendance Manager

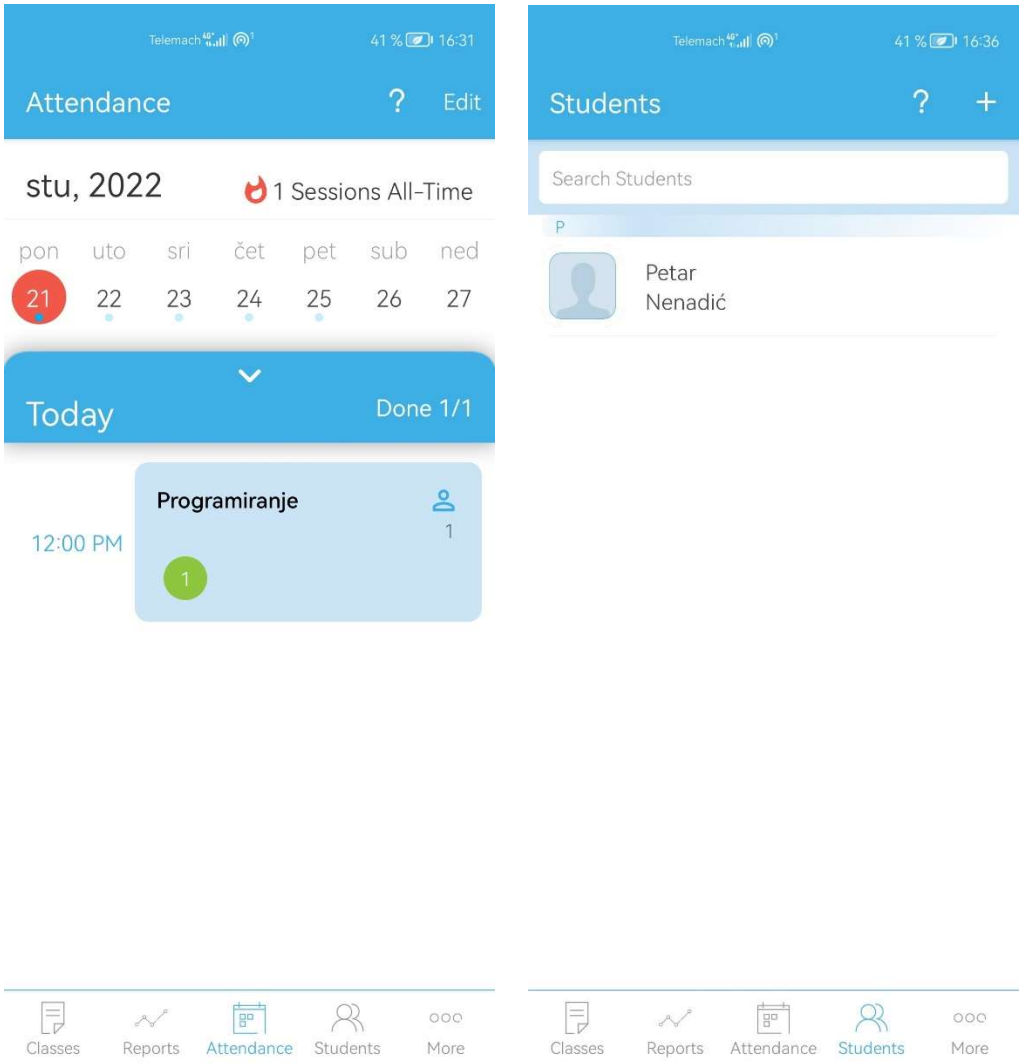
Aplikacija za upravljanje događajima, omogućuje dodavanje grupa, podgrupa, datuma događaja i bilješki za osobe koje prisustvuju događaju. Također, omogućuje sinkronizaciju podataka u oblaku te posjeduje alate za analizu posjećenosti događaja. Nakon svakog događaja, potrebno je upisivati novi datum kada će se taj događaj odvijati [1]. Na slici 2.1. prikazano je sučelje Presli aplikacije.



Sl. 2.1. Sučelje Presli aplikacije

2.2. Alora – Attendance Tracker App

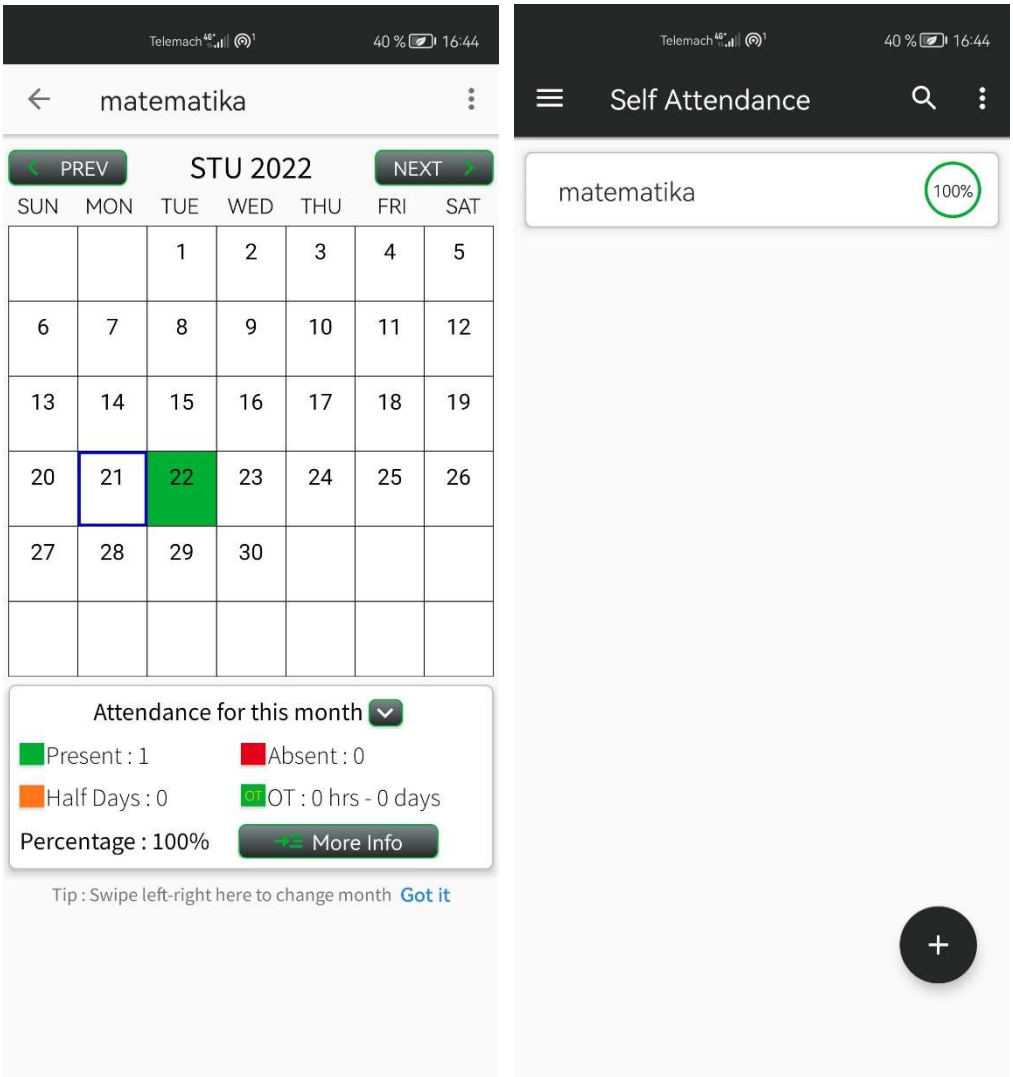
Aplikacija za praćenje koja pomaže uštedjeti vrijeme. Omogućuje postavljanje grupa, dodavanje studenata u grupe te bilješki. Aplikacija sadrži alate za statistiku, ali je potrebno preuzeti plaćenu verziju. Uglavnom je napravljena za nastavnike pa studenti ne mogu vidjeti prisutnost na nekom kolegiju. Nema mogućnost automatskog zabilježavanja prisutnosti, kao što je skeniranje QR koda, pa nastavnik mora prozivati studente kako bi zabilježio njihovu prisutnost [2]. Na slici 2.2. prikazano je sučelje aplikacije.



Sl. 2.2. Sučelje Alora aplikacije

2.3. Self Attendance

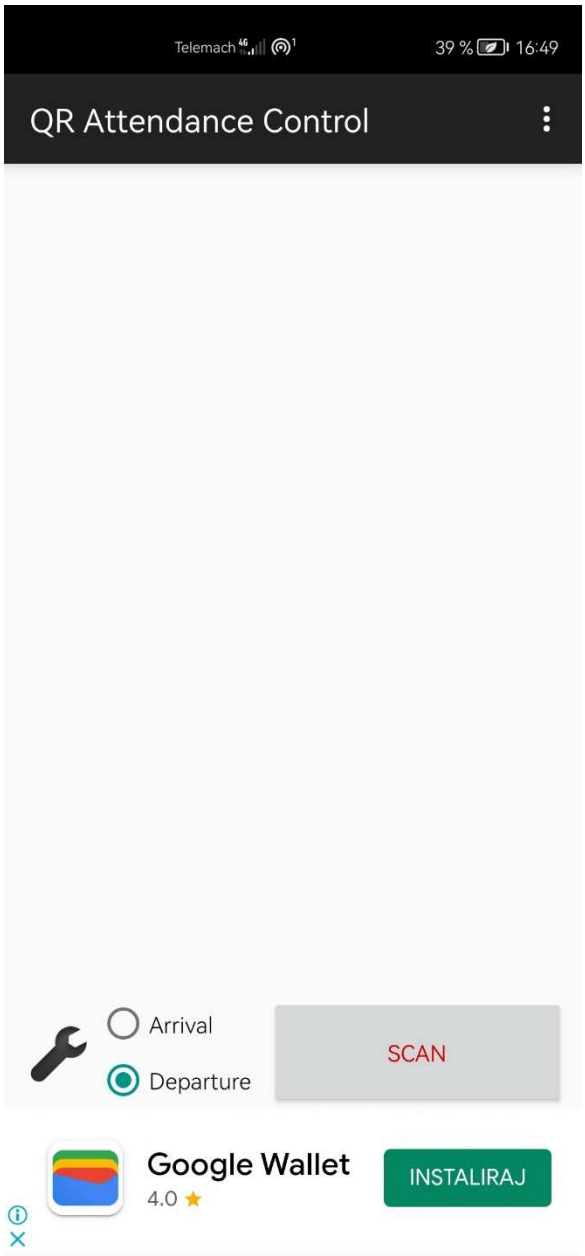
Aplikacija za praćenje dnevne prisutnosti. Aplikacija je uglavnom izrađena za studente, pa studenti sami moraju unositi kolegije u raspored i podatke o prisutnosti. Događajima se mogu dodati i bilješke, kao što je odgoda predavanja i slično. Podatke sprema u memoriju uređaja, te omogućuje sinkronizaciju s Google diskom [3]. Na slici 2.3. prikazano je sučelje aplikacije.



Sl. 2.3. Sučelje Self Attendance aplikacije

2.4. QR Attendance Control

Aplikacija koja omogućuje kontrolu prisutnosti na događajima, vrijeme dolaska i odlaska. Omogućuje izvoz popisa prisutnosti u CSV formatu. Za svaku prijavljenu osobu se kreira jedinstveni QR kod koji se koristi za prijavljivanje [4]. Na slici 2.4. prikazano je sučelje aplikacije.



Sl. 2.4. Sučelje QR Attendance Control aplikacije

3. PREGLED UPOTRIJEBLJENIH TEHNOLOGIJA

U ovom poglavlju opisane su tehnologije koje su upotrijebljene za razvoj aplikacije.

3.1. Android Studio

Android Studio je službeni IDE (engl. *Integrated Development Environment*) za razvoj aplikacija koje se pokreću na operativnom sustavu Android. Trenutno je najpopularnije razvojno okruženje za razvoj Android aplikacija te omogućuje trenutno pokretanje koda na uređaju ili emulatoru. Omogućuje podršku za sve pametne uređaje, kao što su: mobiteli, satovi, tableti, televizori i ostale uređaje s Android operacijskim sustavom.

Android Studio je dostupan za preuzimanje za Windows, MacOS i Linux operacijske sustave. Dolaskom na tržište zamijenio je Eclipse ADT (engl. Ada Development Tools) te nudi niz značajki koje poboljšavaju produktivnost, a neke od njih su:

- Razvojni alati
- Brz emulator
- Podrška za programski jezik C++
- Jedinstveno okruženje kompatibilno sa svim uređajima koji pogone Android operativni sustav

Budući da se za pisanje koda u Android Studiu koristi programski jezik Java, koji je ujedno i temelj za pisanje programskog koda za Android uređaje, potrebno je instalirati i najnoviju verziju *Java Development Kit*-a (JDK).

3.2. Kotlin

Kotlin je trenutno jedan od najmodernijih programskih jezika iako je započeo svoj razvoj 2011. godine. To je jezik koji se neprekidno razvija i stalno se unaprjeđuje novim mogućnostima i funkcionalnostima. Budući da uživa veliku podršku *Google*-a, očekuje se i daljnja podrška za Kotlin, zbog čega ga sve više programera koristi umjesto *Jave*. Može se koristiti i za razvoj *Web* i serverskih aplikacija.

Kotlin se koristi u ovom radu kako bi povezao sve elemente u aplikaciji i dodao im razne funkcionalnosti. Na slici 3.1. prikazan je primjer koda u programskom jeziku Kotlin

```
override fun onBackPressed() {
    val drawerLayout: DrawerLayout = findViewById(R.id.drawer_layout)
    if (drawerLayout.isDrawerOpen(GravityCompat.START)) {
        drawerLayout.closeDrawer(GravityCompat.START)
    } else {
        super.onBackPressed()
    }
}
```

Sl. 3.1. Primjer funkcije

3.3. XML

XML je jezik koji se koristi za označavanje podataka i dokumenata. Zamišljen je da podatke čuva u tekstualnom obliku i uokviruje ih oznakama kako bi se znalo o kakvim podacima se radi.

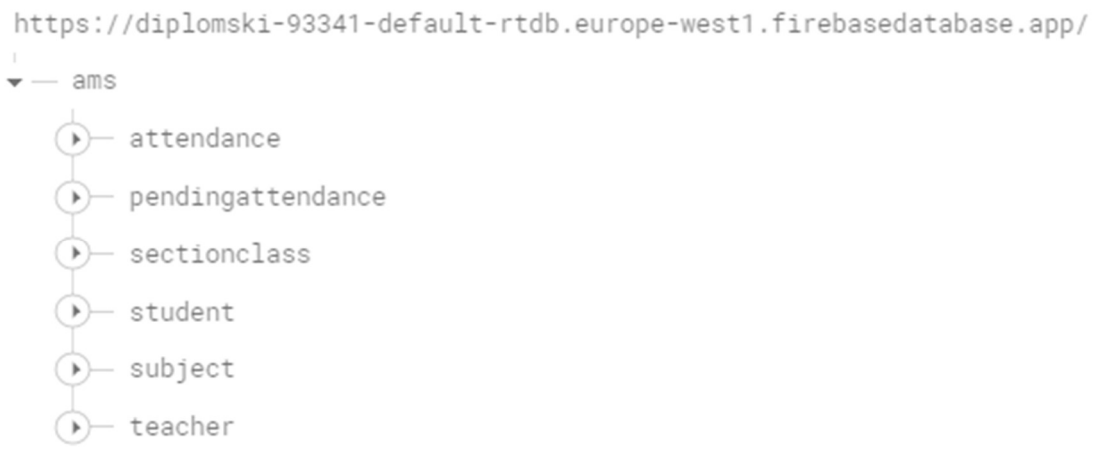
Iako je namjena XML-a prvenstveno za upotrebu u dokumentima, može se koristiti i za druge namjene, kao što su: razmjena podataka, odvajanje podataka od prezentacije, pohrana podataka i povećana dostupnost podataka. XML je standardizirani jezik, za što se pobrinuo World Wide Web Consortium. U ovom radu je korišten za kreiranje korisničkog sučelja. Slika 3.2. prikazuje primjer XML koda korištenog za razvoj aplikacija.

```xml
<?xml version="1.0" encoding="utf-8"?>
<androidx.drawerlayout.widget.DrawerLayout
        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"
        android:id="@+id/drawer_layout"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:fitsSystemWindows="true"
        tools:openDrawer="start">

    <include
            layout="@layout/app_bar_main2"
            android:layout_width="match_parent"
            android:layout_height="match_parent"/>

    <com.google.android.material.navigation.NavigationView
            android:id="@+id/nav_view"
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            android:layout_gravity="start"
            android:fitsSystemWindows="true"
            app:headerLayout="@layout/nav_header_main2"
            app:menu="@menu/activity_main2_drawer"/>

</androidx.drawerlayout.widget.DrawerLayout>
```

Sl. 3.2 Primjer XML koda

3.4. Firebase

Firebase je platforma istoimene tvrtke koja olakšava razvoj aplikacija koje koriste baze podataka. Podržava razvoj web i mobilnih aplikacija te omogućuje korisnicima mnoge servise. Neki od tih servisa su alati za analitiku, autentifikacija korisnika i baza podataka u stvarnom vremenu.



Sl. 3.3. Firebase tablica

Firebase se koristi u ovom radu zato što omogućava pohranu podataka u stvarnom vremenu i što su ti podaci dostupni svugdje u svijetu. Za komunikaciju s Firebase-om potrebni su autentifikacijski podaci i pristup Internetu.

3.5. Senzori i kamera

Gotovo svi pametni telefoni i tableti koji koriste Android operacijski sustav imaju ugrađeno po nekoliko vrsta senzora pomoću kojih je moguće dobiti podatke o trenutnoj lokaciji, brzini kretanja te ubrzanju. Razne kretnje uređaja, kao što su drmanje i položaj u prostoru, pokreću senzore za promatranje trodimenzionalnog položaja u prostoru. U ovom radu se koristi senzor pozicije koji koristi velik broj različitih metoda kako bi dostavio aplikaciji informacije o lokaciji uređaja. U Android Studiu, takve metode se nazivaju dostavljači lokacije (engl. *location providers*) i svaka se može koristiti na različite načine, ovisno o situaciji. Primjer dostavljača lokacije je GPS (engl. *Global Positioning System*) i mrežni dostavljači. Senzori lokacije koriste se u ovom radu kako bi se odredio položaj uređaja koji skeniraju QR kod.

Osim raznih senzora, većina pametnih uređaja ima ugrađene kamere koje mogu poslužiti u razne svrhe, kao što je snimanje videa ili fotografiranje. U Android Studiu, pristup kameri se omogućuje dopuštenjem koje se mora upisati u *Manifest* datoteku, a to dopuštenje je:

<uses-permission android:name="android.permission.CAMERA"/>.

Senzor kamere se koristi u ovom radu kako bi se očitao QR kod prilikom prijave studenta na predavanja.

3.6. AES algoritam za enkripciju

Kako bi se osigurala jedinstvenost podataka, za enkripciju zaporki korišten je AES (engl. *Advanced Encryption Standard*) algoritam za enkripciju. U ovom radu koristi se takozvani Rijndael sustav kojeg je uspostavio Nacionalni institut za standarde i tehnologiju SAD-a (NIST) u 2001. godini. Rijndaelovu blok šifru su razvila dva belgijska kriptografa, Joan Daemen i Vincent Rijmen [5]. Karakteristike AES standarda su sljedeće:

- Šifriranje simetričnim ključem i simetričnim blokom
- 128-bitni podaci, 128/192/256-bitni ključevi
- Bolja enkripcija i brži od Triple DES-a
- Softver koji se može implementirati u C programskom jeziku i Javi
- Enkriptiranje korisničkih zaporki i podataka o QR kodu

Slika 3.4. prikazuje funkciju za enkripciju zaporki pomoću AES standarda.

```
fun String.decrypt(password: String): String {
    val secretKeySpec = SecretKeySpec(password.toByteArray(), "AES")
    val iv = ByteArray(16)
    val charArray = password.toCharArray()
    for (i in charArray.indices) {
        iv[i] = charArray[i].code.toByte()
    }
    val ivParameterSpec = IvParameterSpec(iv)

    val cipher = Cipher.getInstance("AES/GCM/NoPadding")
    cipher.init(Cipher.DECRYPT_MODE, secretKeySpec, ivParameterSpec)

    val decryptedByteValue = cipher.doFinal(Base64.decode(this, Base64.DEFAULT))
    return String(decryptedByteValue)
}
```

Sl. 3.4. Primjer funkcije u programskom jeziku Kotlin

4. RAZVOJ APLIKACIJE

U ovom poglavlju su opisani korisnički zahtjevi na aplikaciju, dijagrami toka za aplikaciju za nastavnike i studente te arhitektura sustava.

4.1. Korisnički zahtjevi

Prije početka izrade same aplikacije potrebno je postaviti korisničke zahtjeve. Aplikacija služi za prijavljivanje studenata na nastavu i ima dvije vrste korisnika: nastavnike i studente. Budući da postoje dvije vrste korisnika, svaki ima svoju aplikaciju, odnosno, izrađene su dvije aplikacije. Jedna aplikacija je za nastavnike, a druga za studente. Svaka od aplikacija ima svoje zahtjeve.

Aplikacija za studente ima sljedeće zahtjeve:

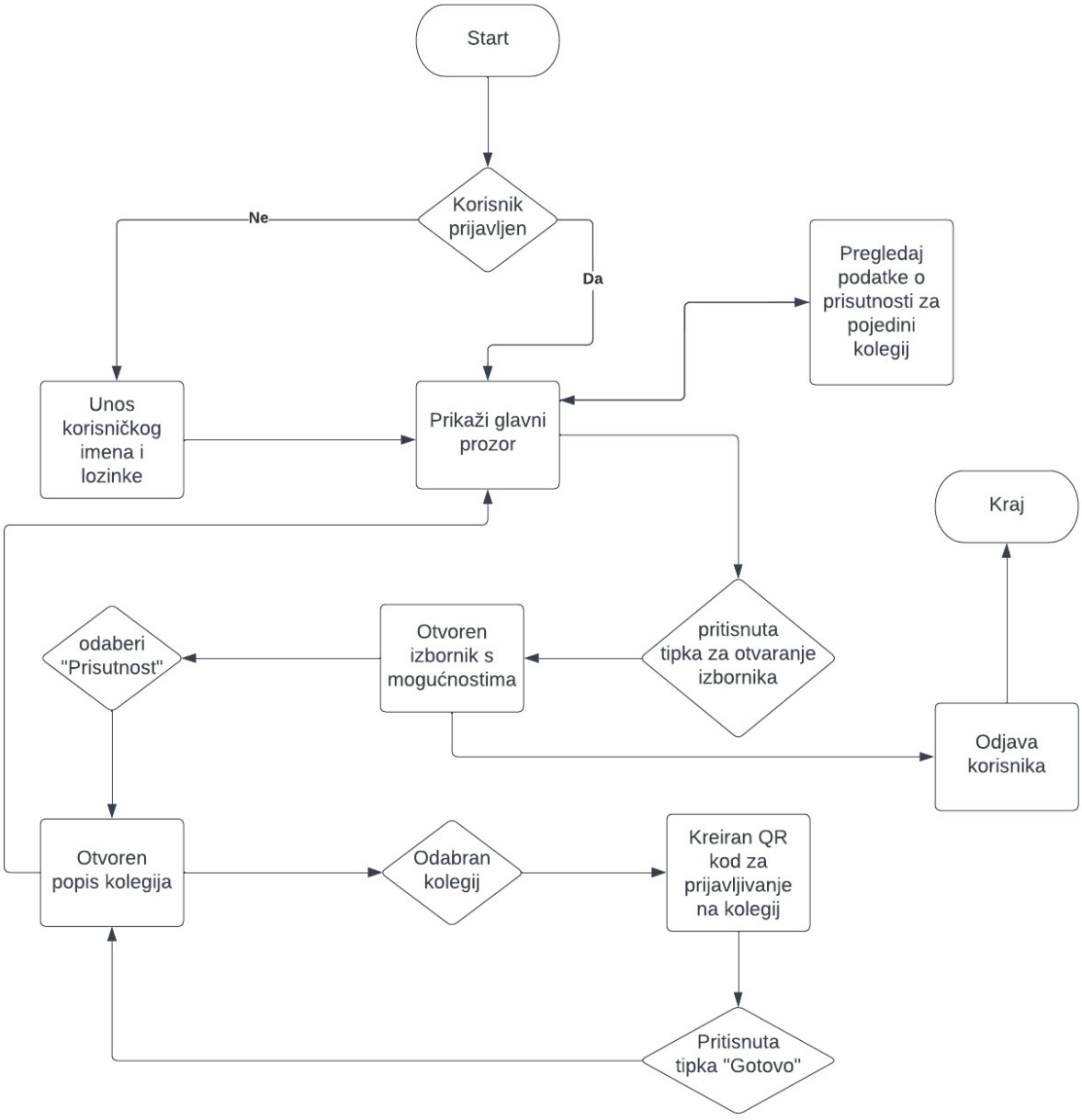
- Prijava u sustav – za prijavu u sustav potrebno je unijeti ispravan ID korisnika i odgovarajuću zaporku
- Promjena zaporke – za slučaj da je korisnik zaboravio zaporku ili želi kreirati svoju zaporku
- Pregled podataka o prisutnosti za pojedini kolegij
- Skeniranje QR koda za prijavu na nastavu
- Prikaz korisničkog profila s podacima o korisniku

Aplikacija za nastavnike ima sljedeće zahtjeve:

- Prijava u sustav – za prijavu u sustav potrebno je unijeti ispravan ID korisnika i odgovarajuću zaporku
- Promjena zaporke – za slučaj da je korisnik zaboravio zaporku ili da želi kreirati svoju zaporku
- Kreiranje QR koda – za evidentiranje prisutnosti studenata na pojedini kolegij
- Spremanje popisa u bazu podataka nakon evidentiranja studenata
- Pregled podataka o prisutnosti pojedinih studenata
- Kreiranje tablice s popisom svih studenata i slanje na e-mail
- Prikaz korisničkog profila s podacima o korisniku

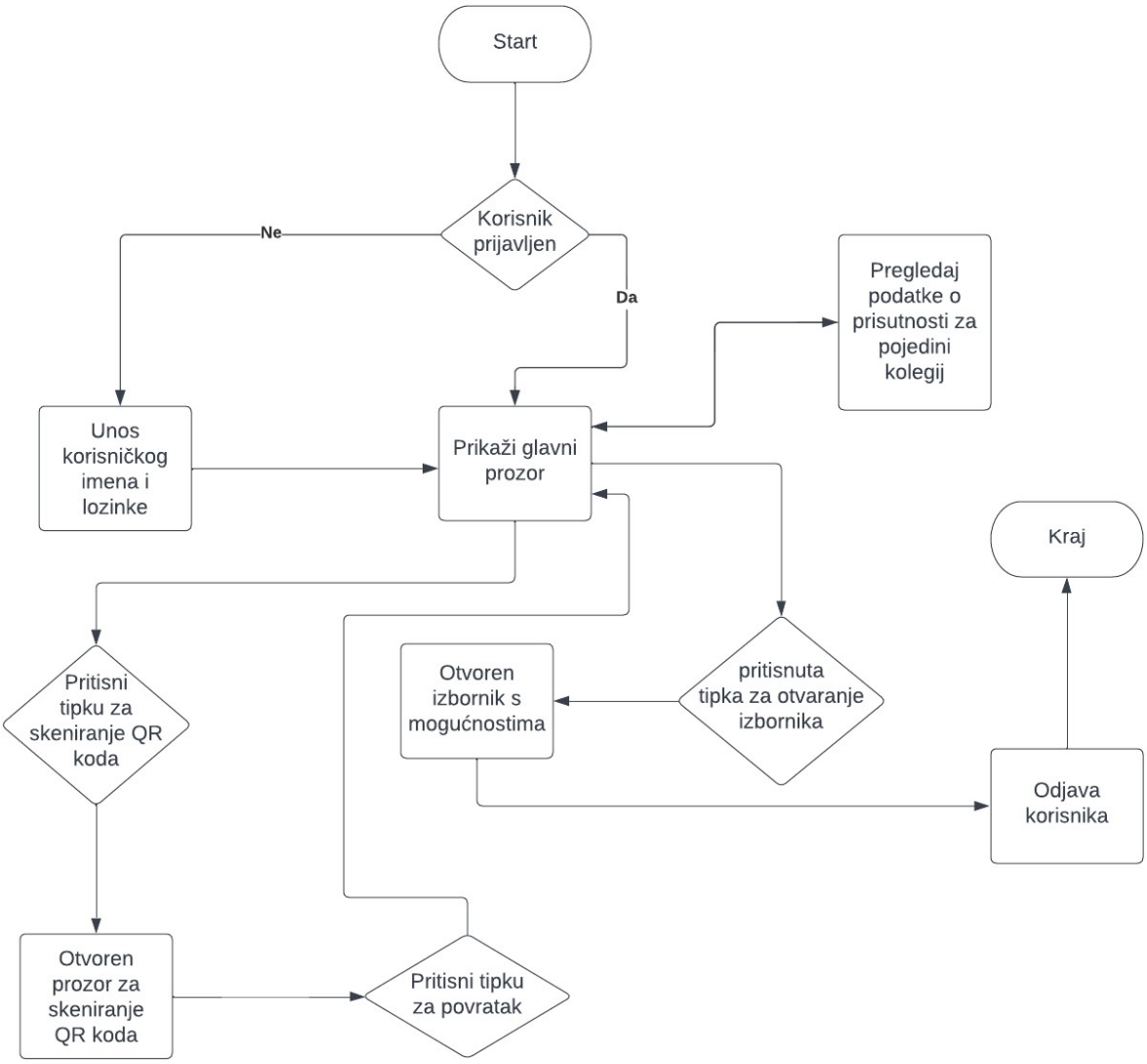
4.2. Dijagram toka

Kako bi bio jasan tijek aplikacije izrađen je dijagram toka. Ovaj dijagram prikazuje sve promjene od pokretanja aplikacije preko korištenja do odjave korisnika. Na slici 4.1 prikazan je dijagram toka aplikacije za nastavnike.



Sl. 4.1. Dijagram toka za aplikaciju za nastavnike

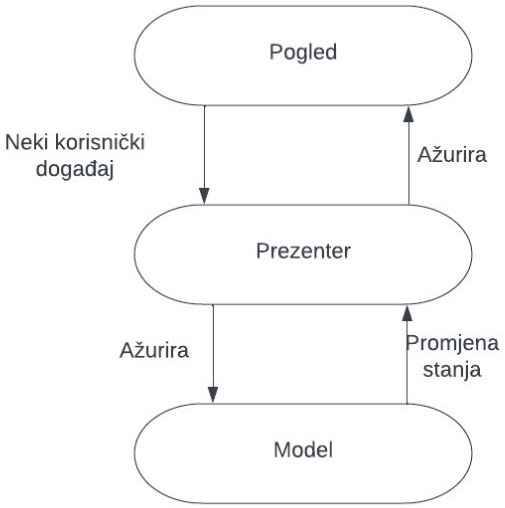
Dijagram toka za aplikaciju za studente prikazan je na slici 4.2.



Sl. 4.2. Dijagram toka za aplikaciju za studente

4.3. Arhitektura sustava

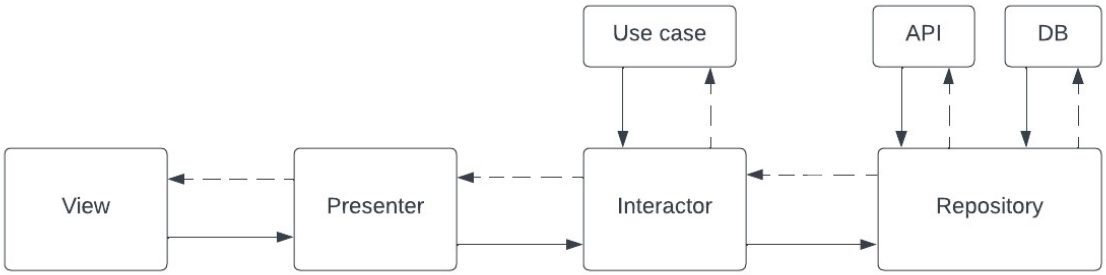
Za izradu aplikacija odabrana je MVP (*Model-View-Presenter*) arhitektura koja predstavlja varijantu *Model-View-Controler* arhitekture i upotrebljava se za izradu korisničkih sučelja [6]. MVP arhitektura se sastoji od tri dijela koji su međusobno povezani. *Model* sadrži sve modele koje aplikacija koristi, kao što su podaci o korisnicima i očitavanje podataka sa senzora za lokaciju. *View* dio arhitekture se brine za sve ono što je „vidljivo korisniku“, kao što je prikaz korisničkog sučelja i teksta. *Presenter* povezuje *Model* i *View,* koristi podatke iz modela kako bi ih prikazao na zaslonu korisnika. Dijagram za *Model-View-Presenter* je prikazan na slici 4.3.



Sl. 4.3. *Model-View-Presenter* dijagram

Blok shema funkcioniranja MVP-a prikazana je na slici 4.4., sastoji se od četiri skupine koje surađuju i nadopunjuju se kako bi pregled koda bio što lakši. Te skupine su:

- *View* – brine se a prikaz podataka na zaslonu uređaja
- *Presenter* – dohvaća podatke i prosljeđuje ih *View* dijelu
- *Interactor – presenter* koristi funkcije napisane u kodu pomoću interaktora
- *Repository* – koristi se za komunikaciju s bazom podataka i vanjskim servisima



Sl. 4.4. Blok shema MVP-a

5. IMPLEMENTACIJA PROGRAMSKOG RJEŠENJA

Prije početka razvoja aplikacije, potrebno je kreirati novi projekt prema standardima korištenja Android Studia.

5.1. Dodavanje ovisnosti

Kod izrade aplikacije, prvi korak je dodavanje ovisnosti (engl. *dependencies*) koje će biti korištene. Ovisnosti se dodaju u *Build.gradle* datoteku i nužne su za uspješno *build*-anje aplikacije. Također, ovisnosti omogućuju korištenje raznih biblioteka koje nije potrebno spremati na pametni uređaj. Slika 4.1.1. prikazuje popis korištenih ovisnosti za izradu aplikacija.

```
implementation fileTree(dir: 'libs', include: ['*.jar'])
implementation 'androidx.appcompat:appcompat:1.4.1'
implementation 'androidx.legacy:legacy-support-v4:1.0.0'
implementation 'com.google.android.material:material:1.5.0'
implementation 'androidx.constraintlayout:constraintlayout:2.1.3'
implementation 'androidx.preference:preference-ktx:1.2.0'
implementation 'com.google.firebase:firebase-storage:20.0.1'
implementation 'com.google.firebase:firebase-auth:21.0.3'
implementation 'com.google.firebase:firebase-database-ktx:20.0.5'
testImplementation 'junit:junit:4.13.2'
androidTestImplementation 'androidx.test:runner:1.4.0'
androidTestImplementation 'androidx.test.espresso:espresso-core:3.4.0'
implementation 'de.hdodenhof:circleimageview:3.0.0'
implementation 'com.github.GoodieBag:Pinview:v1.4'
```

Sl. 5.1. Korištene ovisnosti

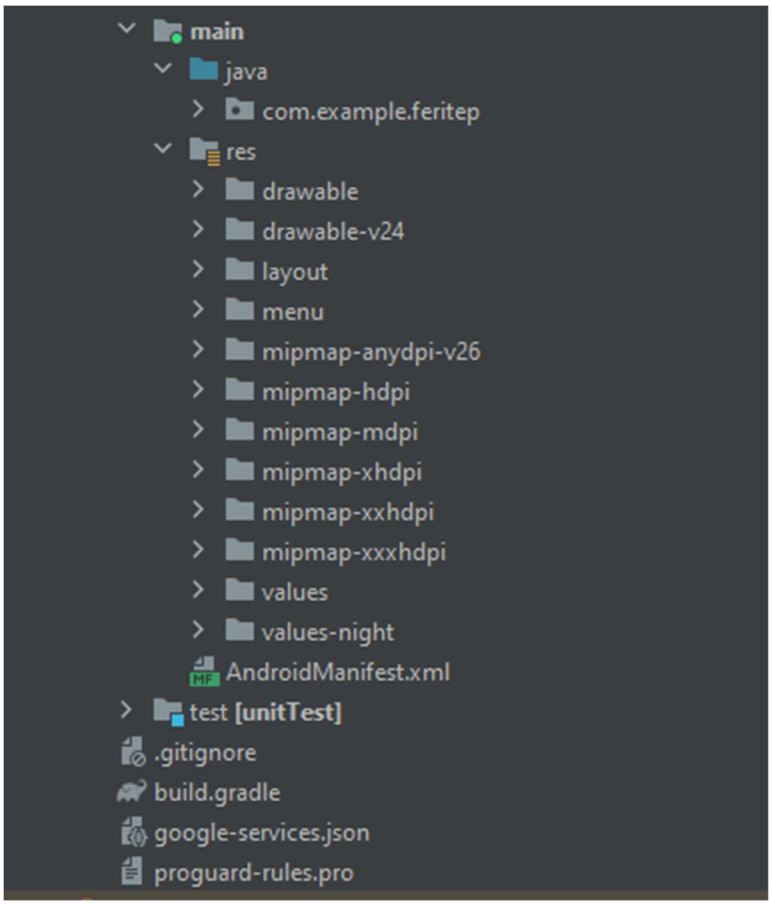
5.2. Struktura projekta

Problem ovog diplomskog rada ostvaren je objektno orijentiranim jezikom Kotlin. Uz pomoć Android Studia i njegovih alata koji su potrebni za uspješno programiranje aplikacija ostvaren je ovaj projekt. Projekt se sastoji od:

- java/ : sadrži izvorne kodove u Kotlinu

- res/ : sadrži ikone, datoteke XML, definicije grafičkih korisničkih sučelja
- test/ : sadrži datoteke za testiranje aplikacije

Detaljniji sadržaj prikazan je na slici 5.2.



Sl. 5.2. Sadržaj projekta

AndroidManifest.xml datoteka sadrži nekoliko bitnih stvari o svojstvima aplikacije koje određuju samu funkcionalnost aplikacije. Neka od tih bitnih stvari je informacija koje dozvole aplikacija koristi. Ako ove dozvole ne bi bile navedene, rješenje aplikacije se ne bi moglo uspješno implementirati. Također, u *AndroidManifest.xml* datoteci su navedene i neke osnovne informacije i elementi. Primjeri dozvola su prikazani na slici 5.3.

```
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
<uses-permission android:name="android.permission.CAMERA"/>
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="com.google.android.providers.gsf.permission.READ_GSERVICES"/>
```

Sl. 5.3. Dozvole aplikacije

Prikaz sučelja ili *layout*-a sastoji se od nekoliko osnovnih elemenata, kao što su:

- *EditText* – okvir za prikaz i uređivanje teksta
- *ImageView* – okvir za prikazivanje slika
- *Button* – tipka koja izvršava radnju deklariranu u kodu

Primjer za prikaz slike je prikazan na slici ispod:

```
<ImageView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:src="@drawable/side_nav_bar" />
```

Sl. 5.4. XML kod za kreiranje slike

Primjer za *Button* je prikazan na slici ispod:

```
<Button android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/button_profilna_slika"
        android:id="@+id/bt_upload_profile_pic_update"
        android:padding="5dp"
/>
```

Sl. 5.5. XML kod za *Button*

Primjer za tekst je prikazan je na slici ispod:

```
<TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Adresa"
        android:textSize="20sp"
        android:textStyle="bold"
        android:paddingStart="16dp"
        android:paddingEnd="16dp"
        android:paddingBottom="5dp"
/>
```

Sl. 5.6. XML kod za tekst

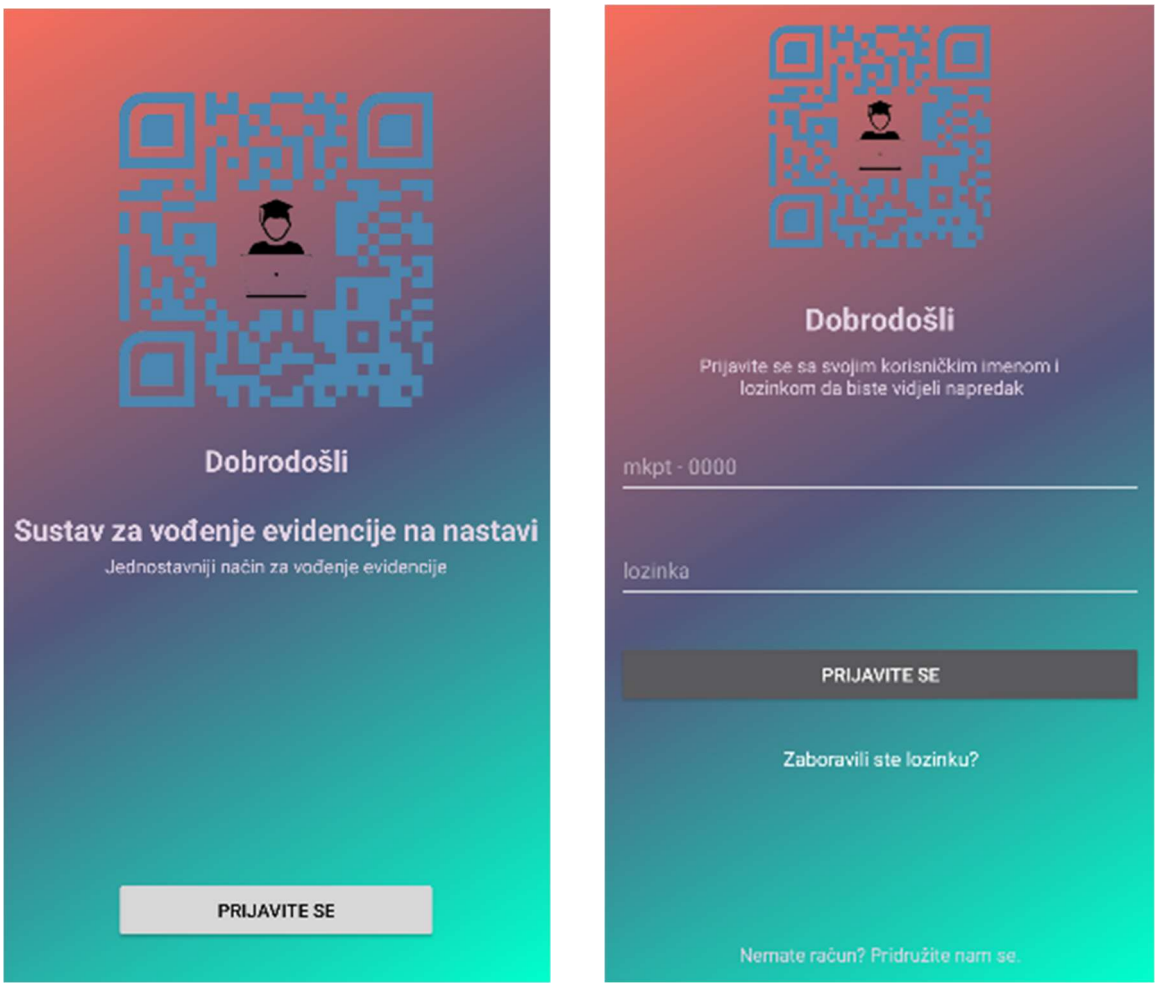
Google-services.json datoteka sadrži podatke koji su potrebni kako bi se aplikacija povezala na Firebase, kao što su API ključ, naziv projekta, osnovne informacije o aplikaciji. Primjer je prikazan na slici ispod:

```
"client_info": {
  "mobilesdk_app_id": "1:168034298035:android:faccd8a0aab3e0be21a8c6",
  "android_client_info": {
    "package_name": "com.example.feritep"
  }
```

Sl. 5.7. Sadržaj *Google-services.json* datoteke

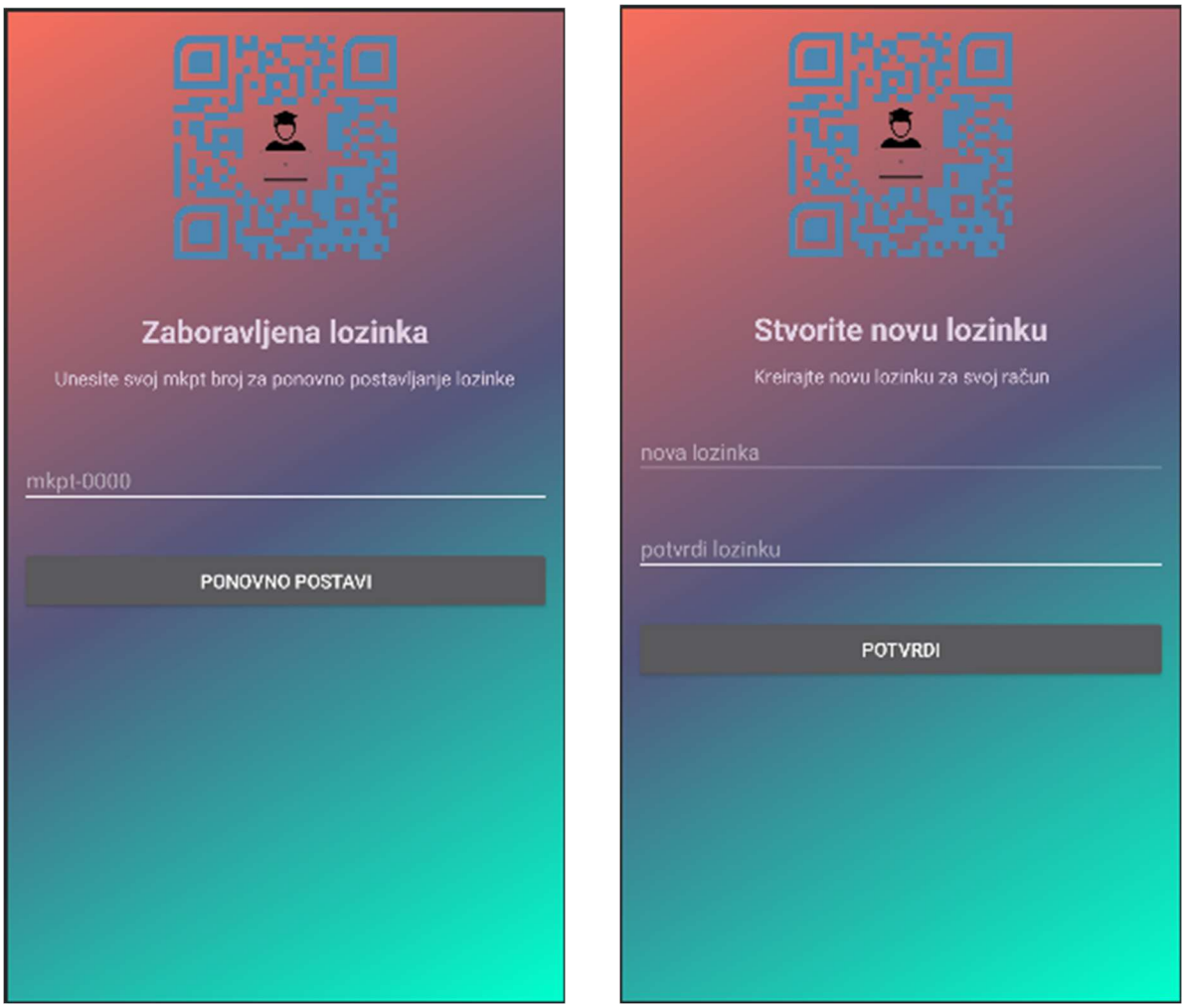
5.3. Aplikacija za studente

Početni zaslon aplikacije za studente se sastoji od poruke dobrodošlice i kontrola za prijavu korisnika. Klikom na kontrolu „PRIJAVITE SE“, korisnik se usmjerava na stranicu za prijavu u sustav. Dodani su zaštitni logo aplikacije i dva polja za unos korisničke oznake i zaporke, koji su definirani pri registraciji, kao što je prikazano na slici 5.8.



Sl. 5.8. Početni zaslon aplikacije (lijevo) i zaslon za prijavu u sustav (desno)

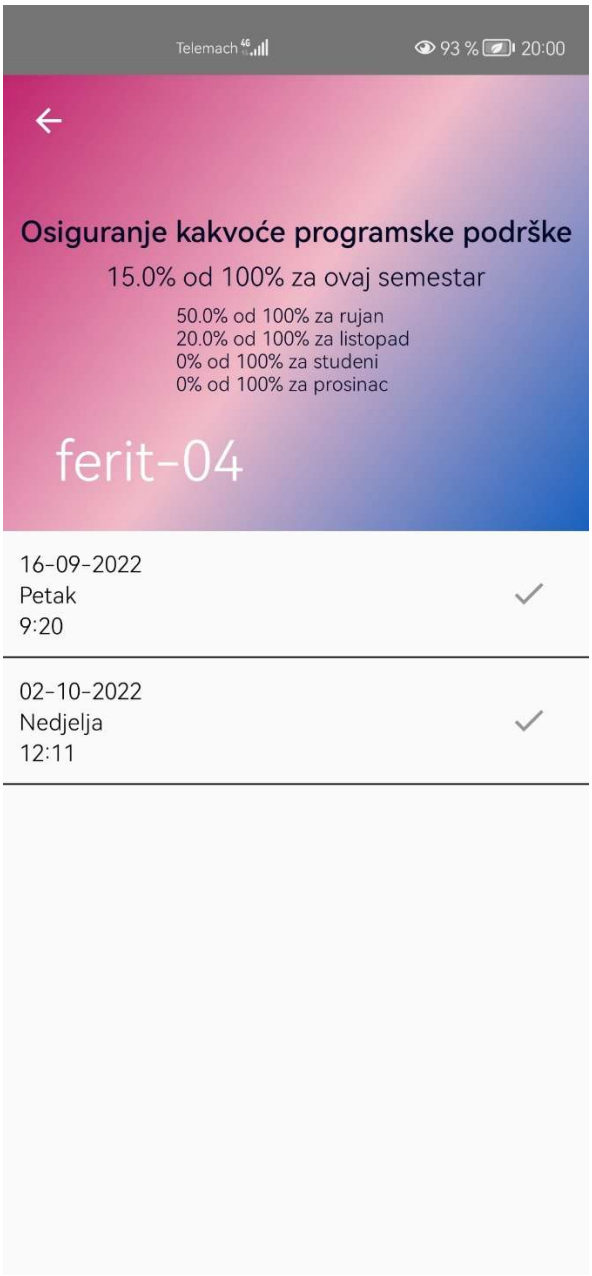
U slučaju da je korisnik zaboravio svoju zaporku, ispod kontrole „PRIJAVITE SE“ nalazi se link za oporavak zaporke pod nazivom „Zaboravili ste lozinku?“. Pritiskom na taj link, korisnik se preusmjerava na stranicu za oporavak zaporke. Slika 5.9. prikazuje zaslone za oporavak zaporke.



Sl. 5.9. Zasloni za ponovno postavljanje zaporke

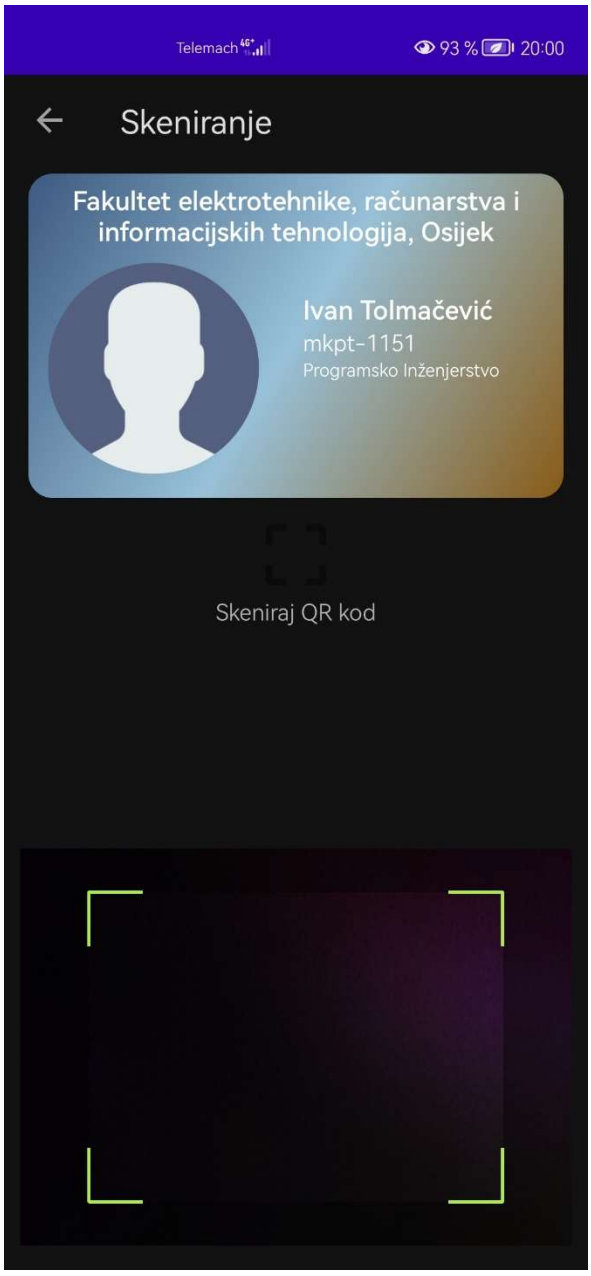
Nakon uspješne prijave u sustav, otvara se zaslon s popisom kolegija koje student pohađa u tekućem semestru, prikazano na slici 5.10. Student odabire kolegij nakon čega se otvara lista koja sadrži detalje o prisutnosti, kao što su datum i vrijeme kada je student skenirao QR kod s nastavnikova uređaja, kao što je prikazano na slici 5.11.

Sl. 5.10. Prikaz kolegija koje student pohađa

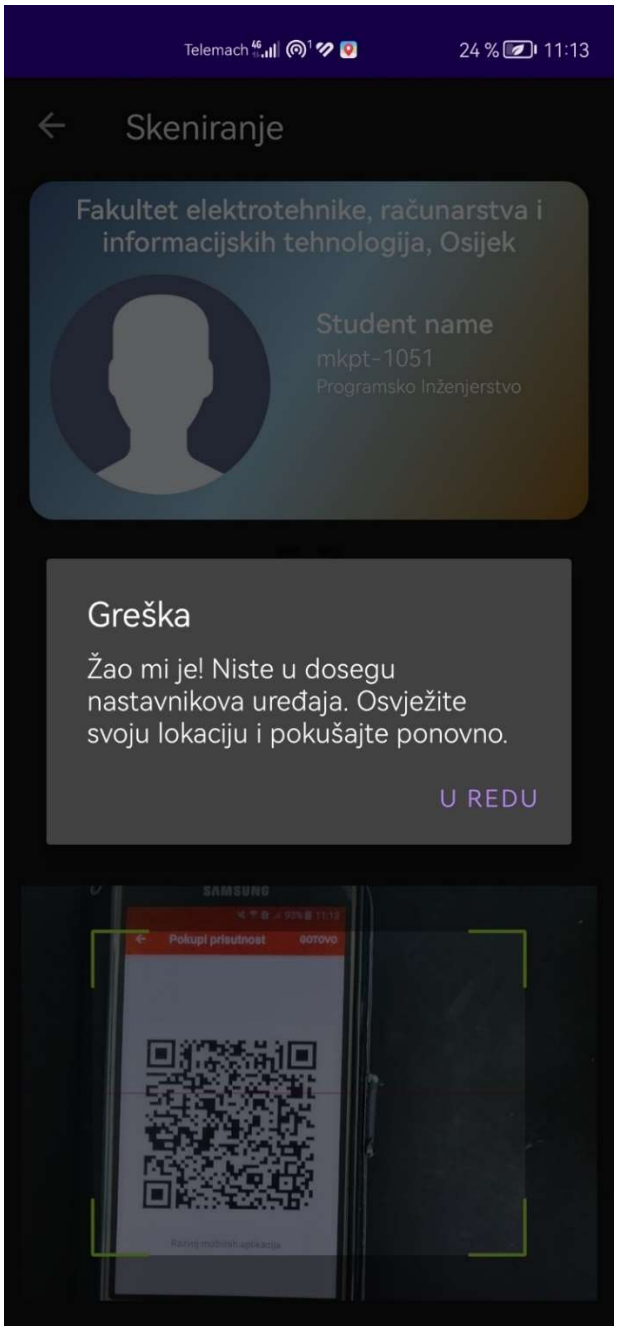


Sl. 5.11. Prikaz detalja o kolegiju

Na početnom zaslonu se nalazi i kontrola koja vodi na zaslon za skeniranje QR koda, kao što je prikazano na slici 5.12. Aplikacija koristi kameru uređaja kako bi skenirala QR kod koji je automatski generiran putem aplikacije za nastavnike. Aplikacija koristi lokacijske usluge kako bi se osiguralo da studenti ne mogu skenirati QR kod ako su udaljeni više od 10 metara od nastavnikova uređaja. Slika 5.13. prikazuje skočni prozor koji obavještava studenta da je previše udaljen od telefona nastavnika.

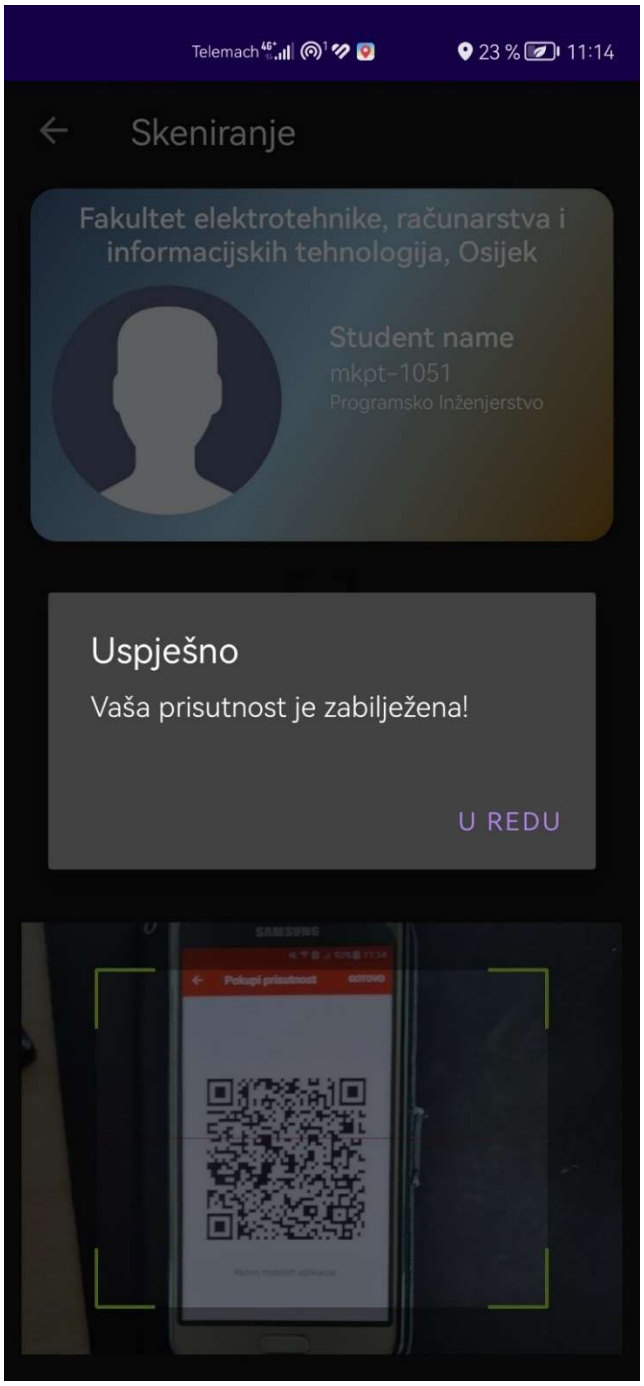


Sl. 5.12. Zaslon za skeniranje QR koda



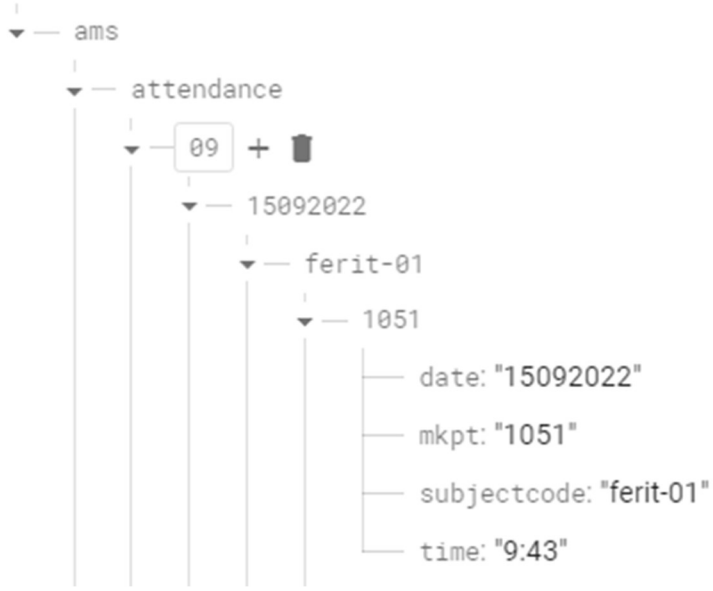
Sl. 5.13. Obavijest za udaljenost

Ako je student uspješno skenirao QR kod za prijavu na nastavu na zaslonu uređaja se pojavljuje obavijest, kao što je prikazano na slici 5.14.



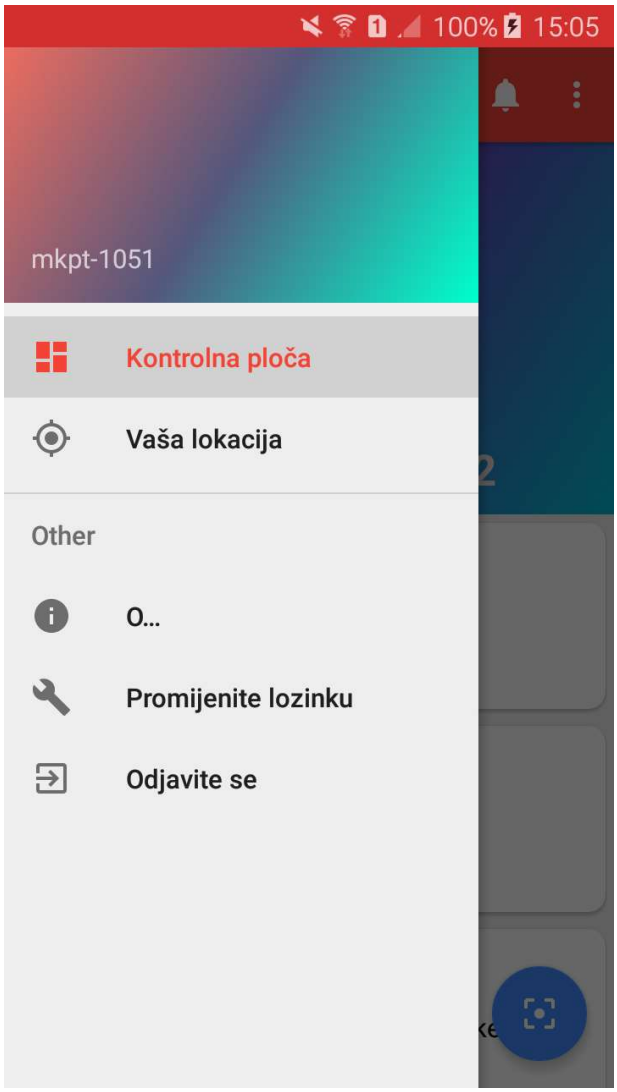
Sl. 5.14. Obavijest o uspješnom prijavljivanju na kolegij

Nakon što student skenira QR kod s nastavnikovog pametnog uređaja, u Firebase se upisuju podaci o prijavi. Ti podaci su kasnije vidljivi i studentu i nastavniku. Slika 5.15. prikazuje podatke upisane u Firebase.



Sl. 5.15. Izgled tablice u Firebase-u nakon prijave studenta na predavanje

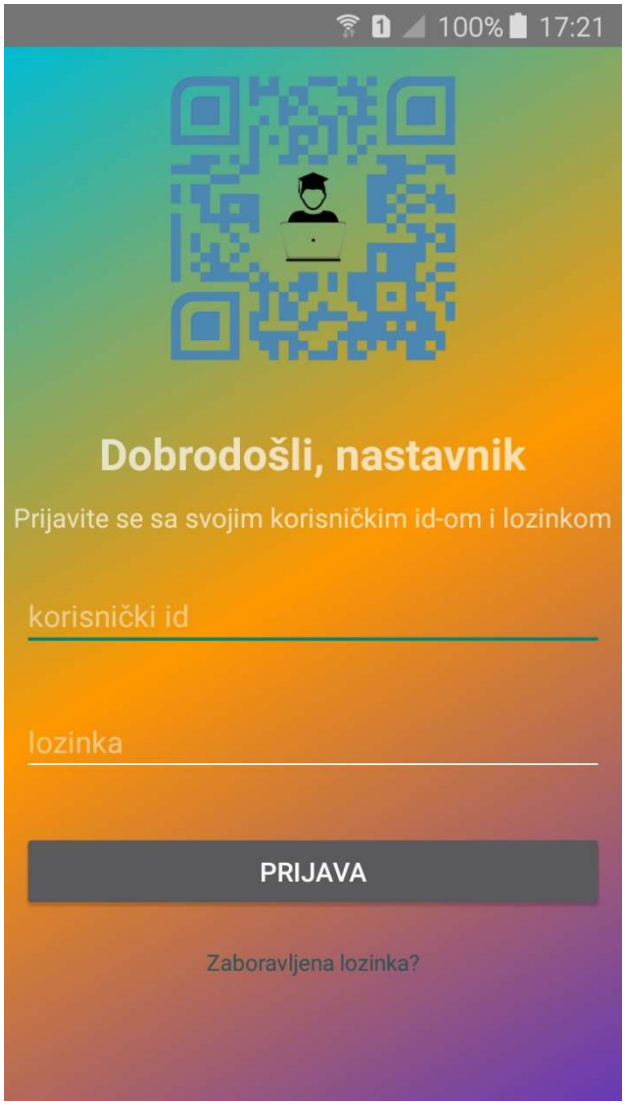
Na početnoj stranici, u gornjem lijevom kutu, nalazi se ikona kojom se otvara izbornik s opcijom za pregled kolegija koje student pohađa, promjenu zaporke i odjavu iz aplikacije, što se vidi na slici 5.16.



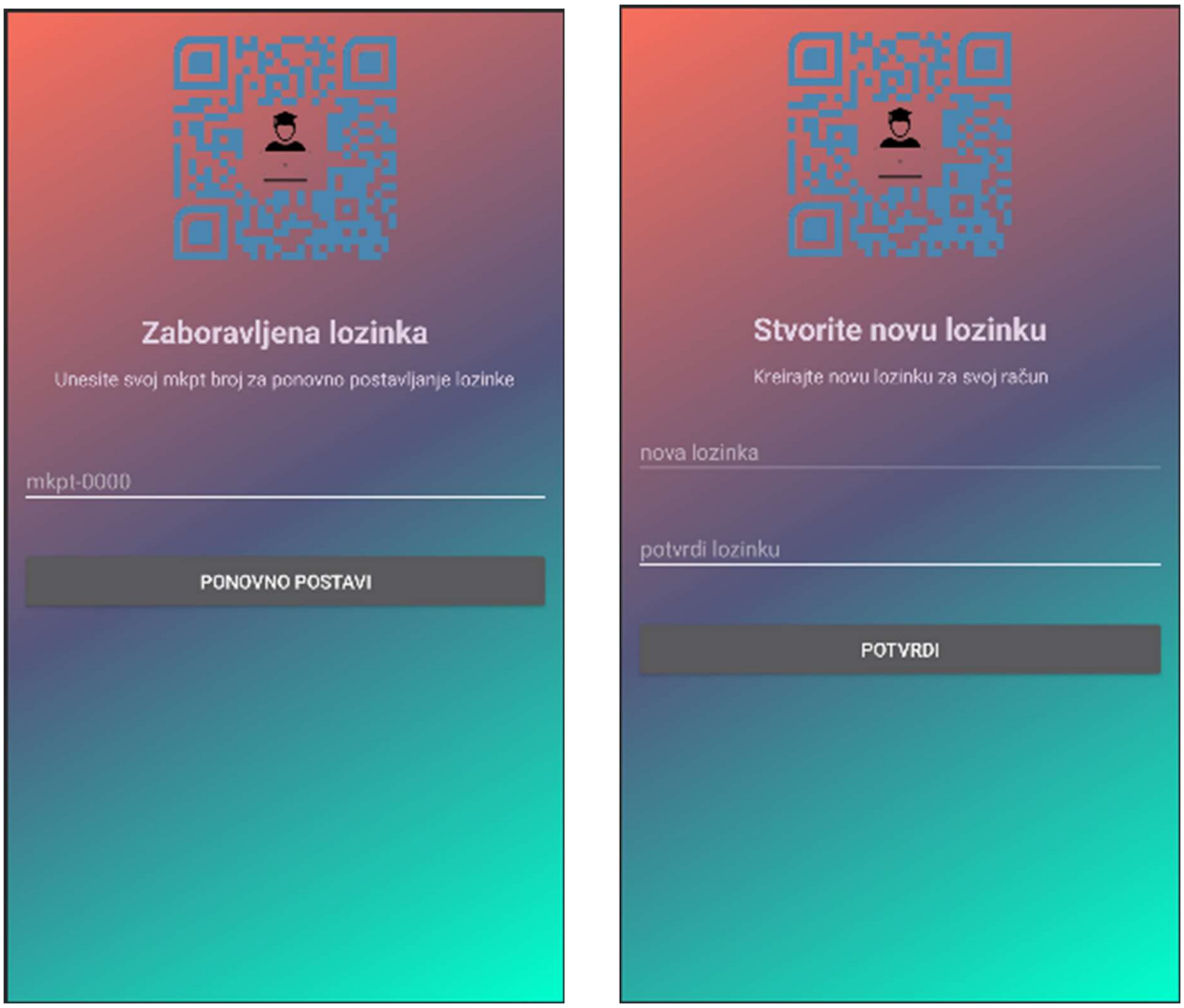
Sl. 5.16. Izbornik s opcijama

5.4. Aplikacija za nastavnike

Početni zaslon aplikacije za nastavnike, prikazan na slici 5.17., sastoji se od poruke dobrodošlice, polja za unos korisničkog ID-a i zaporke te kontrole za prijavu korisnika. Ispod kontrole „Prijava“ nalazi se link pod nazivom „Zaboravljena lozinka?“ koji preusmjerava korisnika na stranicu za ponovno postavljanje zaporke, što je prikazano na slici 5.18.



Sl. 5.17. Početni zaslon aplikacije za nastavnike



Sl. 5.18. Zasloni za ponovno postavljanje zaporke

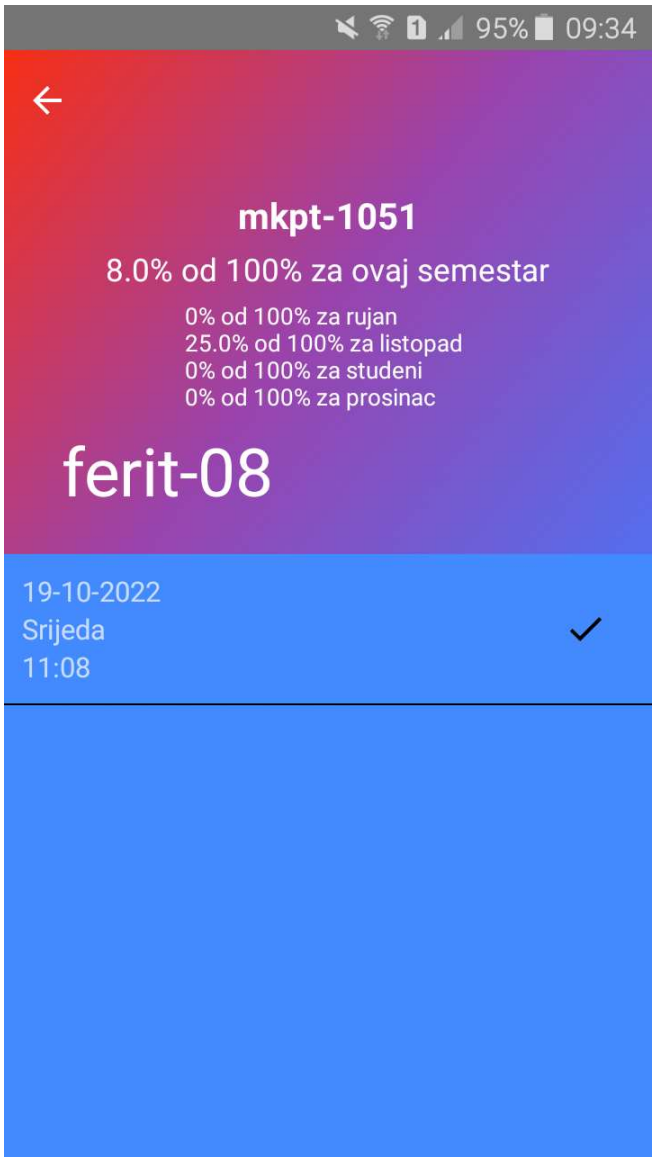
Nakon što se nastavnik uspješno prijavi, preusmjerava se na glavni zaslon s popisom studenata koji pohađaju njegove kolegije, što je prikazano na slici 5.19.

Sl. 5.19. Popis studenata

Klikom na pojedinog studenta, otvara se zaslon s popisom kolegija koje student sluša kod nastavnika koji je trenutno prijavljen u aplikaciju. Slika 5.20. prikazuje popis kolegija.

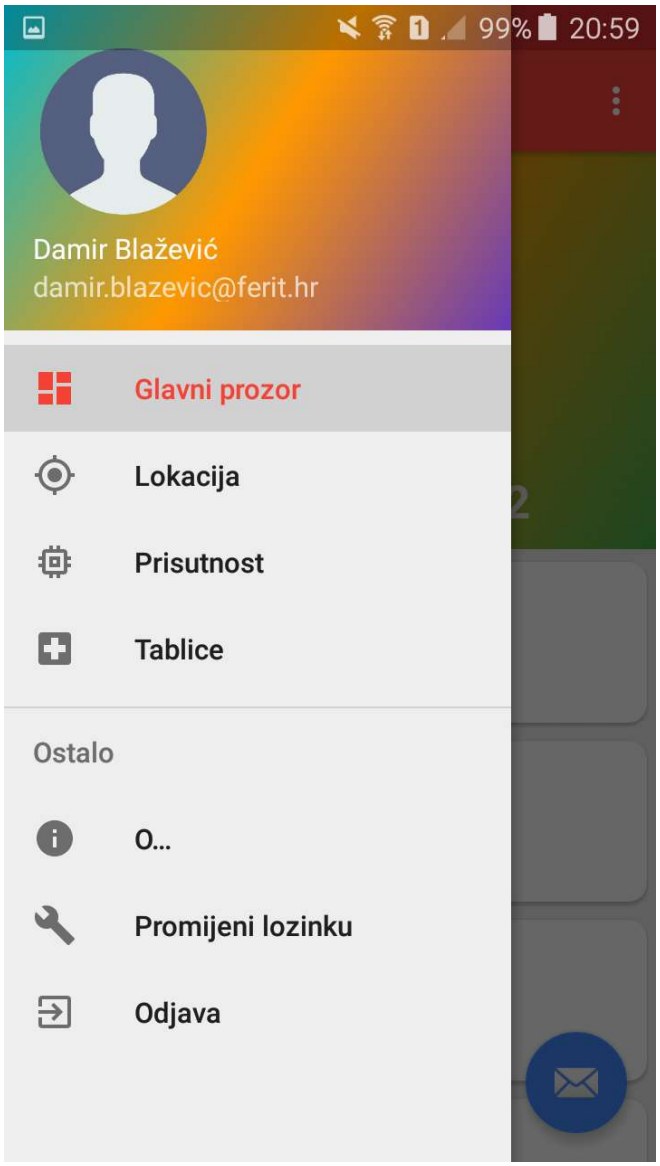
Sl. 5.20. Popis kolegija koje student sluša

Odabirom kolegija otvara se prozor s detaljima o kolegiju, kao što su: ID studenta, kod kolegija, ukupni postotak prisutnosti na kolegiju, postotak prisutnosti za svaki mjesec u kojem se kolegij izvodi te datum i vrijeme za svaku prijavu studenta, kao što je prikazano na slici 5.21.



Sl. 5.21. Prozor s detaljima o kolegiju

Na glavnom zaslonu aplikacije, u gornjem lijevom kutu, nalazi se kontrola za otvaranje izbornika s opcijama za povrat na glavni zaslon, opcija za prikupljanje prisutnosti te opcija za kreiranje tablica s detaljima o prisutnosti studenata. Slika 5.22. prikazuje izbornik s opcijama.



Sl. 5.22. Izbornik s opcijama

Pritiskom na kontrolu „Prisutnost“ otvara se prozor s popisom kolegija koje predaje nastavnik koji je trenutno prijavljen u aplikaciju, kao što je prikazano na slici 5.23.

Sl. 5.23. Popis kolegija koje nastavnik predaje

Odabirom pojedinog kolegija, pojavljuje se obavijest koja traži potvrdu za prikupljanje prisutnosti za odabrani kolegij. Nakon što nastavnik pritisne kontrolu „Pokupite“, prikazano na slici 5.24., aplikacija provjerava vrijeme i datum na pametnom uređaju i uspoređuje ih s vremenom izvođenja kolegija koji su zapisani u Firebase-u. Ako se datum i vrijeme na Firebase-u i pametnom uređaju podudaraju generira se QR kod koji služi za prijavljivanje studenata na nastavu. Slika 5.25. prikazuje zaslon s QR kodom.

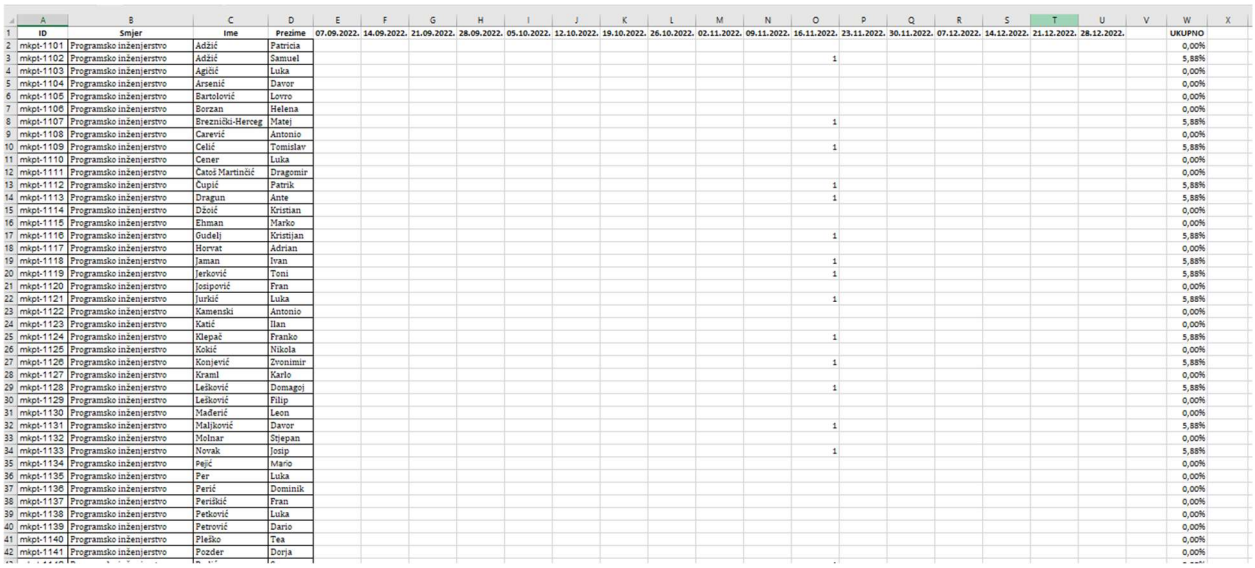
Sl. 5.24. Zaslon s obavijesti

Sl. 5.25. QR kod za prijavljivanje prisutnosti

Kada nastavnik, u izborniku s opcijama odabere „Tablice“, otvara se prozor s popisom kolegija, prikazan na slici 5.26., te se klikom na pojedini kolegij generira tablica s popisom svih studenta koji slušaju odabrani kolegij. U toj tablici se nalaze detalji o prisutnosti za pojedinog studenta, kao što su datum kada je student skenirao QR kod kako bi potvrdio svoju prisutnost na kolegiju te ukupna prisutnost. Prisutnost je prikazana na tablici 5.1.

Sl. 5.26. Popis kolegija

Tablica 5.1. Sveukupna prisutnost

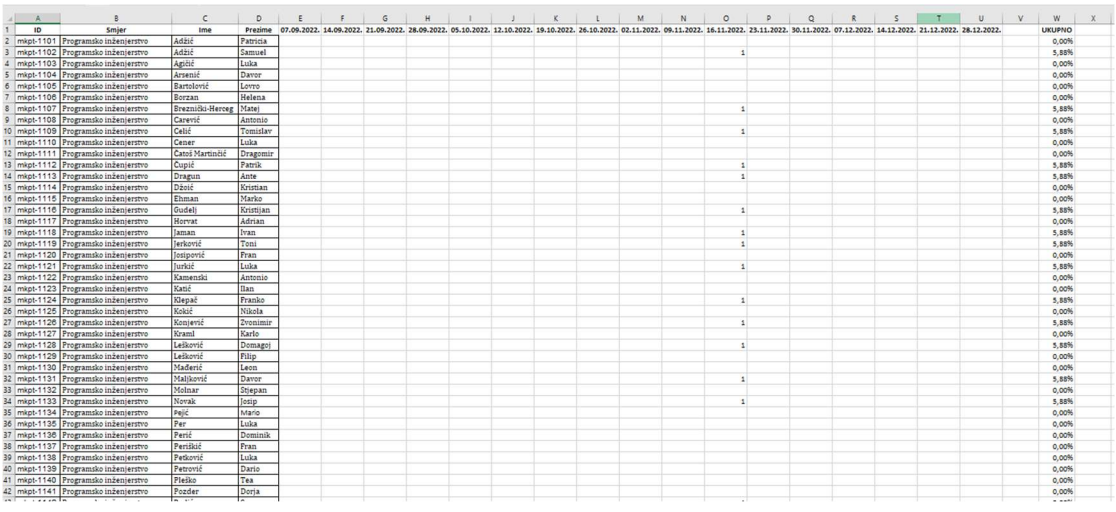


ID	Smjer	Ime	Prezime	07.09.2022.	14.09.2022.	21.09.2022.	28.09.2022.	05.10.2022.	12.10.2022.	19.10.2022.	26.10.2022.	02.11.2022.	09.11.2022.	16.11.2022.	23.11.2022.	30.11.2022.	07.12.2022.	14.12.2022.	21.12.2022.	28.12.2022.		UKUPNO
mkpt-1101	Programsko inženjerstvo	Adžić	Patricia																			0,00%
mkpt-1102	Programsko inženjerstvo	Adžić	Samuel											1								5,88%
mkpt-1103	Programsko inženjerstvo	Agičić	Luka																			0,00%
mkpt-1104	Programsko inženjerstvo	Arsenić	Davor																			0,00%
mkpt-1105	Programsko inženjerstvo	Bartolović	Lovro																			0,00%
mkpt-1106	Programsko inženjerstvo	Borzan	Helena																			0,00%
mkpt-1107	Programsko inženjerstvo	Breznički-Hercog	Matej											1								5,88%
mkpt-1108	Programsko inženjerstvo	Carević	Antonio																			0,00%
mkpt-1109	Programsko inženjerstvo	Celić	Tomislav											1								5,88%
mkpt-1110	Programsko inženjerstvo	Cener	Luka																			0,00%
mkpt-1111	Programsko inženjerstvo	Čatoš Martinčić	Dragomir																			0,00%
mkpt-1112	Programsko inženjerstvo	Čupić	Patrik											1								5,88%
mkpt-1113	Programsko inženjerstvo	Dragun	Ante											1								5,88%
mkpt-1114	Programsko inženjerstvo	Džoić	Kristian																			0,00%
mkpt-1115	Programsko inženjerstvo	Ehman	Marko																			0,00%
mkpt-1116	Programsko inženjerstvo	Gudelj	Kristijan											1								5,88%
mkpt-1117	Programsko inženjerstvo	Horvat	Adrian																			0,00%
mkpt-1118	Programsko inženjerstvo	Jaman	Ivan											1								5,88%
mkpt-1119	Programsko inženjerstvo	Jerković	Toni											1								5,88%
mkpt-1120	Programsko inženjerstvo	Josipović	Fran																			0,00%
mkpt-1121	Programsko inženjerstvo	Jurkić	Luka											1								5,88%
mkpt-1122	Programsko inženjerstvo	Kamenski	Antonio																			0,00%
mkpt-1123	Programsko inženjerstvo	Katić	Ilan																			0,00%
mkpt-1124	Programsko inženjerstvo	Klepač	Franko											1								5,88%
mkpt-1125	Programsko inženjerstvo	Kokić	Nikola																			0,00%
mkpt-1126	Programsko inženjerstvo	Konjević	Zvonimir											1								5,88%
mkpt-1127	Programsko inženjerstvo	Kraml	Karlo																			0,00%
mkpt-1128	Programsko inženjerstvo	Leškovič	Domagoj											1								5,88%
mkpt-1129	Programsko inženjerstvo	Leškovič	Filip																			0,00%
mkpt-1130	Programsko inženjerstvo	Mađerić	Leon																			0,00%
mkpt-1131	Programsko inženjerstvo	Maljković	Davor											1								5,88%
mkpt-1132	Programsko inženjerstvo	Molnar	Stjepan																			0,00%
mkpt-1133	Programsko inženjerstvo	Novak	Josip											1								5,88%
mkpt-1134	Programsko inženjerstvo	Pejić	Mario																			0,00%
mkpt-1135	Programsko inženjerstvo	Per	Luka																			0,00%
mkpt-1136	Programsko inženjerstvo	Perić	Dominik																			0,00%
mkpt-1137	Programsko inženjerstvo	Periškić	Fran																			0,00%
mkpt-1138	Programsko inženjerstvo	Pešković	Luka																			0,00%
mkpt-1139	Programsko inženjerstvo	Petrović	Dario																			0,00%
mkpt-1140	Programsko inženjerstvo	Pleško	Tea																			0,00%
mkpt-1141	Programsko inženjerstvo	Pozder	Dorja																			0,00%

5.5. Testiranje aplikacija

Testiranje aplikacija provedeno je na studentima 2. godine preddiplomskog studija Elektrotehnike i informacijske tehnologije, smjer Komunikacije i informatika, te na 2. godini preddiplomskog studija Računarstvo, smjerovi Programsko inženjerstvo i Računalno inženjerstvo. U testiranju je sudjelovalo 97 studenata. Tablica 5.2. prikazuje popis studenata nakon evidentiranja prisutnosti putem aplikacije. Nažalost, neki studenti nisu mogli biti prijavljeni zbog toga što su koristili pametne uređaje s iOS operacijskim sustavom, pa je njihova prisutnost morala biti evidentirana potpisima na papiru.

Tablica 5.2. Popis studenata



ID	Smjer	Ime	Prezime	07.09.2022.	14.09.2022.	21.09.2022.	28.09.2022.	05.10.2022.	12.10.2022.	19.10.2022.	26.10.2022.	02.11.2022.	09.11.2022.	16.11.2022.	23.11.2022.	30.11.2022.	07.12.2022.	14.12.2022.	21.12.2022.	28.12.2022.		UKUPNO
mkpt-1101	Programsko inženjerstvo	Adžić	Patricia																			0,00%
mkpt-1102	Programsko inženjerstvo	Adžić	Samuel											1								5,88%
mkpt-1103	Programsko inženjerstvo	Agičić	Luka																			0,00%
mkpt-1104	Programsko inženjerstvo	Arsenić	Davor																			0,00%
mkpt-1105	Programsko inženjerstvo	Bartolović	Lovro																			0,00%
mkpt-1106	Programsko inženjerstvo	Borzan	Helena																			0,00%
mkpt-1107	Programsko inženjerstvo	Breznički-Hercog	Matej											1								5,88%
mkpt-1108	Programsko inženjerstvo	Carević	Antonio																			0,00%
mkpt-1109	Programsko inženjerstvo	Celić	Tomislav											1								5,88%
mkpt-1110	Programsko inženjerstvo	Cener	Luka																			0,00%
mkpt-1111	Programsko inženjerstvo	Čatoš Martinčić	Dragomir																			0,00%
mkpt-1112	Programsko inženjerstvo	Čupić	Patrik											1								5,88%
mkpt-1113	Programsko inženjerstvo	Dragun	Ante											1								5,88%
mkpt-1114	Programsko inženjerstvo	Džoić	Kristian																			0,00%
mkpt-1115	Programsko inženjerstvo	Ehman	Marko																			0,00%
mkpt-1116	Programsko inženjerstvo	Gudelj	Kristijan											1								5,88%
mkpt-1117	Programsko inženjerstvo	Horvat	Adrian																			0,00%
mkpt-1118	Programsko inženjerstvo	Jaman	Ivan											1								5,88%
mkpt-1119	Programsko inženjerstvo	Jerković	Toni											1								5,88%
mkpt-1120	Programsko inženjerstvo	Josipović	Fran																			0,00%
mkpt-1121	Programsko inženjerstvo	Jurkić	Luka											1								5,88%
mkpt-1122	Programsko inženjerstvo	Kamenski	Antonio																			0,00%
mkpt-1123	Programsko inženjerstvo	Katić	Ilan																			0,00%
mkpt-1124	Programsko inženjerstvo	Klepač	Franko											1								5,88%
mkpt-1125	Programsko inženjerstvo	Kokić	Nikola																			0,00%
mkpt-1126	Programsko inženjerstvo	Konjević	Zvonimir											1								5,88%
mkpt-1127	Programsko inženjerstvo	Kraml	Karlo																			0,00%
mkpt-1128	Programsko inženjerstvo	Leškovič	Domagoj											1								5,88%
mkpt-1129	Programsko inženjerstvo	Leškovič	Filip																			0,00%
mkpt-1130	Programsko inženjerstvo	Mađerić	Leon																			0,00%
mkpt-1131	Programsko inženjerstvo	Maljković	Davor											1								5,88%
mkpt-1132	Programsko inženjerstvo	Molnar	Stjepan																			0,00%
mkpt-1133	Programsko inženjerstvo	Novak	Josip											1								5,88%
mkpt-1134	Programsko inženjerstvo	Pejić	Mario																			0,00%
mkpt-1135	Programsko inženjerstvo	Per	Luka																			0,00%
mkpt-1136	Programsko inženjerstvo	Perić	Dominik																			0,00%
mkpt-1137	Programsko inženjerstvo	Periškić	Fran																			0,00%
mkpt-1138	Programsko inženjerstvo	Pešković	Luka																			0,00%
mkpt-1139	Programsko inženjerstvo	Petrović	Dario																			0,00%
mkpt-1140	Programsko inženjerstvo	Pleško	Tea																			0,00%
mkpt-1141	Programsko inženjerstvo	Pozder	Dorja																			0,00%

6. Zaključak

U ovom radu opisan je način dizajniranja i korištenje mobilne aplikacije za vođenje evidencije prisutnosti. Cilj je bio napraviti jednostavnu aplikaciju za vođenje evidencije prisutnosti i demonstrirati njeno korištenje. Aplikacija posjeduje intuitivno sučelje i omogućava relativno jednostavan način pojave i rad nastavniku i studentima. Aplikacija je rađena za pametne uređaje s operacijskim sustavom Android i nije ju moguće koristiti na uređajima koji rade na iOS operacijskom sustavu. Prilog daljnjeg rada na aplikaciji obuhvaćao bi izradu aplikacije za uređaje koji rade na iOS operacijskom sustavu, odnosno izradu multi-platformske aplikacije. Također, određena poboljšanja mogla bi se napraviti korekcijama postupka prijave u aplikaciju te nešto drugačijeg načina geo lociranja korisnika s obzirom na ograničenja GPS tehnologije u zatvorenim prostorima.

S obzirom na relativno široku mogućnost primjene, izrada komercijalne aplikacije bi trebala uključivati više stručnjaka različitih profila, od osobe koja će osmisliti cijeli koncept, preko dizajnera, programera za Android i iOS operacijske sustave, pa sve do marketinških stručnjaka koji će adekvatno predstaviti proizvod na tržištu.

Tijekom testiranja aplikacije došlo se do zaključka da bi prijava u aplikaciju bila brža i modernija ako bi se korisnici mogli prijaviti otiskom prsta ili skeniranjem lica. Također, kada bi aplikacija bila izrađena za pametne uređaje s iOS operacijskim sustavima, obuhvatila bi sve korisnike pa u budućnosti ne bi više bilo potrebe za papirom s popisom prisutnih na predavanjima. Umjesto lokacije uređaja putem GPS tehnologije mogla bi se koristiti neka druga tehnologija, na primjer, pametni uređaj nastavnika kao pristupna točka ili NFC (engl. Near Field Communication) tehnologija u slučaju da korisnik ne želi dati aplikaciji pristup svojoj lokaciji.

LITERATURA

[1] Presli – Attendance Manager:

https://play.google.com/store/apps/details?id=com.couchbits.presence

[2] Alora – Attendance Tracker App:

https://play.google.com/store/apps/details?id=com.smartlogicinc.attendancemanager

[3] Self Attendance:

https://play.google.com/store/apps/details?id=saurabhrao.selfattendance

[4] QR Attendance Control:
https://play.google.com/store/apps/details?id=com.qrattendancecontrol

[5] Advanced Encryption Standard, GeeksForGeeks:

https://www.geeksforgeeks.org/advanced-encryption-standard-aes/ [veljača, 2022.]

[6] Model-view-presenter, GeeksForGeeks:

https://www.geeksforgeeks.org/mvp-model-view-presenter-architecture-pattern-in-android-with-example/ [listopad, 2020.]

[7] B.Phillips, C. Stewart, K. Marsciano, Android Programming: The Big Nerd Ranch Guide (3rd Edition), USA, 2017.

[8] R. Meier, Professional Android 4 Application Development, John Wiley and Sons, Indianapolis, USA, 2012.

[9] M. Gargenta, Learning Android, O'Reilly Media, Sebastopol, 2011.

[10] Official Android Developer page: https://developer.android.com/

[11] Official Firebase page: https://firebase.google.com/

[12] Official Kotlin page: https://kotlinlang.org/

[13] FreeCodeCamp YouTube page: https://www.youtube.com/c/Freecodecamp/featured

[14] Philipp Lackner YouTube page: https://www.youtube.com/c/PhilippLackner

[15] Traversy Media YouTube page: https://www.youtube.com/c/TraversyMedia

[16] F. Babić, L. Kordić, N. Srivastava, Kotlin Coroutines by Tutorials, 3rd Edition, Ray Wenderlich, USA, 2022.

[17] Official StackOverflow page: https://stackoverflow.com/

[18] Razvoj mobilnih aplikacija, priručnik za laboratorijske vježbe, Fakultet elektrotehnike, računarstva i informacijskih tehnologija, Osijek, 2019.

SAŽETAK

Zadatak ovog diplomskog rada je bio izraditi mobilnu aplikaciju za evidenciju prisutnosti na nastavi. Bilo je potrebno izraditi plan izrade, bazu podataka i arhitekturu datoteka. Nadalje, trebalo je unaprijed zadati svrhe i vrste objekata, dodijeliti prava različitim korisnicima te detaljno voditi evidenciju prisutnosti za svaki pojedini kolegij kojeg student sluša.

Nakon postavljanja logičkog dijela, sva pažnja je posvećena korisničkom iskustvu te je dizajnirano sučelje koje može manipulirati bazom podataka i pristupiti svim svojstvima aplikacije.

Ključne riječi: Android mobilna aplikacija, evidencija prisutnosti nastavi, student, nastavnik

ABSTRACT

DEVELOPMENT OF A MOBILE APPLICATION FOR KEEPING RECORDS OF CLASS ATTENDANCE

The premise of this thesis was to create a mobile application for attendance records. It was necessary to create plan for whole process, database and file architecture. Furthermore, it was necessary to predetermine the purposes and types of facilities, to distribute rights to different users in order to keep detailed records of attendance for each individual course that student listens to.

After setting up the logical part, all attention is paid to the user experience and an interface has been designed that can manipulate databases and access all application properties.

Keywords: Android, mobile, application, records, attendance, teaching, student, professor

ŽIVOTOPIS

Petar Nenadić rođen je 12. veljače 1992. godine u Splitu, Republika Hrvatska. 2006. godine je završio osnovnu školu Stjepan Radić u Imotskom nakon čega iste godine upisuje Opću Gimnaziju dr. Mate Ujevića u Imotskom. Poslije završetka srednje škole 2010. godine upisuje Fakultet Elektrotehnike, Strojarstva i Brodogradnje (FESB) u Splitu. U akademskoj godini 2014./2015. prebacuje se na Fakultet Elektrotehnike, Računarstva i Informacijskih Tehnologija (FERIT) u Osijeku gdje 2018. godine završava stručni studij Elektrotehnike, smjer Informatika. Iste godine upisuje studij Razlikovnih obveza na istom fakultetu. 2020. godine upisuje diplomski studij Računarstva, smjer Programsko inženjerstvo.

_____
Potpis

PRILOZI

Prilog 1. Link za aplikaciju: https://github.com/nenadicpetar/Aplikacija-za-vo-enje-evidencije-prisutnosti-na-nastavi

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

IZJAVA O ORIGINALNOSTI RADA

Osijek, 13.12.2022.

Ime i prezime studenta:	Petar Nenadić
Studij:	Diplomski sveučilišni studij Računarstvo
Mat. br. studenta, godina upisa:	D-1147R, 13.10.2020.
Turnitin podudaranje [%]:	11

Ovom izjavom izjavljujem da je rad pod nazivom: **Izrada mobilne aplikacije za vođenje evidencije prisutnosti na nastavi**

izrađen pod vodstvom mentora Izv. prof. dr. sc. Damir Blažević

i sumentora Izv.prof.dr.sc. Tomislav Keser

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.
Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:
Petar Nenadić