

# Mobilna aplikacija za pretvorbu mjernih jedinica

---

**Cindrić, Marko**

**Undergraduate thesis / Završni rad**

**2020**

*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:200:969999>

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

*Download date / Datum preuzimanja:* **2025-04-01**

*Repository / Repozitorij:*

[Faculty of Electrical Engineering, Computer Science and Information Technology Osijek](#)



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

**Stručni studij**

**MOBILNA APLIKACIJA ZA PRETVORBU MJERNIH  
JEDINICA**

**Završni rad**

**Marko Cindrić**

**Osijek, 2020.**

## Sadržaj

1. UVOD .....	1
1.1. Zadatak završnog rada .....	1
1.2. Android operacijski sustav .....	1
1.3. Android Studio .....	2
1.4. Programski jezik Java.....	3
1.5. XML .....	4
1.6. Espresso .....	5
2. GLAVNI DIO RADA .....	7
2.1. Izrada grafičkog sučelja.....	7
2.2. Izrada programskog dijela završnog rada.....	8
2.3. Programsko rješenje za klasu <i>main</i> i klasu <i>duljina</i> :.....	9
2.4. Testiranje aplikacije.....	10
3. ZAKLJUČAK .....	12
4. LITERATURA.....	13
5. SAŽETAK.....	14
ABSTRACT .....	15
6. ŽIVOTOPIS .....	16
7. PRILOZI.....	17

# 1. UVOD

Cilj prilikom izrade ovog završnog rada je izraditi aplikaciju koja će biti korisna u stvarnom životu. Mobilna aplikacija za pretvorbu mjernih jedinica je izvrsna ideja zbog toga što korisniku ne treba internet da bi pretvorio neku njemu nepoznatu mjernu jedinicu u mjernu jedinicu na koju je navikao u svom svakodnevnom životu. Aplikacija sadrži sve glavne mjerne veličine sa svim svojim mjernim jedinicama koje se najčešće koriste, jednako kao i neke starije mjerne jedinice koje se danas više i ne koriste, no ipak trebaju biti sastavni dio aplikacije jer se ne smiju zapostaviti niti korisnici starije životne dobi. S obzirom da je aplikacija razvijena za Android operacijski sustav u Android Studio-u, ovdje će se također pisati o Android-u, Android Studio-u kao i o tehnologijama koje su korištene za razvoj ove aplikacije. Tehnologije koje su se koristile za razvoj ove aplikacije su Java i XML. Nakon što je aplikacija dovršena, ona je i testirana da bi se mogle ispraviti sve moguće greške u programu, da bi se mogao osigurati fluidan i siguran rad aplikacije. Tehnologija koja je korištena za testiranje ove aplikacije je Espresso.

## 1.1. Zadatak završnog rada

Zadatak završnog rada je izraditi mobilnu aplikaciju za pretvorbu mjernih jedinica. Aplikacija treba pokriti što više područja da bi bila korisna u stvarnom svijetu.

## 1.2. Android operacijski sustav

Android je operacijski sustav baziran na Linux kernel-u. Razvijen je od strane Android inc. tvrtke koju je Google prvotno potpomagao, a zatim i kupio. To je operacijski sustav koji se koristi prvenstveno na uređajima sa ekranima osjetljivim na dodir npr. pametni mobiteli, tableti, pametni satovi. No može se naći i na uređajima koje nemaju ekrane osjetljive na dodir poput televizija. Android operacijski sustav je predstavljen 2007. godine, a lansiran je u srpnju 2008. godine. Prvi mobilni uređaj koji je koristio Android operacijski sustav je T-Mobile G1. Android je najkorišteniji operacijski sustav na svijetu, sa udjelom od 86.6% tržišnog udjela. Razlog tome je činjenica da je Google dao izvorni kod razvojnim inženjerima besplatno pod *open source* licencom, što je omogućilo tvrtkama ali i entuzijastima da ga besplatno koriste, i dalje razvijaju na način kako njima odgovara, s tim da ipak postoji jedan uvjet koji je Google postavio, a to je da proizvođači uređaja moraju preinstalirati određene Google-ove aplikacije, postaviti Google kao zadani internet pretraživač i osigurati da su traka za pretraživanje i Trgovina *Play* ikone postavljene na početni

zaslon. To je omogućilo i niže cijene samih pametnih telefona, koje si je onda veći dio svjetske populacije može priuštiti. Za razliku od tvrtke Apple koja svoj operativni sustav iOS koristi samo za svoje mobilne uređaje poput iPhone-a, iPod-a i iPad-a. Apple također drži visoku cijenu svojih uređaja koju si visok postotak svjetske populacije ne može priuštiti i zbog toga svega pametni telefoni pogonjeni Android operativnim sustavom vladaju svjetskim tržištem. Ipak treba naglasiti da unatoč tome što je Android besplatan, nikako se ne smije reći da nije kvalitetan, već, dapače, izrazito je kvalitetan i pouzdan operativni sustav. Razvojni inženjeri koji razvijaju aplikacije za Android koriste Android SDK (*Software Development Kit*). SDK sadržava *debugger*, biblioteke, dokumentaciju i emulator. Razvojni inženjeri koriste IDE (*Integrated Development Environments*) za razvoj aplikacija, najpoznatiji su Android Studio i Eclipse. A tehnologije koje je moguće koristiti za razvoj aplikacija su Java i Kotlin.

### **1.3. Android Studio**

Android Studio je službeni IDE za razvoj aplikacija. Predstavljen je u svibnju 2013. godine, na Google-ovoj I/O konferenciji. Prva stabilna verzija je izdana u prosincu 2014. godine. Android Studio omogućuje razvojnom inženjeru razvijanje aplikacija bez potrebe za mobilnim uređajima koji koriste Android operativni sustav, jer u sebi sadrži emulator koji simulira rad mobilnih uređaja, to je besplatna i izrazito korisna opcija, jer omogućava entuzijastima koji nemaju novac za nabavu velikog broja stvarnih mobilnih uređaja simulaciju rada istih. Emulator je moguće prilagoditi na više načina, kao što su na primjer promjena rezolucije, veličine ekrana, pa čak i do testiranja aplikacija na svim mogućim do sada izdanim verzijama android-a. Time je uvelike smanjena krivulja učenja i ispravljanje svih mogućih pogrešaka u radu programa. Ipak, prilikom izrade praktičnog djela završnog rada, emulator nije korišten zbog toga što je to ipak sporiji način instaliranja i pokretanja aplikacije, jer samo pokretanje emulatora traje i do 10-ak minuta, i to svaki put kada se pokreće Android studio prvi put nakon pokretanja računala. Praksa je pokazala da je brži način instalacije i pokretanja aplikacije povezivanjem mobitela i računala putem USB-a. Zbog tih razloga je za provjeru aplikacije korišten isključivo fizički mobitel. Dio studio-a koji se naziva uređivač koda, služi za programiranje, programiran je na način da asistira programeru na način da mu nudi dovršetak određenih dijelova kod-a kao što su zatvaranje zagrada, predlaganje naziva klase, funkcija i varijabli koje programer koristi u svom radu nakon što programer upiše sam početak naziva klase, funkcije ili varijable. Sve aplikacije nakon dovršetka izrade i kompilacije su u APK formatu (*Android Package Format*) i kao takve se mogu postaviti na Google-ovu Trgovinu *Play*, što je Google-ova trgovina aplikacija koja u trenutku pisanja ovoga završnog rada broji više

od 2.9 milijuna mobilnih aplikacija. Android Studio je dostupan za sve tri platforme: Mac, Windows i Linux, i u njemu je integriran Github koji je dao razvojnim inženjerima kolaboracijske mogućnosti.

## 1.4. Programski jezik Java

Java je programski jezik čiji je razvitak započeo 1991. godine. Za njen nastanak je odgovorna tvrtka Sun Microsystems. Njezina prva beta verzija je izdana 1995. godine, dok je Java 1.0 verzija izdana u siječnju 1996. godine. To je programski jezik koji je prvenstveno razvijan za interaktivnu televiziju, no ta tehnologija je bila previše napredna za tadašnju digitalnu kablsku televizijsku tehnologiju. Vođa tima koji je razvijao Java-u je James Gosling kojega nazivamo „ocem Java-e“. James Gosling je mislio da ukoliko tisuću ljudi preuzme i pokuša raditi u Java-i, da to može smatrati uspjehom. Principi po kojima je Java razvijana su: jednostavnost, prenosivost, robusnost, sigurnost, objektna orijentiranost, interpretiranost, dinamičnost, platformska neovisnost, višenitnost, arhitekturna neutralnost i visoke performanse. Ti principi su doveli do toga da je magazin Time proglasio Java programski jezik za jedan od 10 najboljih proizvoda 1995. godine. Java je trenutno jedan od najkorištenijih programskih jezika na svijetu sa udjelom od 18.84%. Što je daleko više od Gosling-ovih očekivanja. Koristi se kod razvoja aplikacija na više platformi među kojima su Windows aplikacije, Web aplikacije, video igre, mobilne aplikacije, poslovne aplikacije i razne druge. Neke od najpoznatijih aplikacija napisanih u Java programskom jeziku su: Twitter, Pinterest, Uber, Kickstarter, Eclipse, Netbeans IDE i drugi. Glavna prednost Java-e leži u činjenici da je napravljena na način da omogućuje razvojnim inženjerima rad svojih programa na raznim platformama bez potrebe za prilagođavanjem programskog koda. Tako je nastao slogan „WORA“ (*Write once, run anywhere*). To je omogućilo na primjer da aplikacija napisana i kompilirana na Macintosh-u, radi na Windows-ima identično kako bi radila i na Macintosh sustavu. Način na koji je to izvedeno je taj da se programski kod koji napiše programer prvo prevodi u bytecode, koji izvršava JVM (*Java Virtual Machine*). JVM je platformski ovisna, i ona je ta koja je napravljena posebno za svaku platformu, time se dobila „WORA“. Na slici je prikazan primjer programskog kod-a napisanog u Java-i, korištenog kod izrade praktičnog djela završnog rada. U odnosu na C++ kod napisan u Java-i se izvršava sporije, no bolja je zbog toga što je prenosiva i ima bolje sigurnosne značajke. Sigurnost se očituje kod toga što virtualni stroj pruža „tampon zonu“ između računala i programa koji se izvodi u virtualnom stroju. Prednost Java-e u odnosu na C++ je i to što je Java programski kod moguće modificirati i za vrijeme izvođenja programa, dok C++ nema tu mogućnost.

```

public class About extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_about);
    }
}

```

Slika 1.1. Primjer Java programskog jezika

## 1.5. XML

Povijest XML-a (*Extensible Markup Language*) je započela ranih 70.-ih godina 20.-og stoljeća sa razvojem SGML-a (*Standardised Generalised Markup Language*) sa Charles-om Goldfarb-om, Ed-om Mosher-om i Ray-em Lorie-em. XML nije pravi programski jezik niti se može kompilirati. On služi za pohranjivanje, prijenos i opisivanje podataka. Jako je sličan HTML-u (*Hypertext Markup Language*) no razlika između njih je zapravo dosta velika. XML služi za prijenos podataka dok HTML služi za prikaz podataka. XML-ovi tag-ovi nisu predefinirani kao što je to slučaj sa HTML-om. Potonji razlog je možda i najbolji opis razlike ova dva jezika, jer XML ima pravilo da tag-ove koje razvojni inženjer definira budu razumljivi i onome koji će te tagove koristiti u daljnjem razvoju aplikacije. Sam XML je platformski neovisan što omogućuje da se podatci napisani u XML formatu bez ikakvih problema šalju na druge platforme bez ikakvih problema i potreba za preinakama. Mogućnosti koje XML nudi su bezbrojne, na primjer ukoliko se razvojni inženjeri dogovore za jedan određeni standard moguće je opisati određeni proizvod na način da računala mogu prikupljati podatke koji su napisani u XML-u i onda te podatke usporediti sa svojim trenutnim podacima. No ključ uspjeha XML-a leži u njegovoj jednostavnosti, jer omogućava organizaciju velikih grupa podataka. U Android Studio-u se XML koristi za dizajniranje grafičkog izgleda aplikacije, poput toga gdje se nalaze gumbi, prostori za unos teksta, prikaz teksta i slično. Na slici je prikazan primjer kod-a napisanog u XML-u korištenog kod izrade praktičnog djela završnog rada.

```
<Button
    android:id="@+id/pretvoriBtn"
    android:layout_width="match_parent"
    android:layout_height="70dp"
    android:text="Pretvori"
    android:textSize="36sp"
    android:textStyle="bold" />
```

Slika 1.2. Primjer XML-a

## 1.6. Espresso

Google je lansirao Espresso framework 2013. godine. Google je omogućio da Espresso bude *open source*, da bi svatko mogao pridonositi njegovom razvitku, i u konačnici da bi cijela zajednica razvojnih inženjera imala koristi od cijelog projekta. Namjena mu je bila da omogući razvojnim inženjerima koji razvijaju Android aplikacije da na jednostavan način simuliraju sve moguće interakcije korisnika sa programom. U početku je bilo moguće testiranje isključivo jedne aplikacije, dakle one koju razvojni inženjeri trenutno razvijaju. No s vremenom je postalo moguće testirati i dijelove aplikacije koji koriste neke vanjske aplikacije. Ipak, nije moguće pristupiti klasama tih vanjskih aplikacija već je kod takvih slučajeva moguće napraviti samo *black box* testiranje. *Black box* testiranje je način testiranja aplikacija na način da korisnik unese određene podatke, i na izlazu očekuje određeni rezultat, s tim da nema uvid na ono što se događa na tom putu između unosa podataka i rezultata na kraju. Na slici je prikazan primjer kod-a napisanog u Espresso-u korištenog kod izrade praktičnog djela završnog rada.



```

@Test
public void testUserChoosingDistance()
{
    Espresso.onView(withId(R.id.duljinaBtn)).perform(click());
    Espresso.onView(withId(R.id.Spinner1)).perform(click());
    Espresso.onData(allOf(is(instanceOf(String.class))))
        .atPosition(3).perform(click());
    Espresso.onView(withId(R.id.firstNumEditText))
        .perform(typeText(testNumber));
    Espresso.closeSoftKeyboard();
    Espresso.onView(withId(R.id.Spinner2)).perform(click());
    Espresso.onData(allOf(is(instanceOf(String.class))))
        .atPosition(4).perform(click());
    Espresso.onView(withId(R.id.pretvoriBtn)).perform(click());
    Espresso.onView(withId(R.id.resultTextView))
        .check(matches(withText("0.05")));
    Espresso.pressBack();

    Espresso.onView(withId(R.id.duljinaBtn)).perform(click());
    Espresso.onView(withId(R.id.Spinner1)).perform(click());
    Espresso.onData(allOf(is(instanceOf(String.class))))
        .atPosition(4).perform(click());
    Espresso.onView(withId(R.id.firstNumEditText))
        .perform(typeText(testNumber));
    Espresso.closeSoftKeyboard();
    Espresso.onView(withId(R.id.Spinner2)).perform(click());
    Espresso.onData(allOf(is(instanceOf(String.class))))
        .atPosition(3).perform(click());
    Espresso.onView(withId(R.id.pretvoriBtn)).perform(click());
    Espresso.onView(withId(R.id.resultTextView))
        .check(matches(withText("500.0")));
    Espresso.pressBack();
}

```

Slika 1.4. Primjer Espresso-a

## 2. GLAVNI DIO RADA

### 2.1. Izrada grafičkog sučelja

Android studio omogućuje izradu grafičkog sučelja na dva načina. Moguće je pisati programski grafičko sučelje, a moguće je i slagati komponente, za koje studio automatski generira programski kod, drugi način je jednostavniji i brži, i on je korišten za izradu ovog grafičkog sučelja. Grafičko sučelje je izrađivano metodom pokušaja i pogrešaka, jer je autor smatrao da je to najbolji način zbog toga što se odmah prilikom crtanja grafičkog sučelja mogla provjeriti njegova funkcionalnost i praktičnost. Prvo je izrađen glavni prozor programa koji se naziva glavna aktivnost (*main activity*). Glavna aktivnost sadrži gumbе, koji prilikom pritiska otvaraju sljedeću aktivnost koji je željena mjerna jedinica koju je korisnik odabrao, i koja mu je potrebna u tom trenutku.



Slika 2.1. Grafički prikaz glavne aktivnosti

Pritiskom na gumb Duljina, korisniku će se otvoriti novi prozor, tj. nova aktivnost u kojoj može pretvarati mjerne jedinice za mjernu veličinu duljina. Prilikom dizajna grafičkog sučelja korištene su grafike jednostavnih boja, zbog toga što je cilj bio maksimalno povećati praktičnost aplikacije. Skroz na vrhu se nalazi zeleni pravokutnik sa napisanim nazivom aplikacije, ispod nje se nalazi

bijeli padajući izbornik, ispod bijelog padajućeg izbornika se nalazi svijetlo sivo polje u koje korisnik unosi vrijednosti. Ispod polja za unos se nalazi tamno sivi gumb koji se koristi za pretvaranje mjerne jedinice. Ispod tog gumba se nalazi drugi bijeli padajući izbornik koji služi za odabiranje mjerne jedinice u koju korisnik želi pretvoriti početnu mjernu jedinicu. I ispod toga se nalazi polje u kojem se ispisuje konačan rezultat. Ove boje i nijanse tih boja su odabrane zbog toga da bi se aplikacija mogla koristiti i u prirodi, odnosno da bi se što bolje vidjela i pri izravnoj sunčevoj svjetlosti.



Slika 2.2. Grafički prikaz aktivnosti Duljina

## 2.2. Izrada programskog dijela završnog rada

Program je zamišljen tako da svaki gumb koji se nalazi na glavnoj aktivnosti predstavlja ulaz na drugu aktivnost u kojem korisnik može pretvarati mjerne jedinice. Pri pisanju programa, prvo su izrađeni gumbi na način da su deklarirani i instancirani. I nakon toga su napisani eksplicitne namjere koje su omogućile komunikaciju između glavne aktivnosti i ostalih aktivnosti.

### 2.3. Programsko rješenje za klasu *main* i klasu duljina:

Nakon pritiska na određeni gumb, na primjer gumb-a duljina program otvara aktivnost duljina. Nakon te interakcije korisnik može koristiti program, sam program je zamišljen da bude što jednostavniji, da bi ga korisnik mogao intuitivno koristiti. Grafički izgled aktivnosti duljina započinje sa padajućim izbornikom u kojem se nalazi popis svih mjernih jedinica, tu korisnik odabire mjernu jedinicu koju trenutno koristi, a želi je promijeniti u neku drugu. Nakon što korisnik odabere mjernu jedinicu, u dnu aplikacije mu se ispisuje poruka u kojoj piše koju je mjernu jedinicu odabrao. Zatim slijedi *editText* područje u koji korisnik unosi broječanu vrijednost trenutne mjerne jedinice, tj. iznos njemu poznate vrijednosti. Nakon toga slijedi gumb pretvori, koji je zaključan te je s njim nemoguće izvršiti interakciju dok korisnik ne unese neku vrijednost u polje *editText*. Nakon što korisnik unese vrijednost, on „preskače“ gumb pretvori iako mu je on već u ovom trenutku otključan i može ga koristiti. Korisnik treba pritisnuti na drugi padajući izbornik u kojem bira mjernu jedinicu u koju želi promijeniti trenutnu mjernu jedinicu. Nakon što korisnik odabere mjernu jedinicu, na dnu aplikacije korisniku se ispisuje poruka u kojoj piše koju je mjernu jedinicu odabrao. Zatim klikne na gumb pretvori i rezultat mu se prikaže ispod drugog padajućeg izbornika. Iako se čini potpuno nelogičnim postaviti gumb pretvori prije nego što korisnik odabere mjernu jedinicu u koju želi promijeniti trenutnu mjernu jedinicu, to je napravljeno da korisnik ne bi slučajno pritisnuo na polje *editText* jer je program razvijan pod pretpostavkom da bi korisnik ovu aplikaciju mogao koristiti u uvjetima u kojima ne bi imao vremena za biti potpuno precizan, na primjer da ga korisnik koristi na gradilištu. Time je uklonjena mogućnost slučajnog klika na polje *editText* i omogućavanje ponovnog unosa broječane vrijednosti. Svaka aktivnost ima vlastitu klasu koja je aktivna u trenutku kad se koristi. To je izrađeno zbog toga što je preporučeno programerima da svoje programe pišu na način da ih podijele na što više manjih dijelova iz razloga da bi im bilo lakše ispravljati moguće greške i da bi ih se moglo implementirati u neke druge programe ukoliko se pokaže prilika za to. Time je uvelike smanjena krivulja učenja i trajanje razvoja praktičnog djela završnog rada. Matematički dio ovoga programa se odvija unutar dvaju *switch case*-a. Unutar prvog *switch case*-a se odvijaju matematičke operacije u kojima se unesena veličina pretvara u privremeni rezultat. Taj privremeni rezultat je osnovna mjerna veličina međunarodnog sustava mjernih jedinica (SI sustav). Nakon toga se privremeni rezultat šalje u drugi *switch case*, i u njemu se on pretvara u odabranu mjernu jedinicu, ta mjerna jedinica je ujedno i konačan rezultat i on se ispisuje na ekranu.



Slika 2.3. Grafički prikaz rada aktivnosti duljina

## 2.4. Testiranje aplikacije

Nakon izrade aplikacije, istu je potrebno i provjeriti zbog toga što je poželjno da kad se aplikacija plasira na tržište ima što manje pogrešaka, odnosno, u idealnom slučaju da ih uopće nema, jer korisnici ne žele imati greške u radu niti jedne aplikacije. Ukoliko bi došlo do nekih većih pogrešaka u radu aplikacije, korisnici bi vrlo jednostavno deinstalirati tu aplikaciju, i potražili novu u trgovini aplikacija. Za tvrtke to sa poslovne strane nikako nije dobro, jer gube na relevantnosti, i uz financijske gubitke, to bi moglo dovesti do propadanja tvrtke. Zbog navedenih razloga prije nego se aplikacija predstavi i pusti u prodaju, ona se testira, do najsitnijeg detalja. Testiranje se može odraditi na više načina. Jedan od načina je i puštanje beta verzije aplikacije koju bi korisnici mogli preuzeti i testirati sami, i prilikom testiranja javljati tvrtkama sve greške u radu aplikacije koje uspiju zamjetiti. A moguće je testirati aplikaciju programski, u nekom od testnih okvira. U ovom slučaju aplikacija je testirana u Espresso-u.. Da bi korisnik testirao program mora napisati programski kod koji će aplikaciju testirati korak po korak. U ovom slučaju je program testiran na način da program klikne na određenu mjernu veličinu, zatim na padajući izbornik, odabere određenu mjernu jedinicu u padajućem izborniku, zatim se unosi testni broj i zatvara se tipkovnica,

otvara se drugi padajući izbornik, bira se druga mjerna jedinica, klikne se na gumb pretvori, dobiveni rezultat se uspoređuje sa očekivanim rezultatom i na kraju se program vraća u početni izbornik i po tome se izlazi iz aplikacije. Ukoliko program nema pogrešaka, programer će dobiti obavijest kako je testiranje uspješno završeno, no ukoliko ipak ima pogrešaka, prilikom testiranja programa će doći do prisilnog prekida izvođenja programa, i korisnik automatski dobija poruku u kojoj je opisano gdje je došlo do pogreške i zbog čega se pogreška pojavila. Tako programer ne mora prolaziti kroz cijeli program, već može otići samo na taj dio gdje se pogreška dogodila, i puno brže ispraviti grešku. Samo testiranje nakon što se testovi napišu se izvodi izrazito brzo, i time uvelike smanjuju vrijeme testiranja, razvojni inženjer koji je zadužen za testiranje aplikacije ne može tom brzinom sam birati koje će se akcije izvoditi, koje će se vrijednosti unositi, i u isto vrijeme provjeravati očekivani rezultat sa dobivenim rezultatom.

### **3. ZAKLJUČAK**

Cilj kod izrade ovog završnog rada je bio izraditi mobilnu aplikaciju koja će biti korisna u svakodnevnom životu. Prilikom izrade završnog rada primjenjivano je sve što studenti uče na fakultetu, od programiranja, do testiranja aplikacija. Na toj „avanturi“ izrade završnog rada student uvelike proširuje svoja znanja iz programiranja, dizajna, i testiranja aplikacija, što će mu koristiti kod zaposlenja. Završni rad je izrađen u programskom jeziku Java jer je Java programski jezik uz Kotlin najkorišteniji programski jezik za izradu Android aplikacija. Da bi se aplikacija mogla koristiti bez problema u njezinom radu, bilo ju je potrebno dizajnirati na način da bude praktična i intuitivna kod korištenja. Da bi aplikacija bila sigurna za korištenje, bilo ju je potrebno testirati, ona je testirana u Espresso okviru. On je odabran zbog toga što brz, jednostavan za učenje i temeljit okvir za testiranje mobilnih aplikacija koje se rade za Android operativni sustav.

## 4. LITERATURA

<https://www.elprocus.com/what-is-android-introduction-features-applications/>

<https://www.idc.com/promo/smartphone-market-share/os>

<https://www.cnet.com/news/yes-android-is-a-free-and-open-platform-but/>

<https://www.youtube.com/watch?v=KitoxUB11go>

<https://www.techopedia.com/definition/25106/android-operating-system>

<https://www.statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store/>

<https://searchmobilecomputing.techtarget.com/definition/Android-Studio>

<https://pypl.github.io/PYPL.html>

<https://www.javatpoint.com/history-of-java>

<https://www.geeksforgeeks.org/the-complete-history-of-java-programming-language/>

<https://howtodoinjava.com/java/basics/what-is-java-programming-language/>

<https://whatis.techtarget.com/definition/XML-Extensible-Markup-Language>

[https://www.w3schools.com/xml/xml\\_what.asp](https://www.w3schools.com/xml/xml_what.asp)

<https://ccollins.wordpress.com/2008/03/03/a-brief-history-of-xml/>

<https://www.youtube.com/watch?v=KeLiQXqVgMI>

<https://www.youtube.com/watch?v=tZE-O5KTwCY>

[https://www.youtube.com/watch?v=YXkuxRL7-\\_M](https://www.youtube.com/watch?v=YXkuxRL7-_M)

<https://www.androidauthority.com/introduction-to-xml-968598/>

<https://www.sciencedirect.com/topics/computer-science/java-programming-language>

<https://www.youtube.com/watch?v=8BsaOqBD-Uo>

<https://www.youtube.com/watch?v=dFIPARW5IX8>

<https://www.youtube.com/watch?v=6ow3L39Wxmg>

[https://www.youtube.com/watch?v=on\\_OrrX7Nw4](https://www.youtube.com/watch?v=on_OrrX7Nw4)

<https://developer.android.com/guide/topics/ui/controls/spinner>

<https://stackoverflow.com/questions/44109437/setting-up-multiple-spinners-in-android>



## 5. SAŽETAK

U ovom završnom radu cilj je bio izraditi mobilnu aplikaciju koja će imati svoju primjenu u stvarnom svijetu. Razvijena je za Android operacijski sustav, za korištenje u uvjetima u kojima korisnik ne bi imao pristup internetu. U prvom dijelu rada, je opisan Android operacijski sustav kao i tehnologije koje su korištene za izradu praktičnog djela završnog rada. A u drugom djelu je opisan način na koji je završni rad izrađen.

**Ključne riječi:** Android operacijski sustav, mobilna aplikacija, Java

## **ABSTRACT**

In this final paper the goal was to develop mobile application that will have its real world application. It was developed for the Android operating system, for using in conditions in which the user wouldn't be able to use the internet. In the first part of this paper, Android operating system was described as well as technologies that were used to create practical part of this paper. Second part of this paper describes how the practical part was created.

**Key words:** Android operating system, mobile application, Java

## 6. ŽIVOTOPIS

Marko Cindrić rođen je 11.06.1997. u Novoj Gradiški. Osnovnu školu je pohađao u Osnovnoj školi Mato Lovrak Nova Gradiška. Završio je Elektrotehničku i ekonomsku školu Nova Gradiška u kojoj stječe strukovnu kvalifikaciju Tehničar za računalstvo. Po završetku srednje škole upisuje preddiplomski stručni studij elektrotehnike, smjer informatika na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija Osijek. Odlično govori engleski jezik. Poznaje rad u Microsoft Office alatima, te se služi programskim jezicima kao što su C, C++ i Java.

## **7. PRILOZI**

Prilozi su priloženi na DVD-u.