

# Mobilna aplikacija E-index

---

**Marinović, Jakov**

**Undergraduate thesis / Završni rad**

**2023**

*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://um.nsk.hr/um:nbn:hr:200:211835>

*Rights / Prava:* [In copyright](#) / [Zaštićeno autorskim pravom.](#)

*Download date / Datum preuzimanja:* **2025-03-10**

*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**

**Mobilna aplikacija E-indeks**

**Završni rad**

**Jakov Marinović**

**Osijek, 2023. Godina**

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA  
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK****Obrazac Z1P - Obrazac za ocjenu završnog rada na preddiplomskom sveučilišnom studiju**

Osijek, 18.09.2023.

Odboru za završne i diplomske ispite

**Prijedlog ocjene završnog rada na preddiplomskom sveučilišnom studiju**

<b>Ime i prezime Pristupnika:</b>	Jakov Marinović
<b>Studij, smjer:</b>	Sveučilišni prijediplomski studij Elektrotehnika i informacijska
<b>Mat. br. Pristupnika, godina upisa:</b>	4704, 22.07.2019.
<b>OIB Pristupnika:</b>	67572976097
<b>Mentor:</b>	izv. prof. dr. sc. Alfonzo Baumgartner
<b>Sumentor:</b>	,
<b>Sumentor iz tvrtke:</b>	
<b>Naslov završnog rada:</b>	Mobilna aplikacija E-index
<b>Znanstvena grana rada:</b>	<b>Procesno računarstvo (zn. polje računarstvo)</b>
<b>Zadatak završnog rad:</b>	[Rezervirano: Jakov Marinović] Napraviti mobilnu aplikaciju koja omogućava bzri prikaz osnovnih informacija o položenim i nepoloženim predmetima određenog studenta, te ostale osnovne podatke o studentu. Podatke je potrebno preuzeti iz odgovarajućeg priređenog resursa na mreži. Podaci bi trebali biti prikazani po uzoru na službene dokumente indeksa kakvi su se prije koristili.
<b>Prijedlog ocjene završnog rada:</b>	Izvrstan (5)
<b>Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:</b>	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 2 bod/boda Jasnoća pismenog izražavanja: 3 bod/boda Razina samostalnosti: 3 razina
<b>Datum prijedloga ocjene od strane mentora:</b>	18.09.2023.
<b>Datum potvrde ocjene od strane Odbora:</b>	24.09.2023.
Potvrda mentora o predaji konačne verzije rada:	Mentor elektronički potpisao predaju konačne verzije.
	Datum:

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA  
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK****IZJAVA O ORIGINALNOSTI RADA**

Osijek, 27.09.2023.

**Ime i prezime studenta:**

Jakov Marinović

**Studij:**

Sveučilišni prijediplomski studij Elektrotehnika i informacijska tehnologija

**Mat. br. studenta, godina upisa:**

4704, 22.07.2019.

**Turnitin podudaranje [%]:**

12

Ovom izjavom izjavljujem da je rad pod nazivom: **Mobilna aplikacija E-index**

izrađen pod vodstvom mentora izv. prof. dr. sc. Alfonzo Baumgartner

i sumentora ,

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:

# SADRŽAJ

1. UVOD .....	1
1.1. Zadatak završnog rada.....	1
2. PREGLED SLIČNIH RJEŠENJA .....	2
2.1. Studomat.....	2
2.2. E-Dnevnik .....	3
2.3. Grade Tracker Pro .....	4
2.4. Grade Pal .....	5
2.5. Moje ocjene .....	6
3. TEORIJA.....	7
3.1. Android.....	7
3.2. Android Studio .....	9
3.3. Java.....	10
3.4. Kotlin programski jezik.....	11
3.5. XML.....	11
3.6. JSON .....	12
4. IZRADA APLIKACIJE E-INDEKS.....	13
4.1. Manifest.....	13
4.2. Glavna aktivnost.....	14
4.3. Aktivnost za izbor semestra .....	16
4.4. Aktivnost za prikaz semestra.....	17
4.5. Upravljanje JSON podacima .....	18
5. PRIKAZ APLIKACIJE E-INDEKS .....	19
LITERATURA.....	24
SAŽETAK.....	25
ABSTRACT .....	26

# 1. UVOD

Tema ovog završnog rada je izrada Android aplikacije koja pomaže pri praćenju aktualnih podataka o studiranju. Naziv aplikacije E-Indeks ukazuje na to da aplikacija predstavlja digitalnu tj. elektroničku verziju knjižice indeks koja sadrži sve glavne informacije o studentu i ostvarene uspjehe na kolegijima. Kako je Internet u današnje vrijeme svima lako dostupan i većina ako ne i svi studenti posjeduju pametne mobilne uređaje, ideja ovog završnog rada je bila prikaz kako bi izgledalo da se fizičke knjižice zamijene mobilnom aplikacijom. Prva od mnogih koristi je ta da se bilo kada i bilo gdje preko mobilnog uređaja može pogledati recimo uspjeh ili ostale informacije o studiranju dok bi za to inače bilo potrebno uvijek nositi knjižicu. Također sam upis ocjena i potpis ocjenjivača za koje se treba organizirati susret profesora i studenta uživo, sada preko aplikacije bi se mogao izvesti puno jednostavnije i sasvim neovisno. Da bi aplikacija mogla prikazati te podatke, mora ih moći čitati iz baze podataka. Stoga je temelj cijelog ovog projekta JSON (eng, *JavaScript Object Notation*) datoteka u kojoj se nalaze svi potrebni podaci o studentu. Svaki student ima svoju unikatnu JSON datoteku te iz tog razloga aplikacija može za izabranog studenta prikazati njegove specifične podatke. U prvom poglavlju je prikazan kratak uvod i opisana osnovna funkcija aplikacije. U drugom poglavlju su navedene slične aplikacije i programi kao rješenja problema praćenja učeničkih i studentskih uspjeha. Kako je za izradu aplikacije korišteno razvojno okruženje Android Studio, u trećem poglavlju je sažet opis što je to Android Studio i ostale tehnologije potrebne za izradu ove Android aplikacije. Nadalje četvrto poglavlje sadrži pregled pojedinih dijelova koda i njihovih funkcija dok peto poglavlje prikazuje rad gotove aplikaciju pokrenute na virtualnom uređaju pomoću emulatora. Te zadnje poglavlje sadrži zaključak ovog završnog rada.

## 1.1. Zadatak završnog rada

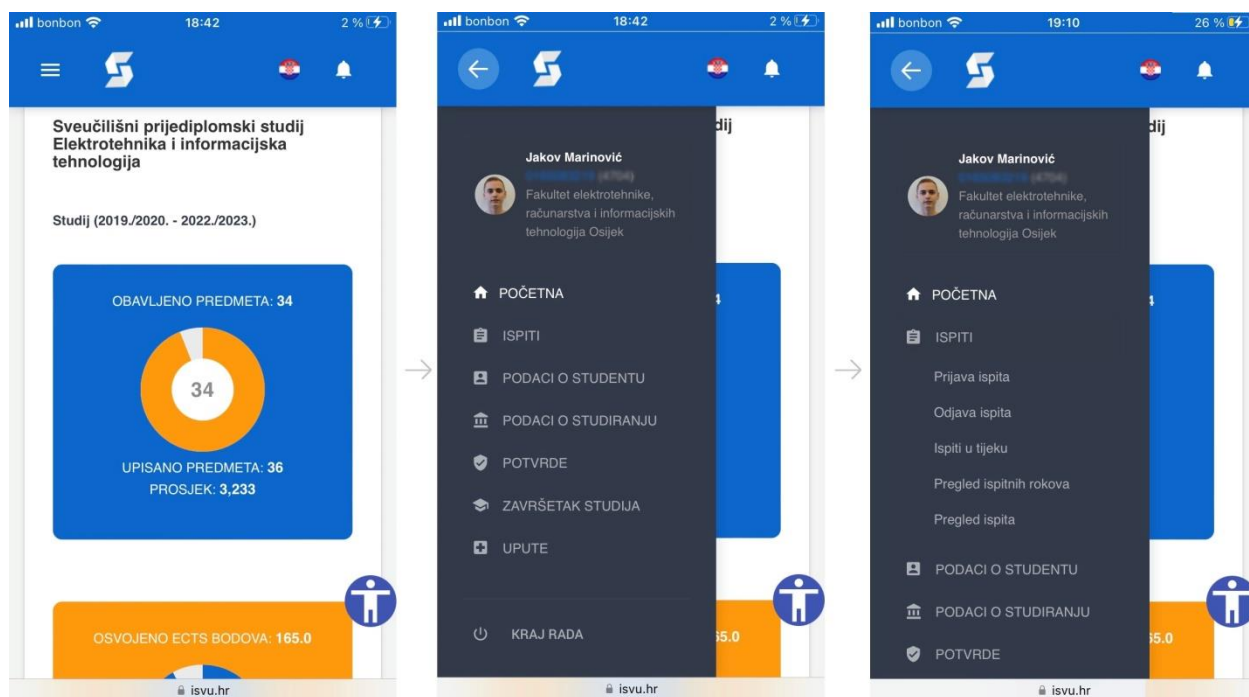
Napraviti mobilnu aplikaciju koja omogućava brzi prikaz osnovnih informacija o položenim i nepoloženim predmetima određenog studenta, te ostale osnovne podatke o studentu. Podatke je potrebno preuzeti iz odgovarajućeg priređenog resursa na mreži. Podaci bi trebali biti prikazani po uzoru na službene dokumente indeksa kakvi su se prije koristili.

## 2. PREGLED SLIČNIH RJEŠENJA

Ovo poglavlje prikazuje nekoliko od trenutno raspoloživih aplikacija i programa za praćenje podataka o studiranju. Uz to dodana je i kratka usporedba s E-indeks aplikacijom za svako od tih rješenja.

### 2.1. Studomat

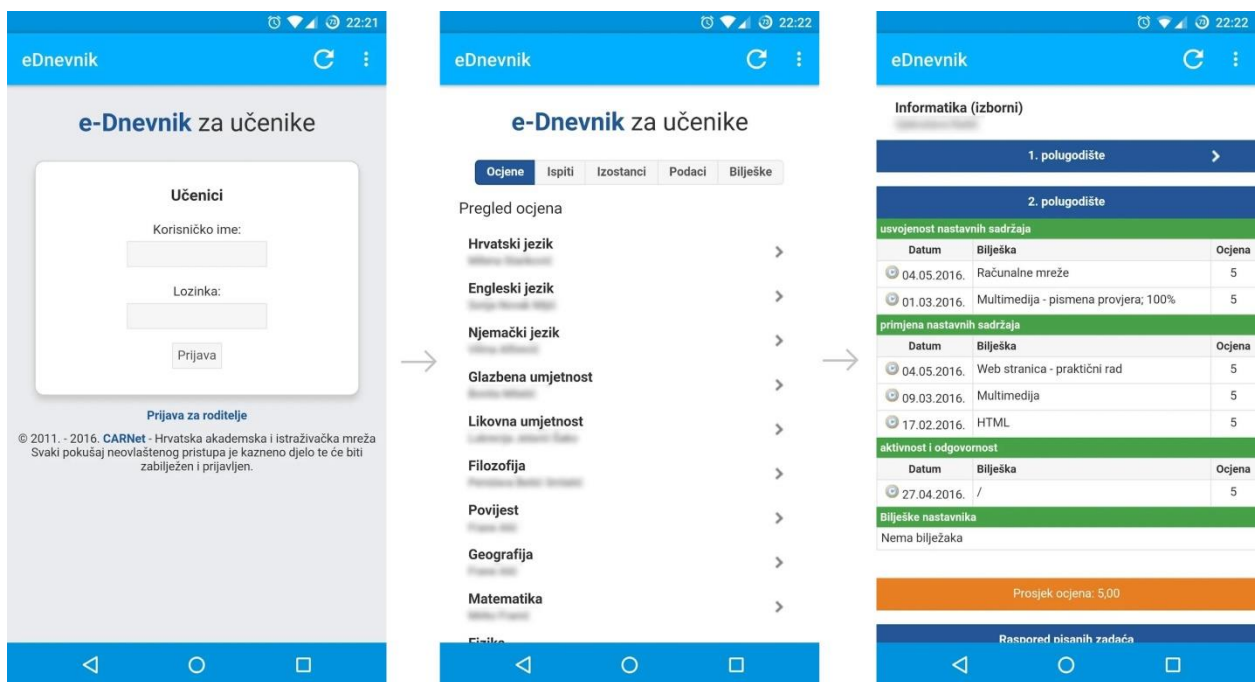
Studomat je web stranica koja nudi studentima jednostavan prikaz informacija tijekom studiranja. Nakon što se otvori stranica, traži prijavu studenta. Svaki student ima svoje korisničko ime i lozinku pomoću kojih ulazi u početni prikaz gdje se može vidjeti koliko je obavljeno predmeta u odnosu na broj upisanih predmeta te prosjek ocjena. Sa strane se nalazi izborna traka gdje su neke od mogućnosti kao što su prijava i odjava ispita, podaci o studentu i podaci o studiranju. Kao takva stranica je vrlo slična aplikaciji koja je tema ovog završnog rada. No aplikacija E-Indeks omogućava brži pristup predmetima jer se može odmah izabrati željeni semestar te će se prikazati predmet koji se traži. [1]



Slika 2.1. Prikaz zaslona web stranice Studomat [1]

## 2.2. E-Dnevnik

E-dnevnik kao web stranica i od nedavno kao aplikacija za mobilne uređaje omogućava učenicima pregled a nastavnicima unos ocjena, bilješki, izostanaka i ostalih bilješki. To je ustvari digitalna inačica knjige koja se zove dnevnik, koju su nastavnici nosili na sat i u istu unosili podatke za svakog učenika. E-dnevnik se danas koristi u više od 1330 škola što je približno 100 posto od svih škola u Hrvatskoj. Isto kao kod Studomata, osoba mora imati korisničko ime i lozinku da bi bio moguć pristup svojim podacima. Budući da je aplikacija E-dnevnik dostupna samo učenicima u osnovnim i srednjim školama, studenti ju ne mogu koristit. Tu mogućnost studentima bi ostvarila aplikacija E-Indeks. [2]

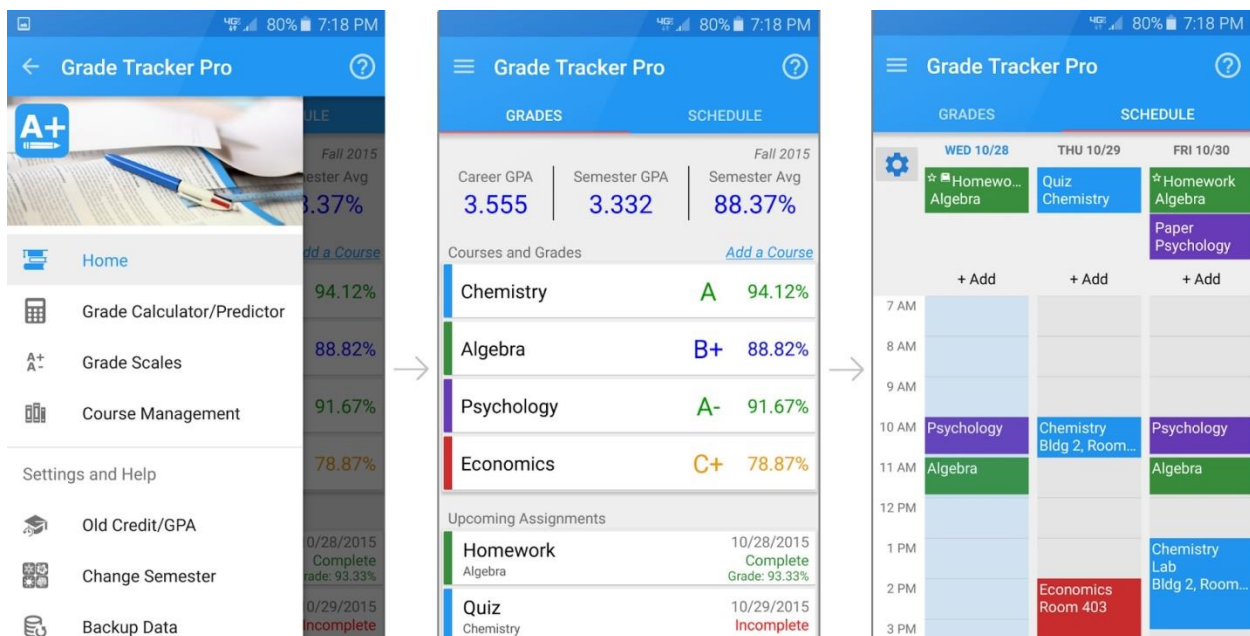


Slika 2.2. Prikaz zaslona aplikacije E-dnevnik [2]



## 2.3. Grade Tracker Pro

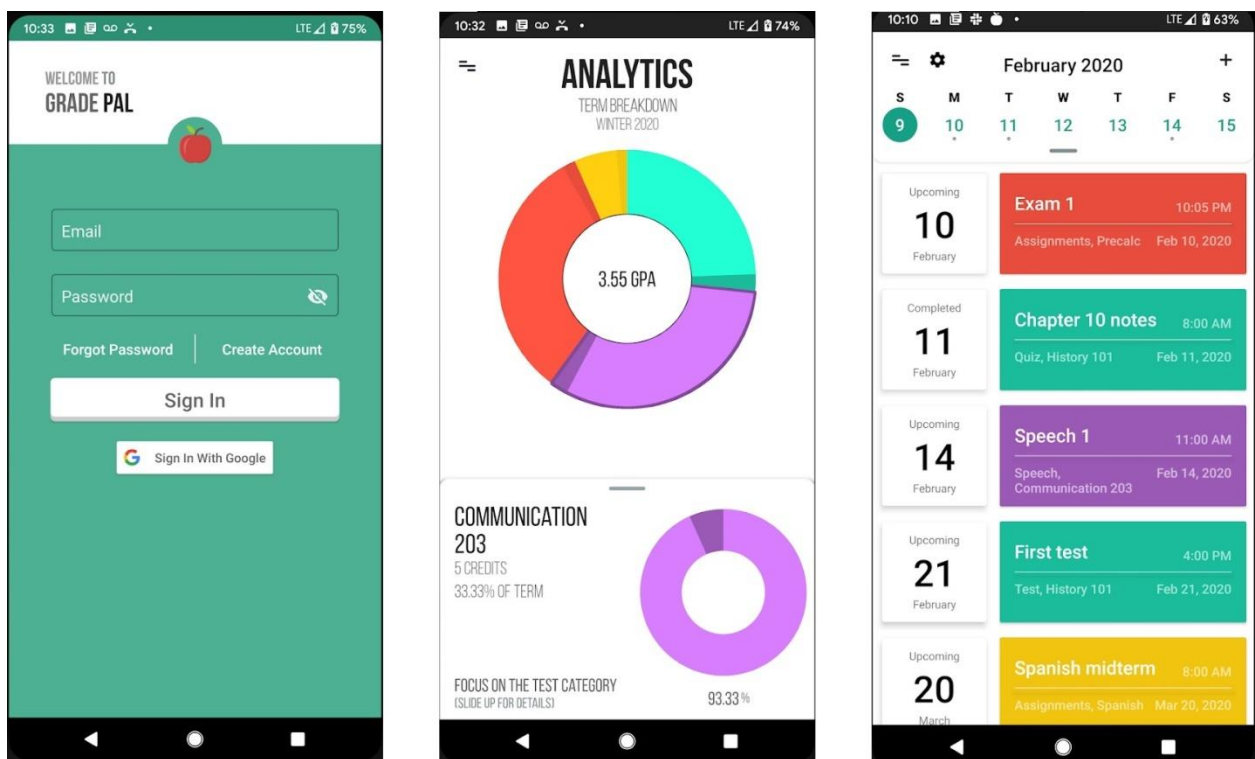
Grade Tracker Pro je jedna od nekoliko sličnih aplikacija na stranom jeziku koja omogućava praćenje akademskog i školskog uspjeha. Glavna razlika je ta što se kod ove aplikacije podaci ne povlače iz nekakve baze podataka već učenik ili student treba sam unositi ocjene u aplikaciju kada položi ispit tj. predmet. Između ostalog postoji planer događaja s kalendarom i zapisivanje bilješki, a unesene ocjene automatski računa u prosjek. Međutim nedostatak ove aplikacije je to što je na engleskom jeziku i što je namijenjena više za američki školski sistem. Shodno tome nije toliko povoljna opcija hrvatskim studentima kao aplikacija za praćenje podataka o studiranju. Zbog toga je E-indeks koncipiran isključivo studentima u Hrvatskoj radi lakšeg korištenja bez potrebe za prilagodbom. [3]



Slika 2.3. Prikaz zaslona aplikacije Grade Tracker Pro [3]

## 2.4. Grade Pal

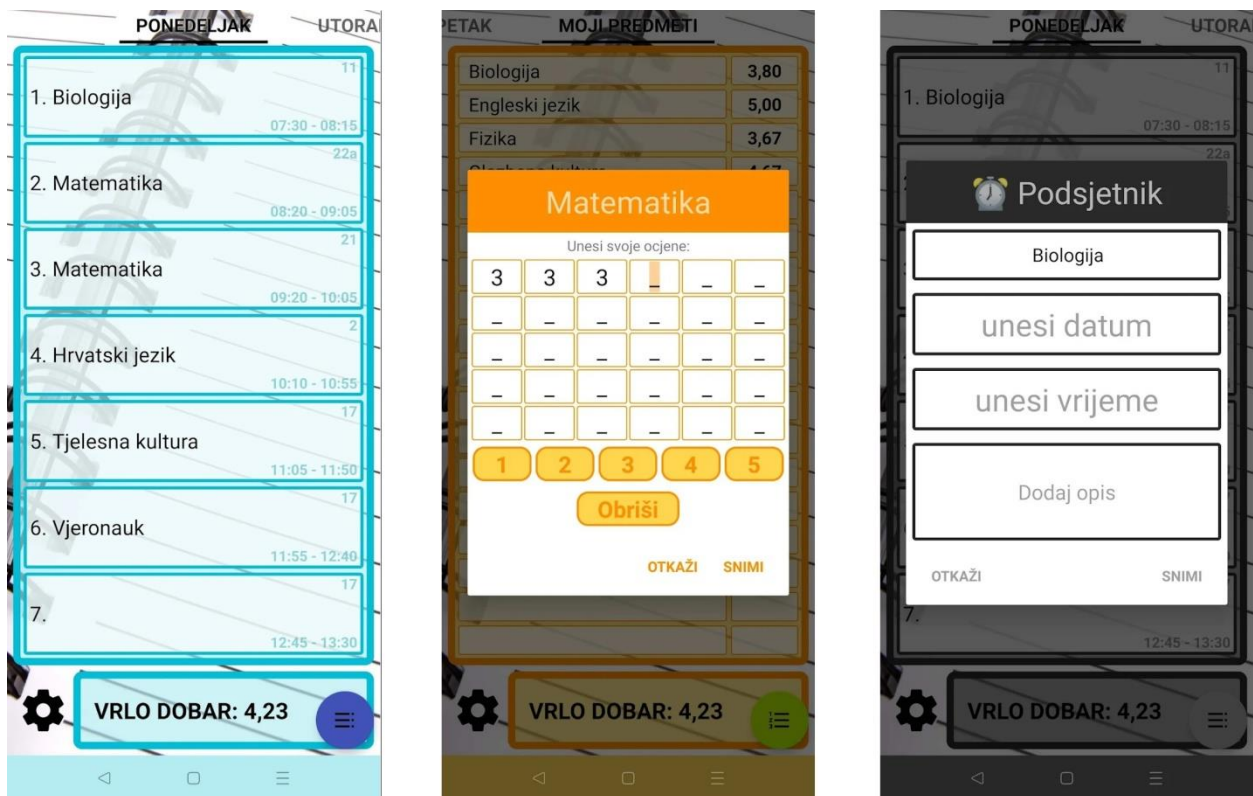
Grade Pal je također mobilna aplikacija koja u cilju ima pojednostaviti praćenje uspjeha studenta. Aplikacija za spremanje podataka koristi mrežu, tako što korisnik prvo treba napraviti vlastiti račun s E-mail adresom i lozinkom, takav sistem osigurava da su podaci zaštićeni. No ocjene ili bilješke nisu povezane sa školskim sustavom te je potrebno sve upisivati u aplikaciju ručno. Nadalje omogućuje kvalitetan analitički prikaz uspjeha na semestru, kalendar s obavijestima te kalkulator ocjena. Kao i ostale strane aplikacije nije često korištena među hrvatskim studentima zbog teže prilagodbe na strani školski sistem bodovanja i tijeka semestara. [4]



Slika 2.4. Prikaz zaslona aplikacije Grade Pal [4]

## 2.5. Moje ocjene

Jedna od popularnijih aplikacija koja se u zadnje vrijeme koristi među učenicima je Moje ocjene. Može se zaključiti po 1+ milijun preuzimanja i preko 7 tisuća recenzija na Google Play trgovini. Kao i ostala slična rješenja za praćenje školskog uspjeha bazira se na tome da prikaže ocjene i računa ukupan prosjek ocjena. Također uz to aplikacija nudi kalendarski podsjetnik i tjedni raspored sati. Vrlo je jednostavna za korištenje, prvo treba postaviti predmete koji se provode za tu akademsku godinu te raspored sati, nakon toga korisnik ručno upisuje ocjene i aplikacija automatski računa prosjek za svaki predmet te prosjek svih ocjena tog učenika. Više je namijenjena učenicima u osnovnim i srednjim školama zbog toga što predmeti sadržavaju veći niz ocjena dok za studente svaki kolegij ima samo jednu finalnu ocjenu. [5]



Slika 2.5. Prikaz zaslona aplikacije Moje ocjene [5]

## 3. TEORIJA

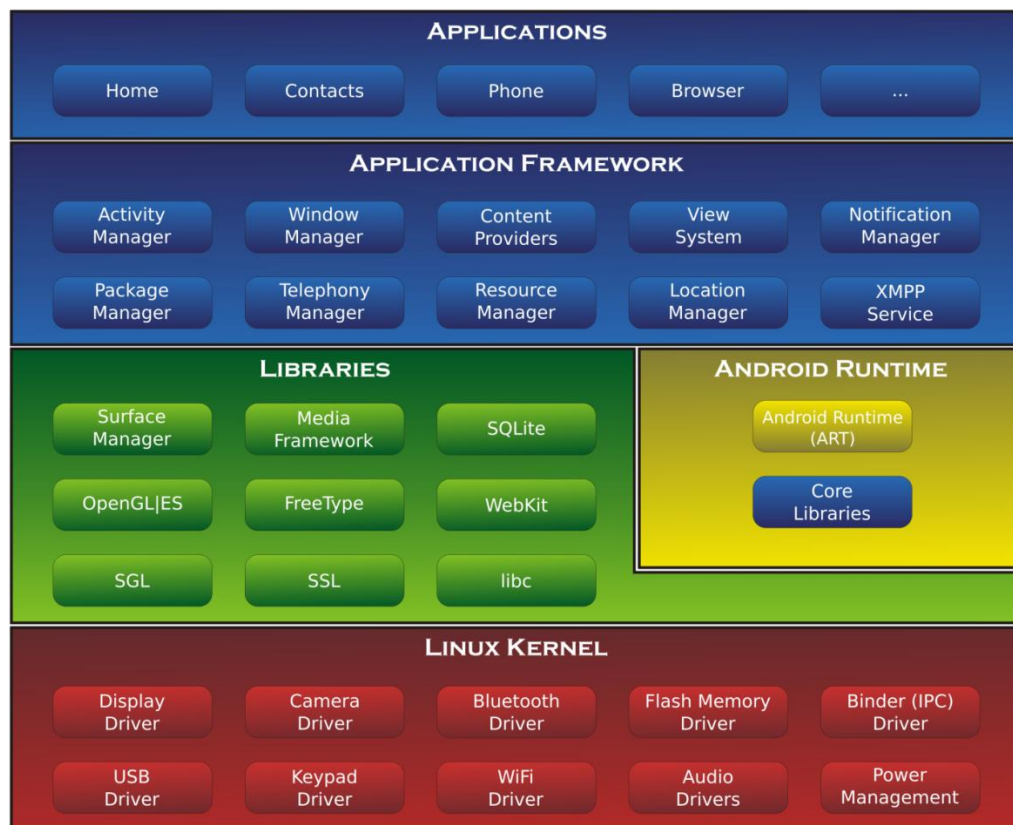
### 3.1. Android

Android je operacijski sustav koji je razvila američka tvrtka Google a baziran je na Linux Kernelu, otvorenog je koda i dizajniran je za uređaje kao što su pametni mobilni telefoni, pametni televizori, tableti i drugo. Andy Rubin, Rich Miner, Nick Sears i Chirs White osnovali su Android 2003. godine kako bi za mobilne uređaje mogli razvijati programe, a zatim je Google tvrtku kupio 2005. godine. Dok je prva beta verzija SDK (eng. *Android Software Development Kit*) za komercijalnu uporabu izdana 2007. godine. Operacijski sustav temeljen je na programskom jeziku C/C++, a zasniva se na Linux 2.6 jezgri. Korisničko sučelje tj. UI (eng. *User Interface*) Androida oslanja se na izravne unose kao što su dodir, pritisak, povlačenje prstom i sl. Uz to sadrži i virtualnu tipkovnicu a korisniku pruža taktilne povratne informacije putem vibracije. Podržava i sve mogućnosti hardvera na primjer žiroskop, kamere, senzore, akcelerometar i ostalo, time povećava potencijal na koji se koristi uređaj. Svaki Android ima početni zaslon takozvani „*Home screen*“ koji je temelj navigacije po operacijskom sustavu jer se od njega polazi nakon pokretanja sustava i odabiremo sljedeću radnju ili aplikaciju. Aplikacije su raspoređene na početnom zaslonu i intuitivno ih otvaramo pritiskom na njihovu ikonu. Također postoji i statusna traka koja se uglavnom nalazi na gornjem rubu zaslona i omogućava brz pristup informacijama recimo status mreže, stanje baterije uređaja, obavijesti i drugo. [6]



*Slika 3.1. Android logo 2023. [6]*

Arhitektura Androida se sastoji od nekoliko slojeva, prvi sloj su aplikacije vidljive korisniku. Napravljene uglavnom pomoću Java programskog jezika. Drugi sloj kao što je prikazano na priloženoj slici (slika 3.2.) je „*Application Framework*“ gdje Android omogućava programerima da izrade inovativne i napredne aplikacije tako što koriste mogućnosti hardvera, pristup informacijama, pokretanje više servisa u pozadini i sl. Nakon toga slijedi sloj „*Libraries*“ u kojem se nalaze glavne biblioteke koje koriste komponente Android operativnog sustava, neke od njih su sistemske C biblioteke, SQLite, OpenGL. Također tu je „*Android Runtime*“ sloj koji kako naziv prilaže omogućava pokretanje aplikacija a čine ga „*Core Libraries*“ što obuhvaća biblioteke programskog jezika Java i Dalvik Virtual Machine koji pokreće aplikacije kao instance virtualnog stroja. Dok zadnji sloj pod nazivom „*Linux Kernel*“ predstavlja Linux verzije 2.6 za glavne procese kao primjerice upravljanje memorijom, upravljanje procesima, upravljanje kamerom...[7]



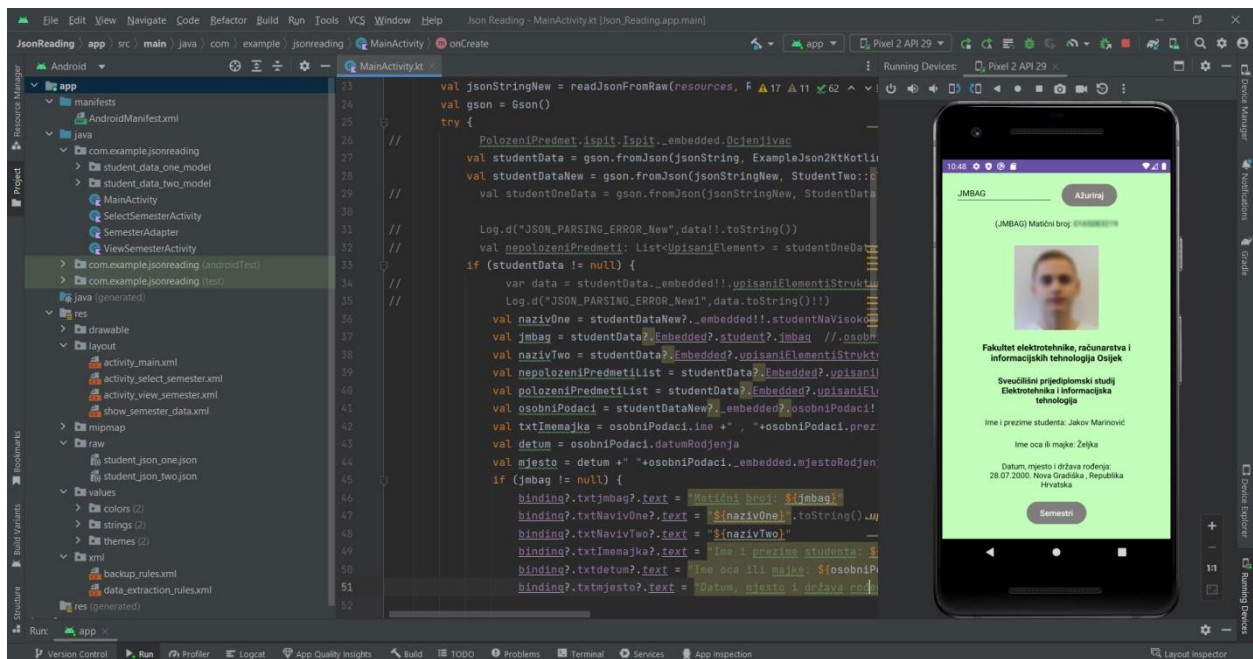
Slika 3.2. Prikaz arhitekture Android operacijskog sustava [7]

## 3.2. Android Studio

Android studio je integrirano razvojno okruženje takozvano IDE (eng. *Integrated Development Environment*) za Google Android platformu kojeg je razvio IntelliJ. Koristi se za izradu Android aplikacija a dostupan je za rad na Windows, MacOS i Linux operacijskim sustavima. Sastoji se od kompilera, debugera i editora za kod ali ima i razne dodatne mogućnosti. Primjerice podrška *Gradle build* sustava koji se koristi za automatizaciju izgradnje, testiranja i implementacije, *Lint* alat za unapređenje kvalitete koda, korisničko sučelje koje na intuitivan način pruža uređivanje dizajna i izgleda aplikacije koristeći *drag and drop* način rada. Između ostalog postoji mogućnost testiranja aplikacije pomoću emulatora pod nazivom AVD (eng. *Android Virtual Device*) koji prikazuje virtualni zaslon uređaja i služi za testiranje funkcionalnosti projekta u izradi. Još neke od mogućnosti su podrška Google *Cloud* platforme, integracija GitHub-a i predlošci koda. Android studio podržava programske jezike od IntelliJ a to su Java, C++, Kotlin s Android 3.0 verzijom pa nadalje, Groovy i drugim jezicima baziranim na JAR (eng. *Java ARchive*). Struktura projekta u Android Studio razvojnom okruženju se sastoji od aplikacijskih modula, biblioteke modula i GAE (eng. *Google App Engine*) modula koji pružaju podršku Google App servisa za aplikacije. Međutim svaki od aplikacijskih modula sadrži sljedeće tri datoteke: manifesti (eng. *Manifests*), java i res. U direktoriju manifesti nalazi se *AndroidManifest.xml* datoteka a java sadrži Kotlin i Java izvorne kodove, uključujući JUnit testni kod. Dok res direktoji obuhvaća sve resurse kao što su UI stringovi i bitmap slike. [8]

Android studio sistemski zahtjevi sustava predlažu isključivo 64-bit Microsoft Windows 8/10/11, najmanje drugu generaciju Intel Core procesora ili svi AMD procesori s podrškom Windows Hypervisor, nadalje preporučeno je 8 GB RAM memorije te minimalno 8GB slobodne memorije na disku za instalaciju kompletnog razvojnog okruženja i emulatora, uz to rezolucija zaslona ne smije biti manja od 1280x800. [9]





Slika 3.3. Prikaz korisničkog sučelja programa Android studio

### 3.3. Java

Java je popularan objektno-orijentirani programski jezik i softverska platforma koja se koristi na milijardama uređaja uključujući pametne mobilne telefone, računala, igraće konzole, tablete, medicinske uređaje i drugo. Programski jezik izumio je James Goslin 1991. godine dok je od strane Sun Microsystems predstavljen 1995. godine. Sintaksa i pravila Jave su bazirani na C i C++ jeziku. Glavni cilj pri dizajniranju jezika je bilo da se pojednostavi prebacivanje koda sa različitih uređaja recimo laptopa na mobilni uređaj, te da uz to bude robusan, objektno-orijentiran, dinamičan itd.. Međutim važno je spomenuti da se Java za razliku od JavaScript-a treba kompajlirati ali zato je moguće pokretanje bilo gdje dok se JavaScript može pokretati samo na web preglednicima. Da bi izradili aplikaciju koristeći Java programski jezik potreban je JDK (eng. Java Development Kit) koji je dostupan za Windows, MacOS i Linux. Nakon što je kod napisan, kompajler pretvara taj program u takozvani Java bytecode gdje se nalaze instrukcije za JVM (eng. Java Virtual Machine) koji čini glavni dio JRE (eng. Java Runtime Environment). Taj proces omogućuje da Java kod može biti pokrenut bilo gdje bez modifikacija. [10]

### 3.4. Kotlin programski jezik

Kotlin je statički pisani programski jezik otvorenog koda (eng. *open source*) koji podržava objektno-orijentirano i funkcionalno programiranje. Razvijen od strane JetBrains te podržavan od Google-a za razvoj Android-a, a prva verzija se pojavila 2011. godine. Sintaksa i koncept Kotlin-a je sličan ostalim programskim jezicima a naročito C#, Java, Scala i drugima, što znači da nije potrebna velika prilagodba prilikom prelaska s drugih jezika. Samim time Kotlin-u ide velika korist što gradi na osnovu desetljeća inovacija i razvoja tih programskih jezika. Dizajniran je s obzirom na to da je moguća međusobna suradnja s Javom, jer kod pisan u Javi se može pozivati bilo u Kotlinu ili Javi. Shodno tomu ako je potrebno omogućava da projekt odjednom sadrži i Kotlin i Java kod. [11]

### 3.5. XML

XML (eng. *Extensible Markup Language*) je jednostavni proširivi jezik za obilježavanje tj. označavanje podataka gdje tekstualni format omogućava prikaz strukturiranih informacija. Treba istaknuti da u XML-u značenje podataka i strukturu definiraju oznake, u tom smislu označava što podaci jesu. Za razliku od HTML-a (eng. *Hypertext Markup Language*) čije oznake opisuju izgled i dojam podataka. Još jedna od glavnih razlika je ta što je HTML ograničen na definirani skup oznaka jednakih za sve korisnike. Dok XML omogućuje stvaranje bilo kakve oznake koja je potreba za označavanje podataka i strukture tih podataka. [12]



### 3.6. JSON

JSON je tekstualni format otvorenog standarda dizajniran za čitljiv i lako razumljiv prijenos i pohranjivanje podataka. Početkom 2000-ih Douglas Crockford je koncipirao originalni format i poslao prvu JSON poruku, a takva datoteka JSON formata se prepoznaje ekstenzijom .json. Danas je uobičajeni format s raznolikom upotrebom u elektroničkoj razmjeni podataka primjerice web aplikacije s poslužiteljima. JSON format podataka je neovisan o jeziku i izveden je iz JavaScript-a međutim mnogi napredni programski jezici uključuju kod za generiranje i analizu podataka u JSON formatu. Osnovni tipovi podataka koji se koriste u JSON formatu su broj, tekst (eng. *String*), logički tip podatka (eng. *Boolean*), polje (eng. *Array*) u kojemu je sortirani niz podataka, također objekt (eng. *Object*) gdje se nalazi nesortirani niz oblika ključ/vrijednost parova i posljednji tip podatka je null vrijednost koja predstavlja praznu vrijednost. JSON format se počinje sve češće koristiti za prijenos strukture podataka u odnosu na XML zbog lakšeg pisanja i razumljivog čitanja koda i uz to također JSON se prevodi kroz standardnu JavaScript funkciju dok XML kroz XML prevoditelj. Na slici (*slika 3.4.*) je prikazan referentni primjer u kojem se može vidjeti razlika između dva koda. [13]

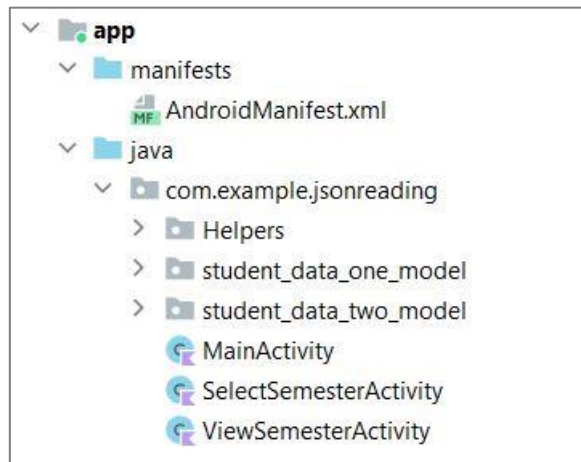
```
1 (JSON)
2 "student":
3 {
4     "ime": "Jakov",
5     "prezime": "Marinovic",
6     "godinaRodjenja": "2000"
7 }
```

```
1 (XML)
2 <student>
3     <ime> Jakov </ime>
4     <prezime> Marinovic </prezime>
5     <godinaRodjenja> 2000 </godinaRodjenja>
6 </student>
```

Slika 3.4. Primjer pisanja JSON i XML koda

## 4. IZRADA APLIKACIJE E-INDEKS

U ovom poglavlju opisani su neki od glavnih dijelova aplikacije uz prikaz programskog koda. Prvo će biti prezentiran AndroidManifest.xml datoteka, naredna potpoglavlja će posebno opisivati po jednu od glavne tri aktivnosti tj. zaslona aplikacije redom, to je MainActivity, SelectSemesterActivity te ViewSemesterActivity. Tim redoslijedom se i korisnik susreće s prikazom zaslona pri korištenju aplikacije. Struktura je prikazana na slici (Slika 4.1.).



Slika 4.1. Prikaz manifesta i aktivnosti

### 4.1. Manifest

AndroidManifest.xml datoteka je vrlo bitna datoteka za cijeli projekt jer sadrži metapodatke i opisuje strukturu aplikacije. U njoj su definirane aktivnosti (eng. *Activity*) uz ostale podatke. Na slici (Slika 4.2.) prikazan je kod gdje su tri glavne aktivnosti definirane te da je na MainActivity postavljen Intent-Filter za glavnu aktivnost aplikacije pri pokretanju.

```
<activity
    android:name=".ViewSemesterActivity"
    android:exported="false" />
<activity
    android:name=".SelectSemesterActivity"
    android:exported="false" />
<activity
    android:name=".MainActivity"
    android:exported="true">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

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

Slika 4.2. Prikaz definiranja aktivnosti u manifest kodu

Između ostalog u ovom manifestu je definirano dopuštenje (eng. *Permission*) za korištenje interneta prikazano na slici (Slika 4.3.).

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

Slika 4.3. Prikaz definiranja dopuštenja u manifest kodu

## 4.2. Glavna aktivnost

U glavnoj aktivnosti tzv. MainActivity korisnik prvo ima mogućnost unošenja svog JMBAG-a te se na temelju toga dohvaćaju JSON podaci i fotografija studenta što je vidljivo na slikama (Slika 4.4. i Slika 4.5.). Za učitavanje slike korištena je Glide biblioteka koja pojednostavljuje proces učitavanja i prikazivanja slika iz vanjskih izvora. Dok za dohvaćanje JSON podataka preko HTTP (eng. *HyperText Transfer Protocol*) zahtjeva korištena je Fuel biblioteka.

```
private fun fetchImage(jmbag: String) {
    Glide.with( activity: this).load( string: "https://mrkve.etfos.hr/slike/studenti/"+jmbag+".jpg").into(binding?.imgUser!!);
}
```

Slika 4.4. Prikaz dohvaćanja slike koristeći Glide biblioteku

```
private fun fetchJson(jmbag: String): String = runBlocking { this: CoroutineScope
    Fuel.get( url: "https://mrkve.etfos.hr/mycache/studentv2jmbag"+ jmbag + ".json").body
}
```

Slika 5.5. Prikaz dohvaćanja podataka koristeći Fuel biblioteku

Nadalje svojstvo za povezivanje prikaza korišten je tzv. View Binding u Kotlin-u koji nakon što je omogućen generira binding klasu za svaku xml datoteku u tom modulu. Slika 4.6. prikazuje korištene binding klase. Na primer prvi postavlja text „(JMBAG) Matični broj:“ uz pridruženu vrijednost varijable jmbag, na id pod nazivom txtjmbag u xml-u.

```
binding?.txtjmbag?.text = "(JMBAG) Matični broj: ${jmbag}"
binding?.txtNavivOne?.text = "${nazivOne}"
binding?.txtNavivTwo?.text = "${nazivTwo}"
binding?.txtImemajka?.text = "Ime i prezime studenta: ${txtImemajka}"
binding?.txtdetum?.text = "Ime oca ili majke: ${osobniPodaci.imeMajka}"
binding?.txtmjesto?.text = "Datum, mjesto i država rođenja: ${mjesto}"
```

Slika 4.6. Prikaz binding koda u glavnoj aktivnosti

Također primjer u xml datoteci id naziva txtjmbag prepoznat je element te se ispod nalaze metapodaci o izgledu, položaju, margini, boji teksta itd. tog podatka. Prikazan textView id-a txtjmbag na slici (Slika 4.7.)

```
<TextView
    android:id="@+id/txtjmbag"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="24dp"
    android:text="Maticni broj: 00000000"
    android:textColor="#000000"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/jmbagInputText" />
```

Slika 4.7. Prikaz textView koda iz xml datoteke

Isto tako dodan je binding za id btnNext što predstavlja gumb pod nazivom „Semestri“ gdje koristeći metodu setOnClickListener omogućuje da se zaslona aplikacije prebaci na sljedeću aktivnost u ovom slučaju SelectSemesterActivity. (Slika 4.8.)

```
fun goToSemesterChoose()
{
    binding?.btnNext?.setOnClickListener()
    { it: View!
        startActivity(Intent( packageContext: this, SelectSemesterActivity::class.java))
    }
}
```

Slika 4.8. Prikaz korištenja setOnClickListener metode

Dok za funkcionalnost nakon unošenja različitih JMBAG brojeva, pri spremanju podataka preko tog broja korišten je Shared Preference API. Jer taj broj opisuje svakog studenta, jedinstven je pa ga koristimo kao ključ (eng. Key) pri pozivu sharedPreferences. (Slika 4.9.)

```
fun update()
{
    binding?.updateButton?.setOnClickListener()
    { it: View!
        val jmbag = binding?.jmbagInputText?.text?.toString()
        val sharedPreferences = getSharedPreferences( name: "PREFERENCE_NAME", Context.MODE_PRIVATE)
        var editor = sharedPreferences.edit()
        editor.putString("jmbag", jmbag)
        editor.commit()
        mainfun(jmbag)
    }
}
```

Slika 4.9. Prikaz korištenja sharedPreferences

### 4.3. Aktivnost za izbor semestra

U ovoj aktivnosti pod imenom `SelectSemesterActivity` je opisan zaslon koji prikazuje tipke za odabir semestra pomoću metode `setOnClickListener` koja je opisana ranije. Slika (Slika 4.10.) prikazuje dio koda iz te aktivnosti. Uz to postoji funkcija `moveToSemester` koja će pokrenuti sljedeću aktivnost te prosljeđuje broj semestra što se vidi na slici ispod (Slika 4.11.).

```
private fun setOnClickListeners() {
    semesterBinding.btnSemeterOne.setOnClickListener()
    { it: View!
        moveToSemester( semesterNo: 1)
    }
    semesterBinding.btnSemeterTwo.setOnClickListener()
    { it: View!
        moveToSemester( semesterNo: 2)
    }
    semesterBinding.btnSemeterThree.setOnClickListener()
    { it: View!
        moveToSemester( semesterNo: 3)
    }
    semesterBinding.btnSemeterFour.setOnClickListener()
    { it: View!
        moveToSemester( semesterNo: 4)
    }
    semesterBinding.btnSemeterFive.setOnClickListener()
    { it: View!
        moveToSemester( semesterNo: 5)
    }
    semesterBinding.btnSemeterSix.setOnClickListener()
    { it: View!
        moveToSemester( semesterNo: 6)
    }
}
```

Slika 4.10. Prikaz korištenja `setOnClickListener` iz aktivnosti za izbor semestra

```
private fun moveToSemester(semesterNo: Int)
{
    startActivity(Intent( packageContext: this, ViewSemesterActivity::class.java).putExtra( name: "semesterNo", semesterNo.toString()))
}
```

Slika 4.11. Prikaz funkcije `moveToSemester`

## 4.4. Aktivnost za prikaz semestra

Kod aktivnost za prikaz pojedinog semestra korišten je RecyclerView koji omogućuje dinamično stvaranje liste. U ovom projektu se odnosi na to da svaki kolegij može prikazati na zaslonu a uz to poboljšava efikasnost jer kolegiji koji se trenutno ne vide na zaslonu neće biti učitani dok za to ne bude potrebe. Na slici (*Slika 4.12.*) vidi se kako je implementiran RecyclerView u xml datoteci.

```
<androidx.recyclerview.widget.RecyclerView
    android:id="@+id/rvSemesterData"
    android:layout_width="match_parent"
    android:layout_height="0dp"
    tools:listitem="@layout/show_semester_data"
    app:layoutManager="androidx.recyclerview.widget.LinearLayoutManager"
    app:layout_constraintTop_toBottomOf="@id/txtSemester"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    android:layout_marginStart="22dp"
    android:layout_marginEnd="22dp" />
```

*Slika 4.12. Prikaz korištenja RecyclerView*

Nadalje kako je postavljen sharedPreferences u glavnoj aktivnosti, ovdje u aktivnosti za prikaz semestra nalazi se funkcija fetchJsonData koja dohvati JMBAG i proslijedi broj na fetchJson. Tu je prikazana i zamučena default vrijednost broja koja je bila postavljena u svrhu testiranja pri prvom korištenju aplikacije vidljivo na slici (*Slika 4.13.*)

```
private fun fetchJsonData(): String {
    val sharedPreferences = getSharedPreferences( name: "PREFERENCE_NAME", Context.MODE_PRIVATE)
    val json = sharedPreferences.getString("jmbag", "XXXXXXXXXX")

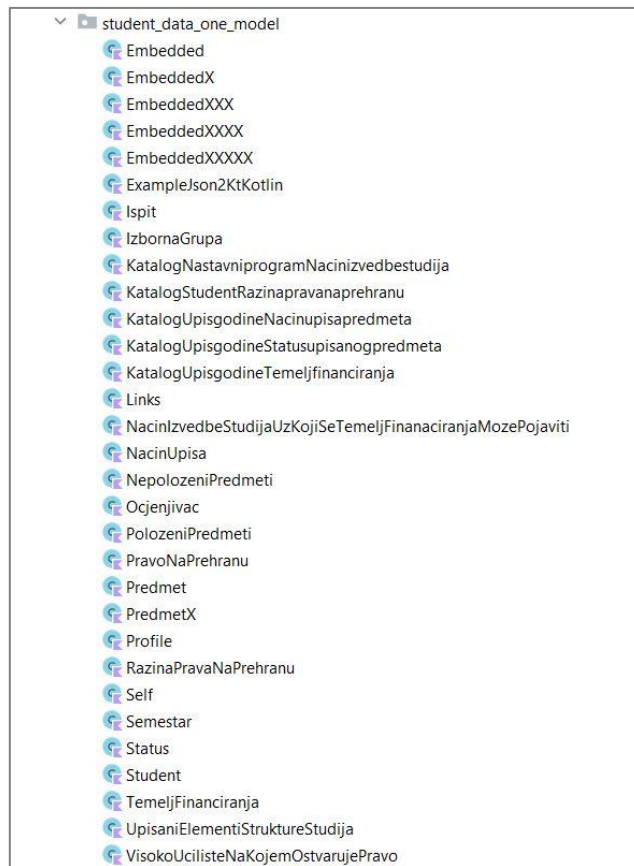
    return fetchJson(json!!)
}
```

*Slika 4.13. Prikaz dohvata podataka iz sharedPreferences*

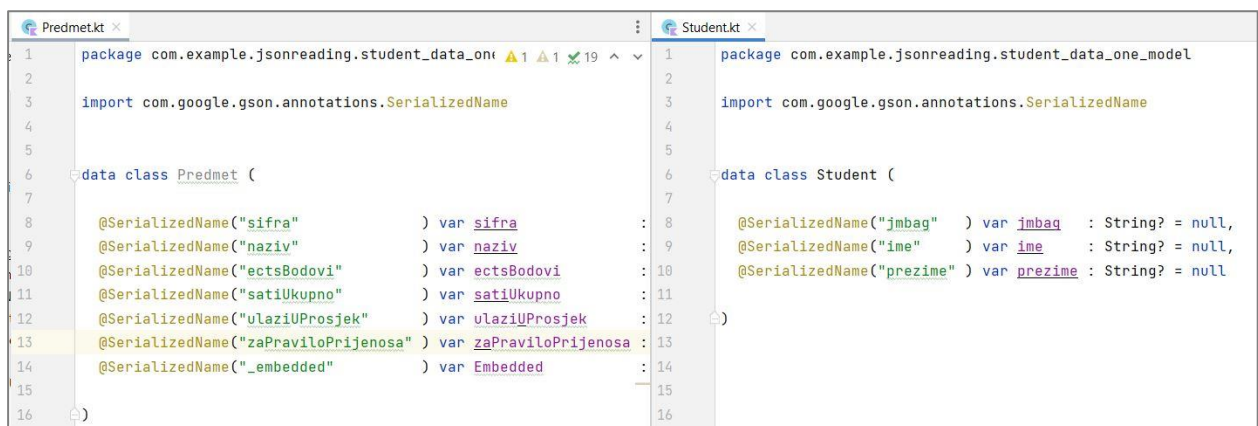


## 4.5. Upravljanje JSON podacima

Automatski generiranje Kotlin klase iz JSON formata datoteke pomoću jednog od plug-ina prezentirano na slici 4.14. dok su za primjer prikazane dvije klase koje su korištene u ovom projektu na slici 4.15.



Slika 4.14. Prikaz svih generiranih klasa iz JSON datoteke



Slika 4.15. Prikaz podataka dvije otvorene klase korištenih u projektu

## 5. PRIKAZ APLIKACIJE E-INDEKS

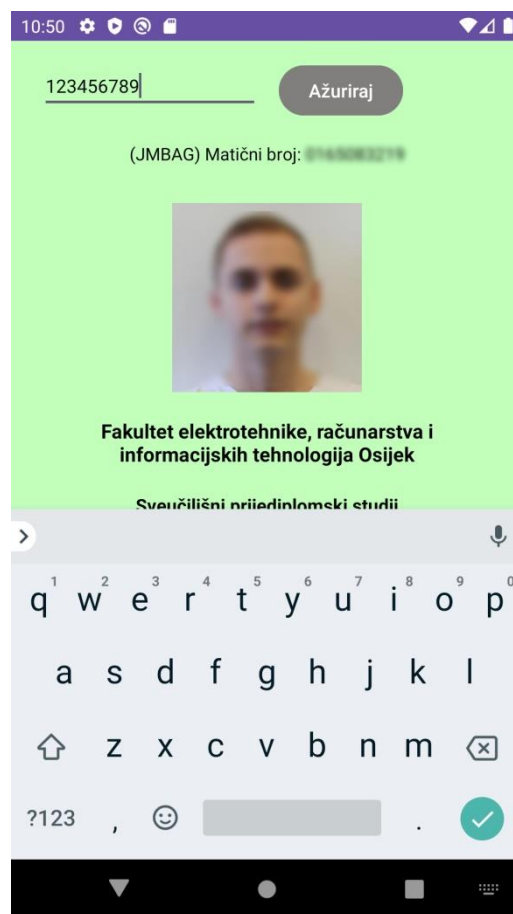
U ovom poglavlju će biti ukratko opisani osnovni zasloni i funkcionalnost same aplikacije.

Prvi zaslon (*slika 5.1.*) prikazuje najbitnije podatke o studentu, inspiracija je uzeta iz fizičke indeks knjižice gdje je također početni zaslon popunjen glavnim podacima kao što je naziv fakulteta, smjer obrazovanja, matični broj, ime i prezime studenta itd... Shodno tomu na ovom zaslonu vidljiva je slika studenta.

Pri samom vrhu zaslona nalazi se traka za tekstualni unos JMBAG-a (Jedinstveni Matični Broj Akademskog Građana), klikom na traku pojavit će se virtualna tipkovnica (*slika 5.2.*). Time svaki student nakon što prvi puta upiše vlastiti JMBAG tj. matični broj, podaci će se s mreže dohvatiti te prikazati. Svako iduće otvaranje aplikacije neće biti potreban ponovni unos već će aplikacija zapamtiti vaš matični broj te učitati aktualne podatke vezane za njega.



*Slika 5.1. Prikaz prvog zaslona*

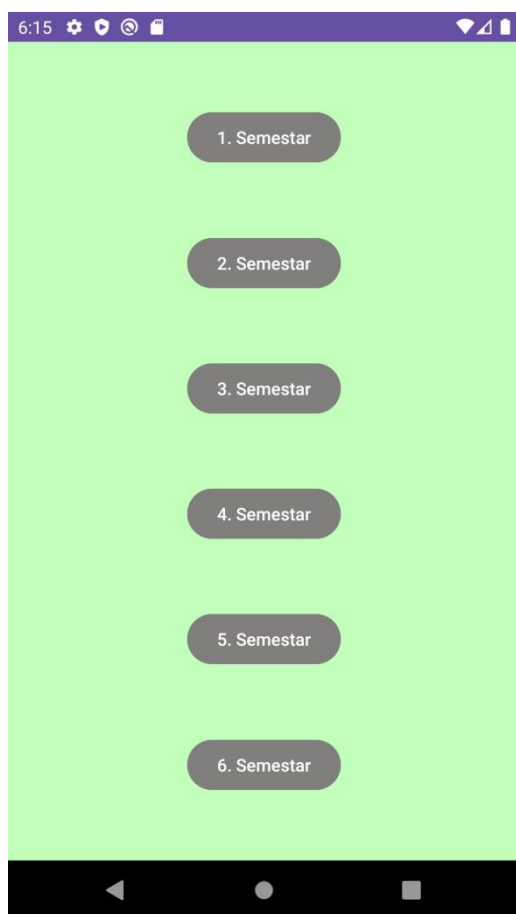


*Slika 5.2. Prikaz prvog zaslona s virtualnom tipkovnicom*

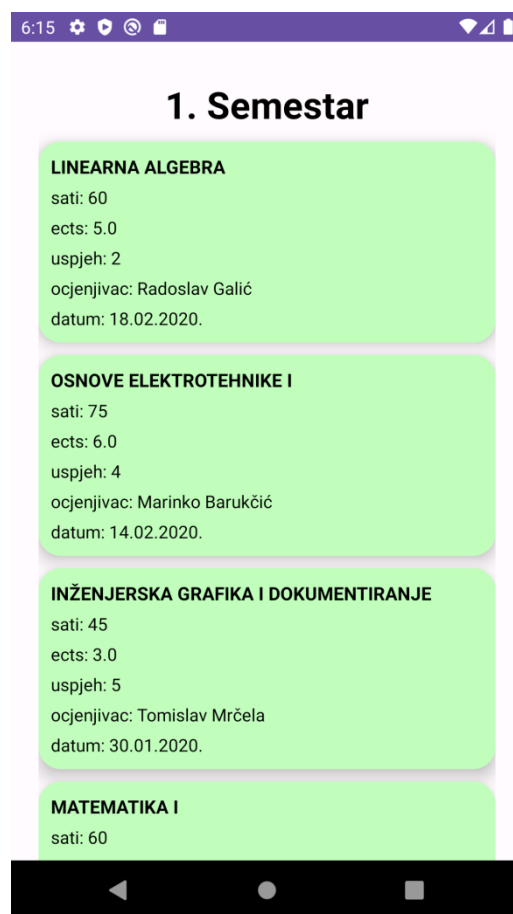


Također na dnu početnog zaslona postavljen je gumb koji vodi na idući prikaz za izbor jednog od ponuđenih semestara (*slika 5.3.*). Tim načinom grupiranja je olakšan pronalazak pojedinog kolegija jer nema potrebe za listanjem kroz sve kolegije od svih semestara skupa.

Nadalje kada pritisnemo na jedan od ponuđenih semestara u ovom slučaju na slici (*slika 5.4.*) je snimljen prvi, gdje su naravno prikazani samo kolegiji sa tog semestra. Isto kao i kod početnog zaslona, struktura na zaslonu semestra je slična onoj kao u indeks knjižici, gdje su upisani kolegiji prikazani jedan iznad drugog i o svakom su vidljivi ključni podaci. Tako da u slučaju da je položen kolegij vidljivi su podaci o ostvarenom uspjehu, tko je ocjenjivač i datum kada je položen ispit.

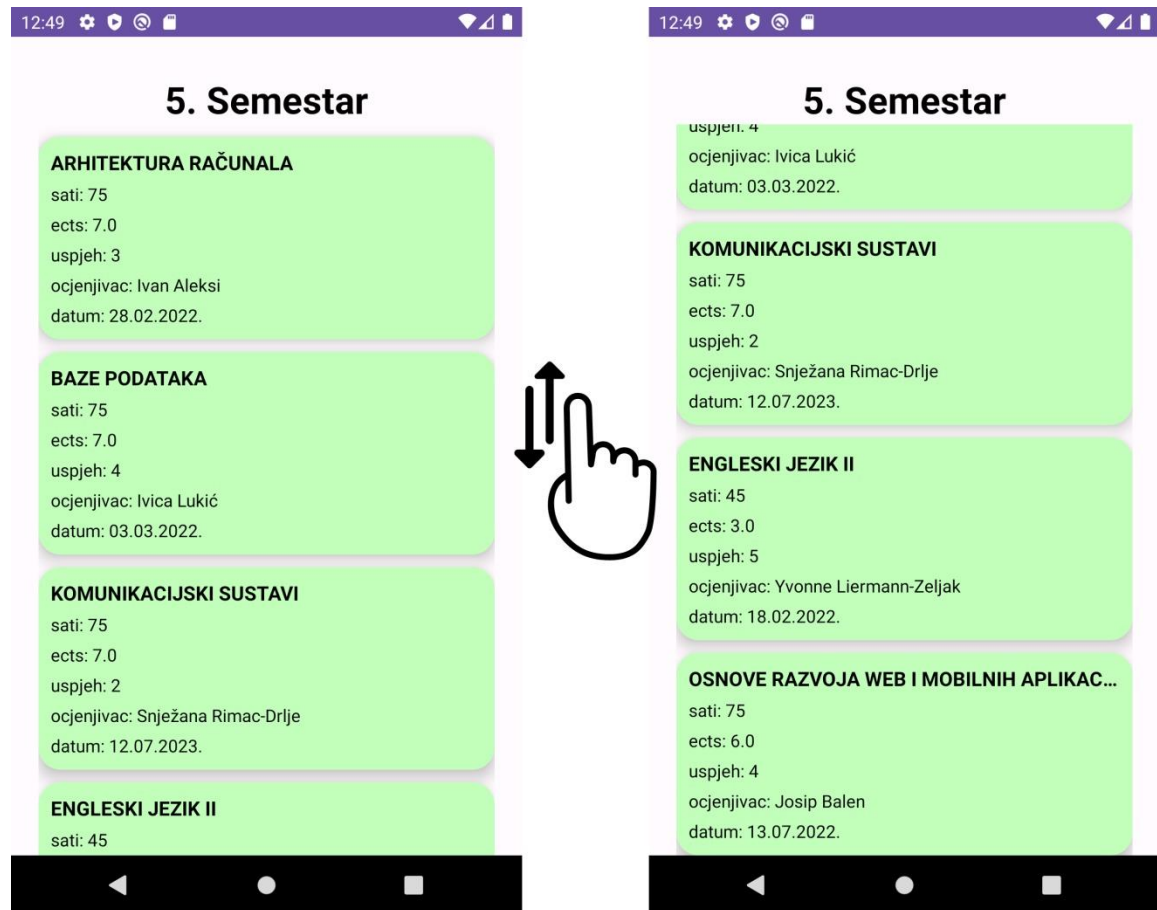


*Slika 5.3. Prikaz drugog zaslona*



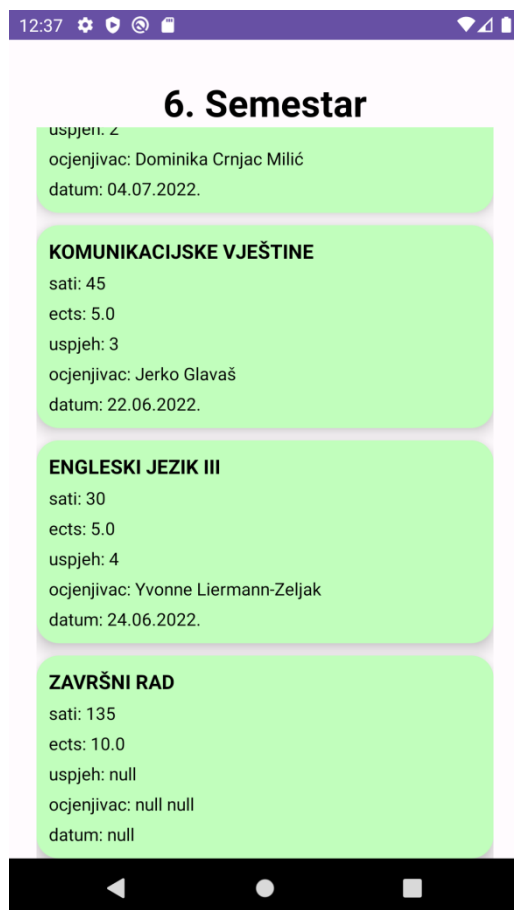
*Slika 5.4. Prikaz zaslona prvog semestra*

Povlačenjem prsta prema gore ili dolje tzv. *scroll* omogućuje fluidno i intuitivno kretanje kroz prikaz jednog semestra. Tako da neovisno koliko kolegija ima u jednom semestru mogu se vidjeti svi kao što je prikazano na slici (slika 5.5.).



Slika 5.5. Prikaz kompletnog petog semestra korištenjem radnje „scroll“

Budući da će kod položenih kolegija biti prikazani svi podaci jer su dostupni, kod nepoloženih kolegija nema podataka o ostvarenom uspjehu, ocjenjivaču te datumu kada je ispit položen, zbog toga ostaju prazne vrijednosti što je prezentirano na slici (slika 5.6.) s kolegijem pod nazivom Završni rad.



Slika 5.6. Prikaz zaslona semestra s nepoloženim kolegijem

## 6. ZAKLJUČAK

U ovom projektu postignuti su svi ciljevi zadanog zadatka završnog rada, a to je bilo napraviti funkcionalnu mobilnu aplikaciju koja omogućava brz i pregledan prikaz informacija o studentu te njegovim položenim i nepoloženim predmetima. A ti podaci su preuzeti sa resursa na mreži preko studentovog JMBAG-a. Testirano je više različitih JMBAG brojeva od pojedinih studenata uz njihovo dopuštenje, te je utvrđeno da aplikacija funkcionira bez problema. Ideja aplikacije je da studentima olakša praćenje uspjeha na fakultetu tako da im nije više potrebna knjižica već su sve najbitnije informacije dostupne na mobilnom uređaju. Također dizajn aplikacije je uspješno napravljen tako da bude jednostavan i intuitivan te da jasno priliči fizičkoj indeks knjižici studenta. Rad u programu Android Studio zahtijevao je dobro poznavanje programskog jezika Java i Kotlin no nije bilo nikakvih poteškoća zahvaljujući sofisticiranom korisničkom sučelju razvojnog okruženja i lako dostupnim pomoćnim rješenjima raspoloživim na različitim edukativnim internetskim stranicama. Pri izradi aplikacije pomoglo je i stečeno znanje s pojedinih kolegija. Naravno postoji mogućnost napretka aplikacije daljnjim dorađivanjem vizualnog izgleda i po potrebi prikazati više informacija o studentu i predmetima iz JSON datoteka. Aplikacija bi bila namijenjena prvobitno studentima na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija (FERIT) Osijek zbog toga što je izrađena prema uzorku JSON datoteke za studente sa FERIT-a. No također je moguće proširiti aplikaciju za korištenje studentima sa drugih fakulteta uz značajnije dodatke. Naime to nije bilo u cilju ovog zadatka ali dakako da bi poboljšalo iskoristivost aplikacije.

## LITERATURA

- [1] ISVU, Studomat,  
<https://www.isvu.hr/studomat/hr/index> [7. rujna 2023.]
- [2] Google Play, E-dnevnik,  
[https://play.google.com/store/apps/details?id=com.carnet.ocjene\\_mobile](https://play.google.com/store/apps/details?id=com.carnet.ocjene_mobile) [7. rujna 2023.]
- [3] Google Play, Grade Tracker Pro,  
<https://play.google.com/store/apps/details?id=com.NsouthProductions.gradetrackerpro>  
[7. rujna 2023.]
- [4] Google Play, Grade Pal,  
[https://play.google.com/store/apps/details?id=com.gradetracker.jguttromson.graderelease&hl=en\\_US](https://play.google.com/store/apps/details?id=com.gradetracker.jguttromson.graderelease&hl=en_US) [7. rujna 2023.]
- [5] Google Play, Moje ocjene,  
<https://play.google.com/store/apps/details?id=com.shoyo.ocene&hl=hr> [7. rujna 2023.]
- [6] Wikipedia, Android (operating system),  
[https://en.wikipedia.org/wiki/Android\\_\(operating\\_system\)](https://en.wikipedia.org/wiki/Android_(operating_system)) [8. rujna 2023.]
- [7] Android-pit, Arhitektura Androida  
<http://android-pit.blogspot.com/2012/01/arhitektura-androida.html> [8. rujna 2023.]
- [8] Wikipedia, Android Studio,  
[https://en.wikipedia.org/wiki/Android\\_Studio](https://en.wikipedia.org/wiki/Android_Studio) [8. rujna 2023.]
- [9] Android Developers, Install Android Studio,  
<https://developer.android.com/studio/install> [8. rujna 2023.]
- [10] IBM, What is Java?  
<https://www.ibm.com/topics/java> [8. rujna 2023.]
- [11] Wikipedia, Kotlin (programming language),  
[https://en.wikipedia.org/wiki/Kotlin\\_\(programming\\_language\)](https://en.wikipedia.org/wiki/Kotlin_(programming_language)) [9. rujna 2023.]
- [12] Microsoft, XML za početnike,  
<https://support.microsoft.com/hr-hr/office/xml-za-po%C4%8Detnike-a87d234d-4c2e-4409-9cbc-45e4eb857d44> [9. rujna 2023.]
- [13] Start-it, JSON,  
<https://start-it.com.hr/json/> [9. rujna 2023.]

## SAŽETAK

Cilj ovog završnog rada bio je napraviti Android mobilnu aplikaciju E-Indeks te opisati i prikazati sve tehnologije potrebne za izradu takve aplikacije. Glavna ideja bila je omogućiti pregled svih informacija o studentu i njegovim uspjesima kroz semestre na brz i jednostavan način. Aplikacija podatke u JSON formatu preuzima s priređenog resursa na mreži preko JMBAG broja te ih prikazuje na zaslonu. Za kreiranje aplikacije koristilo se Android Studio razvojno okruženje te Kotlin programski jezik.

**Ključne riječi:** Android, E-Indeks, JSON, JMBAG, Android Studio, Kotlin

## **ABSTRACT**

The goal of this final paper was to create an Android mobile phone Application E-index and to describe and show all the technologies necessary to make such an application. The main idea was to enable a quick and easy overview of all the information about the student and his achievements throughout the semesters. The application downloads data in JSON format from the prepared resource on the network via the JMBAG number and displays it on the screen. The Android Studio development environment and the Kotlin programming language were used to create this application.

**Keywords:** Android, E-Index, JSON, JMBAG, Android Studio, Kotlin