

ŠOLSKI CENTER CELJE  
Srednja šola za kemijo, elektrotehniko in računalništvo

# **IZDELAVA ŠPORTNE APLIKACIJE**

Raziskovalna naloga

PODROČJE: Računalništvo

Avtorici:

Nuša Alja KRALJ, R-4. a

Maja ŠKORJANC, R-4. a

Mentor:

Boštjan LUBEJ, dipl. inž.

Mestna občina Celje, Mladi za Celje

Celje, april 2021

## IZJAVA

Mentor Boštjan Lubej, v skladu z 20. členom Pravilnika o organizaciji mladinske raziskovalne dejavnosti »Mladi za Celje« Mestne občine Celje, zagotavljam, da je v raziskovalni nalogi z naslovom Izdelava športne aplikacije, katere avtorici sta Nuša Alja Kralj in Maja Škorjanc:

- besedilo v tiskani in elektronski obliki istovetno,
- pri raziskovanju uporabljeno gradivo navedeno v seznamu uporabljene literature,
- da je za objavo fotografij v nalogi pridobljeno avtorjevo dovoljenje in je hranjeno v šolskem arhivu,
- da sme Osrednja knjižnica Celje objaviti raziskovalno nalogo v polnem besedilu na knjižničnih portalih z navedbo, da je raziskovalna naloga nastala v okviru projekta Mladi za Celje,
- da je raziskovalno nalogo dovoljeno uporabiti za izobraževalne in raziskovalne namene s povzemanjem misli, idej, konceptov oziroma besedil iz naloge ob upoštevanju avtorstva in korektnem citiranju,
- da smo seznanjeni z razpisni pogoji projekta Mladi za Celje.

Celje, 13.5.2021



Podpis mentorja

Podpis odgovorne osebe

## ZAHVALA

Zahvaljujemo se vsem, ki so kakorkoli pomagali pri izdelavi te raziskovalne naloge. Brez pomoči drugih naloga ne bi nastala, pa naj je šlo le za spodbudne besede, majhno idejo ali nasvete in kritike pri izdelovanju izdelka.

Posebna zahvala gre vsem anketirancem, ki so si vzeli čas, rešili anketo in nam s tem pomagali pri empiričnem delu raziskovalne naloge.

Največja zahvala gre mentorju, Boštjanu Lubeju, za ves trud, čas, podporo ter vztrajnost, ki jo je vložil v izdelovanje raziskovalne naloge, ter nudenje pomoči in napotkov skozi celotno nalogo. Zahvalili bi se mu tudi za posredovanje ankete med dijake.

Zahvalili bi se mag. Marijani Marinšek za pregled povzetka v angleščini in mag. Andreji Tkalec za jezikovni pregled naloge.

## POVZETEK

**Ključne besede:** šport, športna aplikacija, Android Studio, tekmovalnost

Cilj raziskovalne naloge je bil ugotoviti, kakšen vpliv ima uporaba športnih aplikacij na posameznika, ter raziskati tekmovalnost in primerjanje dosežkov, ki se z njimi pojavljajo. Za odpravljanje nezdrave tekmovalnosti in primerjanja, ki imajo na posameznika negativen vpliv, smo izdelali športno aplikacijo Active. Z njo ima posameznik dostop le do svojih športnih dosežkov in spremlja le svoj napredek.

V okviru raziskovalne naloge smo na podlagi empiričnih podatkov, pridobljenih s pomočjo ankete, ugotovili, da večina anketirancev pri športnih aplikacijah ne zaznava nezdrave tekmovalnosti. Kljub temu bi nekoliko manj kot dve tretjini anketirancev verjetno raje uporabljalo aplikacijo, s katero bi spremljali le svoje športne aktivnosti.

V zaključku raziskovalne naloge so navedene končne ugotovitve rezultatov raziskovalne naloge in vizija aplikacije v prihodnosti.

## **ABSTRACT**

**Keywords:** sports, sports application, Android Studio, competition

The aim of this research paper has been to establish what effects the use of sports applications has on individuals, as well as to research competition and comparison of achievements that occur with such use. To eliminate unhealthy competition and comparison of individual achievement, both of which have a negative effect on one's performance, we have created a sports application, called Active. With this application one can access only their own sporting achievements and track only their own progress.

Based on the empirical data acquired through a survey, we have found out that most of the respondents are not aware of unhealthy competition. However, slightly less than two thirds of respondents would still rather use an app enabling them to track their sporting activities.

We have listed final conclusions and our vision of the app in the conclusion of the research paper.

## KAZALO VSEBINE

1	UVOD.....	1
1.1	HIPOTEZE.....	2
1.2	RAZISKOVALNE METODE .....	2
2	UPORABLJENI PROGRAMI .....	3
2.1	ANDROID STUDIO.....	3
2.2	GOOGLE CLOUD PLATFORM .....	4
2.3	HEIDISQL .....	5
2.4	XAMPP .....	6
3	APLIKACIJA ACTIVE .....	8
3.1	NAMEN APLIKACIJE.....	8
3.2	DELOVANJE APLIKACIJE.....	8
3.3	TEŽAVE PRI USTVARJANJU APLIKACIJE.....	25
4	MOŽNOSTI NADGRADNJE V PRIHODNOSTI .....	26
5	PREDSTAVITEV REZULTATOV RAZISKOVALNE NALOGE.....	27
6	ZAKLJUČEK .....	29
7	VIRI IN LITERATURA.....	30
8	PRILOGE .....	32
8.1	PREDSTAVITEV OSEB .....	32
8.2	ANKETNI VPRAŠALNIK.....	32

## KAZALO SLIK

Slika 1: Logotip platforme Android Studio [11].....	3
Slika 2: Logotip programskega jezika Kotlin [12]. ....	3
Slika 3: Logotip programskega jezika Java [13].....	4
Slika 4: Logotip platforme Google Cloud Platform s funkcijami, ki jih ponuja [7].....	5
Slika 5: Shema podatkov v SQL bazi [9].....	5
Slika 6: Logotip programa HeidiSQL [8]. ....	6
Slika 7: Logotip razvojnega okolja XAMPP [10].....	7
Slika 8: Obrazec za prijavo. ....	9
Slika 9: Obrazec za registracijo. ....	9
Slika 10: Tabela za podatke o uporabniku. ....	10
Slika 11: Tabela za podatke o aktivnostih uporabnika. ....	10
Slika 12: Uporabniški vmesnik razvojnega okolja XAMPP.....	10
Slika 13: Koda za registracijo uporabnika. ....	11
Slika 14: Koda za prijavo uporabnika.....	11
Slika 15: Koda gumba za registracijo. ....	12
Slika 16: Koda za PutData metodo pri registraciji.....	12
Slika 17: Koda gumba za prijavo.....	13
Slika 18: Začetna stran aplikacije. ....	14
Slika 19: Koda za iskanje trenutne uporabnikove lokacije.....	14
Slika 20: Omogočanje storitve Maps SDK for Android v knjižnici GCP. ....	15
Slika 21: Pridobljen API ključ. ....	15
Slika 22: Implementacija API ključa v kodo. ....	15
Slika 23: Zahteva za omogočanje lokacijske storitve.....	16
Slika 24: Izbiranje športne dejavnosti.....	17
Slika 25: Koda za premik na novo stran. ....	18
Slika 26: Koda za izpis imena izbrane športne dejavnosti.....	19
Slika 27: Koda za izrisovanje uporabnikove poti. ....	19
Slika 28: Sledenje za športno dejavnost pohodništvo.....	20
Slika 29: Koda za vnašanje aktivnosti v bazo podatkov.....	21
Slika 30: Koda za povezavo med bazo podatkov in saveData metodo.....	21
Slika 31: Koda za povezavo med bazo podatkov in PutData metodo. ....	22
Slika 32: Koda za pridobitev podatkov o aktivnostih iz baze podatkov.....	22

Slika 33: Koda za klicanje metode Adapter.....	23
Slika 34: Koda za deklariranje metode Adapter. ....	23
Slika 35: Koda za metodo Adapter, ki poskrbi za izpis podatkov. ....	23
Slika 36: Shranjeni podatki o športni dejavnosti. ....	24
Slika 37: Graf o mnenju anketirancev glede nezdrave tekmovalnosti.....	28
Slika 38: Graf o mnenju anketirancev glede spremljanja le svojih aktivnosti.....	28



## UPORABLJENE KRATICE

IDE – Integrated Development Environment (integrirano razvojno okolje)

SQL – Structured Query Language

NoSQL – Not only SQL

API – Application Programming Interface (vmesnik za programiranje aplikacij)

GCP – Google Cloud Platform

IaaS – Infrastructure as a service (infrastruktura kot storitev)

PaaS – Platform as a service (platforma kot storitev)

SaaS – Software as a service (programska oprema kot storitev)

IT – informacijska tehnologija

PHP – Hypertext Preprocessor (programski jezik za razvoj spletnih strani)

HTTP – Hypertext transfer protocol (protokol za spletne strežnike)

https – Hypertext transfer protocol secure

## 1 UVOD

V sodobnem svetu vsakodnevno uporabljamo številne aplikacije na vseh področjih našega življenja. Tako so postale zelo popularne tudi številne aplikacije za šport. Športne aplikacije spodbujajo gibanje in zdrav življenjski slog, saj so lahko dostopne in ponujajo ogromno povratnih informacij o posamezni fizični dejavnosti.

Videti je, da se tako človek kot družba v celoti danes vse bolj zavedata, da redna telesna dejavnost vpliva ne le na naše fizično, ampak tudi duševno zdravje in je bistvenega pomena za kakovostno življenje. Zato bi morala biti del vsakdana vsakega izmed nas v vseh starostnih obdobjih. Pri otrocih in mladostnikih pozitivno vpliva na njihov celostni razvoj (telesni, čustveni, socialni in razvoj kognitivnih sposobnosti), preventivno pa deluje tudi pred pojavom različnih odvisnosti (od tobaka, alkohola, prepovedanih drog). [17]

Pri uporabi športnih aplikacij, ki omogočajo shranjevanje športnih dejavnosti, pa se pojavljajo tudi negativni vplivi, ki so posledica primerjanja športnega napredka z drugimi. Večina ljudi se niti ne zaveda, kakšen vpliv ima to primerjanje na posameznika. Znano je, da socialno okolje in interakcija z ljudmi močno vpliva na našo motivacijo, prav tako tudi deljenje informacij na družbenih omrežjih. Motivacija je pomembna za napredek in vztrajanje na vseh področjih našega življenja, med drugim tudi v športu. Pri športnih aplikacijah se lahko pojavlja težava, da posameznik s spremljanjem športnih dejavnosti drugih razvija predvsem zunanjo motivacijo, ki se imenuje ekstrinzična. Notranja ali intrinzična motivacija, za katero je značilno, da se z dejavnostjo ukvarjamo zaradi lastnega veselja in želje po izboljšanju svojih sposobnosti na nekem področju, zato pride manj do izraza. [15]

Odločili smo se poglobiti v problematiko tekmovalnosti, ki je prisotna pri uporabi športnih aplikacij, zato smo ustvariti aplikacijo Active. Z njo posameznik spremlja le svoj športni napredek in shranjuje le svoje športne aktivnosti brez primerjanja z dosežki drugih. V nadaljevanju bomo opisali program, ki smo ga uporabili za ustvarjanje aplikacije. To je program Android Studio, v katerem se uporablja programski jezik Java ali Kotlin. Izbrali smo programski jezik Java, v Android Studio pa smo vključili tudi razvojni paket XAMPP, ki vključuje Apache spletni strežnik in SQL podatkovno bazo za prijavo in shranjevanje podatkov uporabnikov. Nato bomo predstavili ustvarjeno aplikacijo Active, njen namen in delovanje, opisali bomo funkcije, ki jih ima, ter možnosti nadgradnje v prihodnosti. Na koncu pa bomo predstavili rezultate raziskovalne naloge.

## **1.1 HIPOTEZE**

Pred začetkom raziskovalne naloge smo si postavili naslednje hipoteze:

- Android Studio se bo pokazal kot dobra izbira platforme za ustvarjanje aplikacije.
- Programski del nam bo predstavljal večjo težavo kot grafična podoba aplikacije.
- Pri uporabi športnih aplikacij se pojavlja nezdrava tekmovalnost, ki se je večina mladostnikov ne zaveda.

## **1.2 RAZISKOVALNE METODE**

Pri raziskovanju smo za pridobivanje informacij uporabili metodo analize različnih virov, in sicer smo v raziskovalno nalogo vključili podatke, ki so na voljo na spletu glede gibanja in tekmovalnosti pri športu.

Z metodo anketiranja smo ugotavljali, katere prednosti in slabosti ljudje vidijo v ostalih, že obstoječih športnih aplikacijah, ter pridobili informacije, kako bi jih naša aplikacija izboljšala. Za anketo smo se odločili, ker smo želeli pridobiti mnenje ljudi o predvsem nezdravi tekmovalnosti ter o primerjanju športnih dosežkov z drugimi, ki se pojavlja pri športnih aplikacijah.

V anketi je sodelovalo 110 anketirancev, starih od 15 do 35 let. Največji delež anketirancev je pripadalo starostni skupini od 16 do 25 let. Razmerje med moškimi in ženskami, sodelujočimi v anketi, ni bilo enako, saj je bilo 13 % predstavnic ženskega in 87 % predstavnikov moškega spola.

## 2 UPORABLJENI PROGRAMI

### 2.1 ANDROID STUDIO

Android Studio predstavlja IDE, ki razvijalcem aplikacij ponuja različna orodja za lažje in hitrejšo razvijanje aplikacij. Slika 1 ponazarja logotip Android Studio platforme.



*Slika 1: Logotip platforme Android Studio [11].*

To okolje med pisanjem kode preverja napake ter s tem omogoča hitrejšo iskanje in odpravljanje napak. Poleg tega razvijalcem med pisanjem kode daje predloge, s katerimi je proces razvijanja aplikacij hitrejši. Brez tega bi bil razvojni postopek veliko daljši in lažje bi bilo narediti vse mogoče napake. Android Studio zgradi in zažene ustvarjeno aplikacijo na dejanski Android napravi ali na tako imenovanem emulatorju, orodju, ki računalniku omogoča simulacijo Android naprave. [3]

Android Studio ima vgrajeno podporo za programski jezik Java in Kotlin. Tako Java kot Kotlin imata svoje prednosti in slabosti. Za razvoj naše aplikacije smo uporabili programski jezik Java, saj smo se z njim že srečali pri strokovnem predmetu. Najpomembnejši razlog za uvedbo programskega jezika Kotlin v Android Studio je bil zmanjšati število vrstic kode. Logotip programskega jezika Kotlin ponazarja slika 2.



*Slika 2: Logotip programskega jezika Kotlin [12].*

Java je objektno usmerjen programski jezik z ogromno odprtokodnimi orodji in knjižnicami za pomoč razvijalcem, Kotlin pa je statični programski jezik z nekaterimi sodobnimi funkcijami. Kotlin je dobra izbira za razvoj aplikacij na strežniški strani, saj uporabnikom omogoča pisanje aplikacij z manj kode, kar izboljšuje njeno vzdrževanje in berljivost. Kljub vsem razlikam med jezikoma sta Java in Kotlin interoperabilna. Katere koli metode, ki se uporabljajo v Javi, se lahko uporabljajo tudi v Kotlinu in obratno. Logotip programskega jezika Java ponazarja slika 3. [16]



*Slika 3: Logotip programskega jezika Java [13].*

## **2.2 GOOGLE CLOUD PLATFORM**

Google Cloud Platform je platforma v oblaku, ki omogoča, da lahko uporabniki preko spletnega vmesnika dostopajo do orodij oziroma storitev, ki jih potrebujejo in uporabljajo. Na sliki 4 je logotip platforme Google Cloud Platform s številnimi funkcijami, ki jih ponuja. Uporabniki lahko te različne vire koristijo za razvoj zelene infrastrukture. Izraz oblak pomeni zagotavljanje IT storitev. Google Cloud Platform omogoča različne vrste modelov storitev v oblaku, in sicer infrastrukturo (IaaS), platformo (PaaS) in programsko opremo (SaaS). Čeprav temeljijo na isti osnovni arhitekturi, so ti modeli glede na področje uporabe različno zasnovani.

IaaS je model, ki omogoča uporabniku nadzor nad operacijskim sistemom, pomnilnikom in razvitimi aplikacijami, ki se izvajajo na navideznih virih, dodeljenih uporabniku, nima pa nadzora nad osnovno infrastrukturo v oblaku.

Paas temelji na modelu IaaS. Uporabnik ima nadzor nad osnovno infrastrukturo, na primer operacijskim sistemom in vmesno programsko opremo. Vmesna programska oprema vključuje vsa potrebna okolja s programi, orodji, zbirkami podatkov, spletnimi storitvami, torej komponentami, ki so zasnovane za izdelovanje aplikacij. Te uporabnikom omogočajo, da z njimi sami izdelajo svojo aplikacijo.

Saas pa omogoča uporabnikom zaganjanje aplikacij lokalno, v računalniku. Zaradi tega uporabnikom tudi ni treba skrbeti za vzdrževanje ter nadgradnjo strojne in programske opreme. Vse je na voljo prek spleta. Uporabniki se le preprosto prijavijo za dostop do virov.

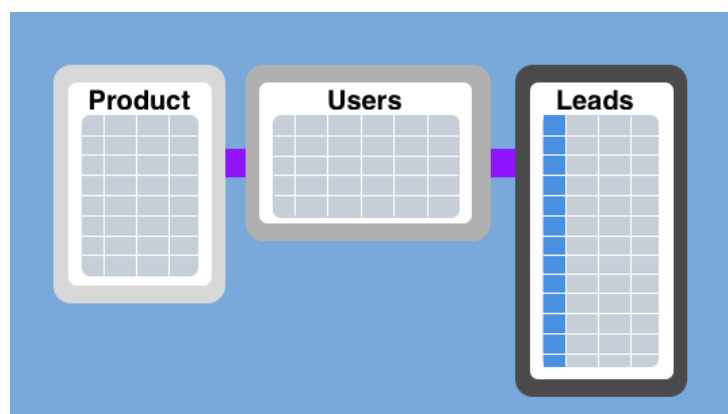
Poleg tega Google Cloud Platform ponuja še številne druge storitve, kot so na primer skladiščenje in zbiranje podatkov, uporaba umetne inteligence in strojnega učenja, orodja za upravljanje, identiteto ter varnost. [1]



Slika 4: Logotip platforme Google Cloud Platform s funkcijami, ki jih ponuja [7].

### 2.3 HEIDISQL

Podatkovne baze so zbirke podatkov, ki so med sabo logično povezani. Poznamo relacijsko podatkovno bazo, ki podatke shranjuje v tabelah, te pa so med sabo povezane in tvorijo vertikalno shemo podatkov, kot je prikazano na sliki 5.



Slika 5: Shema podatkov v SQL bazi [9].

Baze takšnega tipa uporabljajo za kreiranje in urejanje programski jezik, imenovan SQL. Imamo pa tudi NoSQL, ki je nova možnost za shranjevanje podatkov. Ne potrebuje tabelarične sheme in se širi horizontalno. Zaradi dobrega predznanja smo uporabili SQL. [5]

HeidiSQL je brezplačna programska oprema za kreiranje in vzdrževanje podatkovnih baz. Logotip programa lahko vidite na sliki 6. Program poskrbi za povezavo s strežnikom, v našem primeru je to Apache strežnik, pa tudi za preprosto brisanje in dodajanje uporabnikov. [6]

Podpira podatkovne baze MariaDB, MySQL, Microsoft SQL, PostgreSQL in SQLite. V naši raziskovalni nalogi smo se odločili uporabiti MySQL, saj ga podpira XAMPP in zadošča vsem zahtevam za naš projekt. Konkretno pa smo to orodje uporabili za registracijo in prijavo uporabnikov ter za shranjevanje njihovih športnih aktivnosti.



*Slika 6: Logotip programa HeidiSQL [8].*

## **2.4 XAMPP**

XAMPP je razvojno okolje, ki nam pomaga pri namestitvi spletnega strežnika. Njegov logotip je razviden s slike 7. Sestavljen je iz Apache spletnega strežnika, MariaDB baze podatkov, PHP in Pearl programskega jezika. Na voljo je za Windows, Mac in Linux operacijske sisteme. [2]

Apache je brezplačen HTTP strežnik. Deluje že dvajset let in je eden od najbolj zanesljivih spletnih strežnikov. S pomočjo https protokola na vratih 443 skrbi za varno povezavo med stranko in strežnikom. Tako podatki niso javno dostopni med potovanjem po omrežju. Ponuja tudi veliko dodatnih funkcij, kot je na primer preverjanje gesla. [18]

PHP je strežniški programski jezik, ki se uporablja za razvoj dinamičnih spletnih strani. Z njim lahko šifriramo podatke, pošiljamo piškotke, odpremo in urejamo datoteke na strežniku ter spreminjamo in dodajamo podatke v podatkovni bazi. S pomočjo tega jezika smo v naši nalogi

povezali športno aplikacijo s strežnikom in podatkovno zbirko z vsemi potrebnimi podatki o uporabniku. S tem bomo omogočili prijavo z različnih naprav. [4]

Vsebuje tudi programsko orodje PHPMyAdmin, ki skrbi za administracijo MySQL baz kar preko interneta. Kot je razvidno iz imena, je napisan v zgoraj omenjenem programskem jeziku PHP. Omogoča nam kreiranje baz preko uporabniškega vmesnika, ki nam sam generira potrebno SQL kodo. Ponuja pa nam tudi možnost neposrednega pisanja SQL stavkov. [14]



*Slika 7: Logotip razvojnega okolja XAMPP [10].*



### **3 APLIKACIJA ACTIVE**

#### **3.1 NAMEN APLIKACIJE**

Aplikacija Active je zgrajena tako, da sledi športnim dejavnostim uporabnika. Uporabnik lahko izbira med pohodništvom, tekom in kolesarjenjem. Izbrano športno dejavnost aplikacija analizira in sproti preračunava čas, povprečno hitrost in število opravljenih kilometrov. Poleg tega, glede na dano lokacijo, sproti izrisuje pot na zemljevid. Ker želimo uporabnikom omogočiti večjo zasebnost in preprečiti nezdravo tekmovalnost, lahko shranjujejo in spremljajo le svoje športne dejavnosti oziroma dosežke.

#### **3.2 DELOVANJE APLIKACIJE**

Ob zagonu aplikacije se najprej pojavi obrazec za prijavo (slika 8). Brez prijave vstop ni možen, saj drugače ne bi bilo možno razvrščanje aktivnosti po uporabnikih v bazi podatkov. Če uporabnik ni prijavljen, lahko to stori s klikom na gumb za registracijo. Po vpisu osebnih podatkov lahko uporabnik vstopi v osrednji del aplikacije.

Ob kliku na gumb »Registracija« se nam pojavi okno, ki je prikazano na sliki 9. Tu moramo vnesti e-poštni naslov, poljubno uporabniško ime in geslo. Po uspešni registraciji se lahko prijavimo.

## ACTIVE

Prosimo, da se prijavite ali registrirate za nadaljno uporabo.

uporabniško ime \_\_\_\_\_

geslo \_\_\_\_\_

**PRIJAVA**

[Registracija](#)

*Slika 8: Obrazec za prijavo.*

## Registracija

Prosimo, da vnesete vse potrebne podatke za uspešno registracijo.

uporabniško ime \_\_\_\_\_

e-poštni naslov \_\_\_\_\_

geslo \_\_\_\_\_

**VČLANI SE**

*Slika 9: Obrazec za registracijo.*

Za shranjevanje podatkov o uporabnikih smo ustvarili podatkovno bazo z imenom ActiveBase. V njej smo ustvarili dve tabeli. Prva je namenjena podatkom, ki jih uporabnik vnese ob registraciji (slika 10), druga pa beleženju njegovih aktivnosti (slika 11).

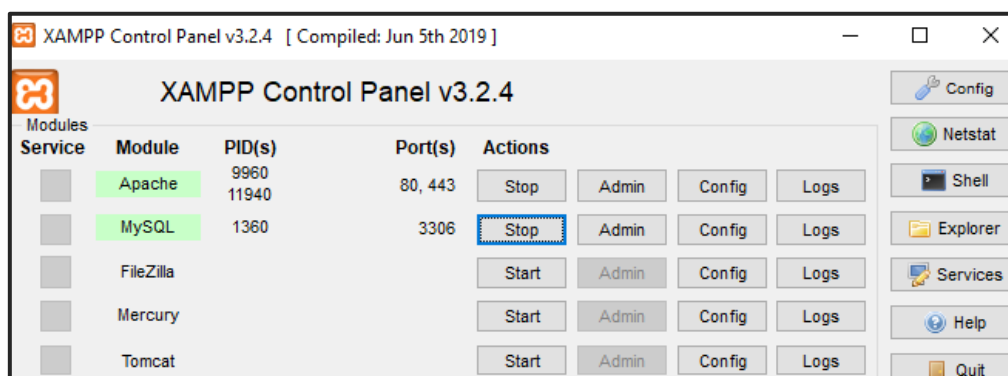
#	Ime	Podatkovni tip	Length/Set	Unsign...	Dopusti ...	Zerofill	Privzeto
1	ID	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREMENT
2	username	VARCHAR	30	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
3	password	VARCHAR	255	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
4	email	VARCHAR	50	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL

Slika 10: Tabela za podatke o uporabniku.

#	Ime	Podatkovni tip	Length/Set	Unsign...	Dopusti ...	Zerofill	Privzeto
1	ID	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREMENT
2	username	VARCHAR	30	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
3	kategorija	VARCHAR	50	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
4	cas	INT	11	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
5	povHitrost	INT	11	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
6	pot	INT	11	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL

Slika 11: Tabela za podatke o aktivnostih uporabnika.

To bazo smo povezali z Apache strežnikom, ki poskrbi za povezavo s SQL strežnikom, ki je na lokalnem omrežju. Tako omogočamo določeno raven varnosti. Ta strežnik preprečuje vpis in dostopanje do podatkov neregistriranim uporabnikom oziroma nedovoljenim zahtevam. Slika 12 ponazarja uporabniški vmesnik.



Slika 12: Uporabniški vmesnik razvojnega okolja XAMPP.

S PHP kodo smo uredili, da se podatki pri registraciji pravilno vnesejo v tabelo (slika 13) in iz nje tudi pridobijo pri prijavi (slika 14). Če se podatki ujemajo, je prijava uspešna.

```
function login($table, $username, $password)
{
    $username = $this->prepareData($username);
    $password = $this->prepareData($password);
    $this->sql = "select * from " . $table . " where username = '" . $username . "'";
    $result = mysqli_query($this->connect, $this->sql);
    $row = mysqli_fetch_assoc($result);
    if (mysqli_num_rows($result) != 0) {
        $dbusername = $row['username'];
        $dbpassword = $row['password'];
        if ($dbusername == $username && /*password_verify($password, $dbpassword) */ $dbpassword == $password) {
            $login = true;
        } else $login = false;
    } else $login = false;

    return $login;
}
```

Slika 13: Koda za registracijo uporabnika.

```
function signUp($table, $email, $username, $password)
{
    $username = $this->prepareData($username);
    $password = $this->prepareData($password);
    $email = $this->prepareData($email);
    //Shash_password = password_hash($password, PASSWORD_DEFAULT);
    $this->sql =
        "INSERT INTO " . $table . " ( username, password, email) VALUES ('" . $username . "','" . $password . "','" . $email . "')";
    if (mysqli_query($this->connect, $this->sql)) {
        return true;
    } else return false;
}
```

Slika 14: Koda za prijavo uporabnika.

S pripravljeno bazo smo vse potrebne podatke vnesli v Android Studio in s tem povezali bazo z našo aplikacijo. Z onClick metodo se ustvarita dve polji podatkov. V prvem se ustvarijo spremenljivke, v drugem pa se vanje vnesejo podatki. Za zagotovitev pravilnosti podatkov smo naredili tudi nekaj if-stavkov, ki preverjajo dolžino uporabniškega imena in gesla, ter ali so potrebni podatki sploh vpisani. Na sliki 15 sta dva primera teh zahtev. Dokler podatki niso pravilni, se ne morejo shraniti v bazo.

Če so bili vsi podatki pravilno vneseni, se izvede koda, ki je prikazana na sliki 16. Za lažjo povezavo smo uporabili programsko knjižnico, ki nam z metodami, kot je na primer PutData, omogoča, da se podatki, ki jih je vnesel uporabnik, nato vnesejo v podatkovno bazo.

```

registration_button.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View view)
    {
        String[] field = new String[3];
        field[0] = "username";
        field[1] = "password";
        field[2] = "email";
        String[] data = new String[3];
        data[0] = Username.getText().toString();
        data[1] = Password.getText().toString();
        data[2] = Email.getText().toString();
        boolean signupReady = true;
        String Failmessage = "";

        if(Username.length() <= 5)
        {
            signupReady = false;
            Failmessage = "Uporabniško ime ne sme biti krajše od petih znakov";
        }

        if(Username.toString().isEmpty())
        {
            signupReady = false;
            Failmessage = "Obvezno polje";
        }
    }
}

```

Slika 15: Koda gumba za registracijo.

```

if(signupReady) {
    PutData putData = new PutData( url: "http://192.168.0.31/Active/php/signup.php", method: "POST", field, data);
    if (putData.startPut()) {
        if (putData.onComplete()) {
            String result = putData.getResult();
            Toast.makeText(getApplicationContext(), result, Toast.LENGTH_SHORT).show();
            if (result.equals("Registracija uspesna")) {
            }
        }
    }
}
else
{
    Toast.makeText(getApplicationContext(), Failmessage, Toast.LENGTH_SHORT).show();
}
}

```

Slika 16: Koda za PutData metodo pri registraciji.

Podatke, pridobljene pri registraciji, potrebujemo za prijavo. Ponovno uporabimo metodo PutData, ki je prikazana na sliki 17, da podatke vstavimo v bazo. Ta jih pregleda, če se ujemajo s tistimi, ki so bili vneseni pri registraciji. Prijava je uspešna pod pogojem, da je uporabnik vnesel pravilne podatke.

```

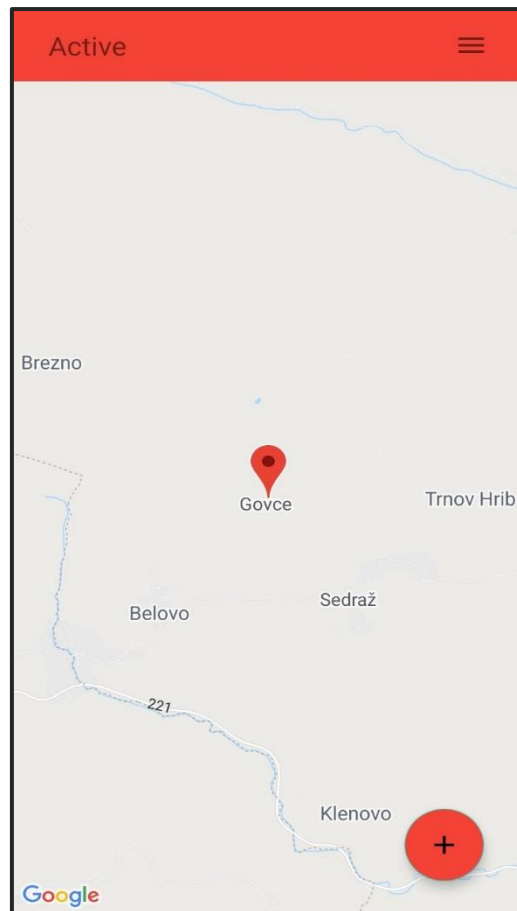
login_button.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View view)
    {
        String[] field = new String[2];
        field[0] = "username";
        field[1] = "password";
        String[] data = new String[2];
        data[0] = Username.getText().toString();
        data[1] = password.getText().toString();

        PutData putData=new PutData( url: "http://192.168.0.31/Active/php/login.php", method: "POST", field, data);
        if(putData.startPut()) {
            if(putData.onComplete()){
                String result=putData.getResult();
                if(result.equals("Uspesna prijava")){
                    Intent intent = new Intent(getApplicationContext(),MainActivity.class);
                    startActivity(intent);
                    SharedPreferences prefs = PreferenceManager.getDefaultSharedPreferences(getApplicationContext());
                    prefs.edit().putString("USERNAME",Username.getText().toString()).apply();
                    prefs.edit().putString("PASSWORD",password.getText().toString()).apply();
                }
                else
                {
                    Toast.makeText(getApplicationContext(),result,Toast.LENGTH_SHORT).show();
                }
            }
        }
    }
});

```

Slika 17: Koda gumba za prijavo.

Po prijavi oziroma registraciji se prikaže začetna stran (slika 18). Aplikacija poizveduje o stanju dovoljenja za potrebne funkcije naprave, do katerih potrebuje dostop. Z uporabnikovim dovoljenjem lokacijskih storitev, ki uporabljajo GPS lokacijske senzorje na telefonu, prikaže trenutno lokacijo uporabnika, kot je prikazano na sliki 18. Koda za pridobivanje trenutne lokacije je na sliki 19.



Slika 18: Začetna stran aplikacije.

```
private void Trenutnatakacija()
{
    if (ActivityCompat.checkSelfPermission(context: this, Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED &&
        ActivityCompat.checkSelfPermission(context: this, Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED)
    {
        // TODO: Consider calling
        // ActivityCompat#requestPermissions
        // here to request the missing permissions, and then overriding
        // public void onRequestPermissionsResult(int requestCode, String[] permissions,
        // int[] grantResults)
        // to handle the case where the user grants the permission. See the documentation
        // for ActivityCompat#requestPermissions for more details.
        return;
    }
    Task<Location> task = client.getLastLocation();

    task.addSuccessListener(new OnSuccessListener<Location>()
    {
        @Override
        public void onSuccess(Location location)
        {
            if(location != null)
            {
                mapFragment.getMapAsync(new OnMapReadyCallback()
                {
                    @Override
                    public void onMapReady(GoogleMap googleMap)
                    {
                        LatLng latlng = new LatLng(location.getLatitude(), location.getLongitude());

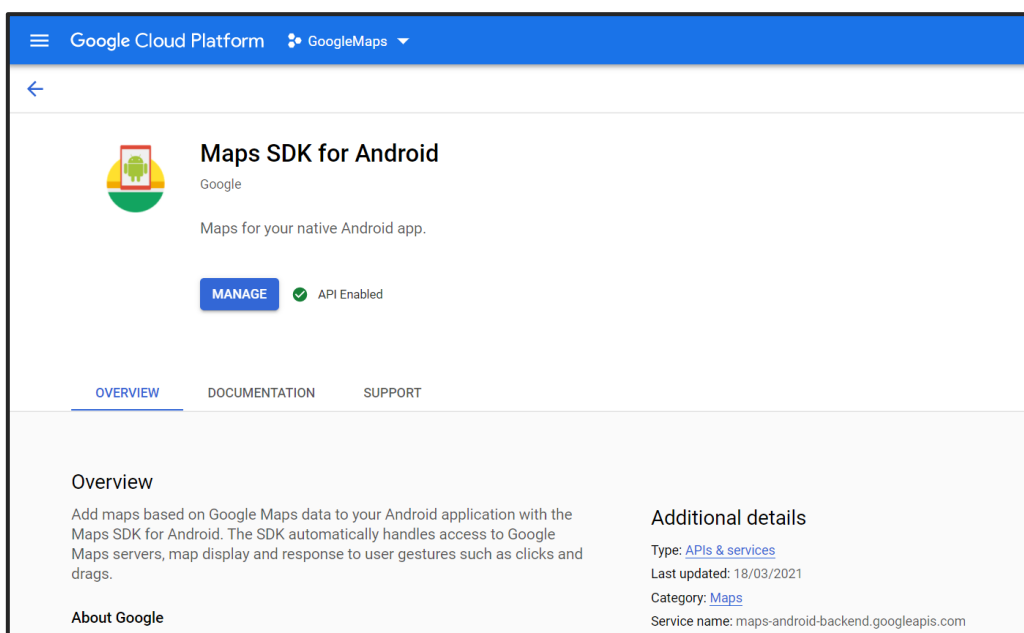
                        MarkerOptions markerOptions = new MarkerOptions().position(latlng);

                        googleMap.animateCamera(CameraUpdateFactory.newLatLngZoom(latlng, 14));

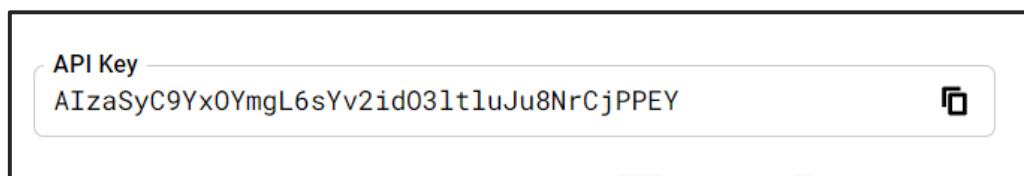
                        googleMap.addMarker(markerOptions);
                    }
                });
            }
        }
    });
}
```

Slika 19: Koda za iskanje trenutne uporabnikove lokacije.

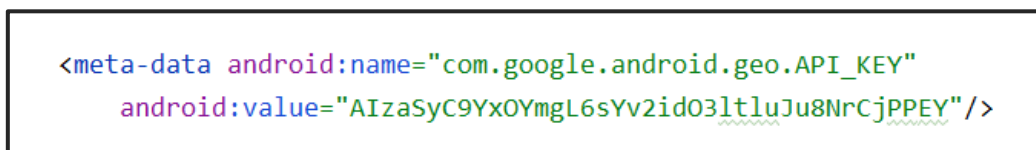
Čeprav smo sprva želeli, da bi naša aplikacija delovala tudi brez internetne povezave, smo ugotovili, da to ne bo mogoče. Najbolj varen in zanesljiv način, da so podatki o lokaciji uporabnika na voljo, je z vzpostavljeno internetno povezavo in z implementacijo GoogleMaps zemljevida. Za implementacijo GoogleMaps zemljevida je potrebno pridobiti tako imenovan API ključ. Za pridobitev je potreben GCP račun, ki smo ga že imeli, saj smo s to storitvijo že delali. Ustvarili smo nov projekt in v tako imenovani knjižnici poiskali storitev Maps SDK for Android, ki jo ponuja Google. Pridobljeni ključ smo vključili v aplikacijo, kot je razvidno s slike 22.



Slika 20: Omogočanje storitve Maps SDK for Android v knjižnici GCP.



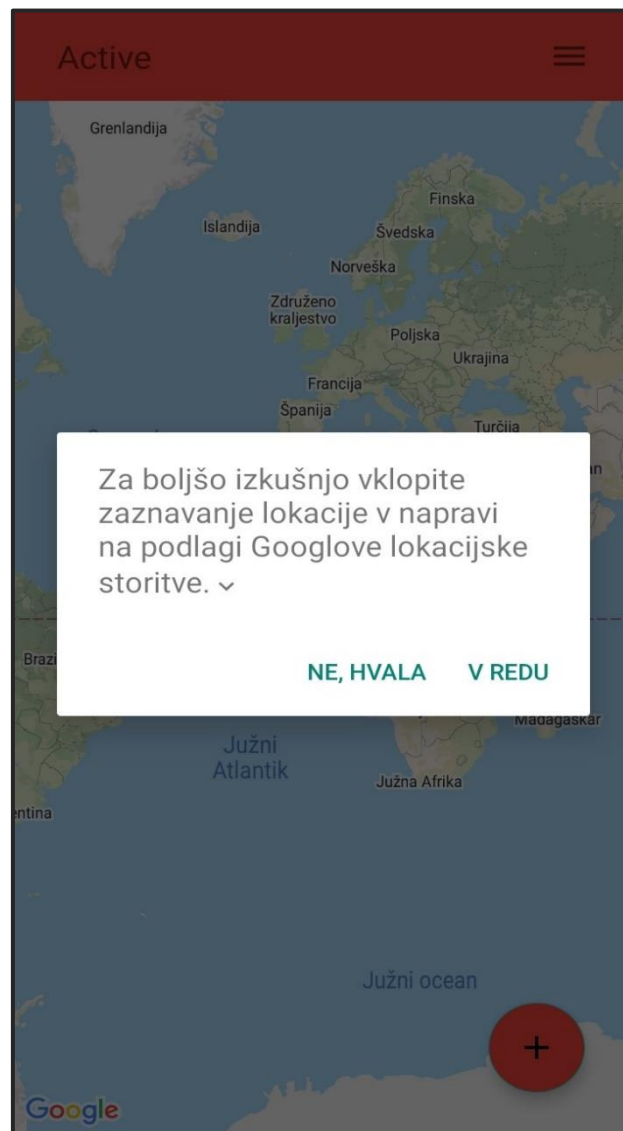
Slika 21: Pridobljen API ključ.



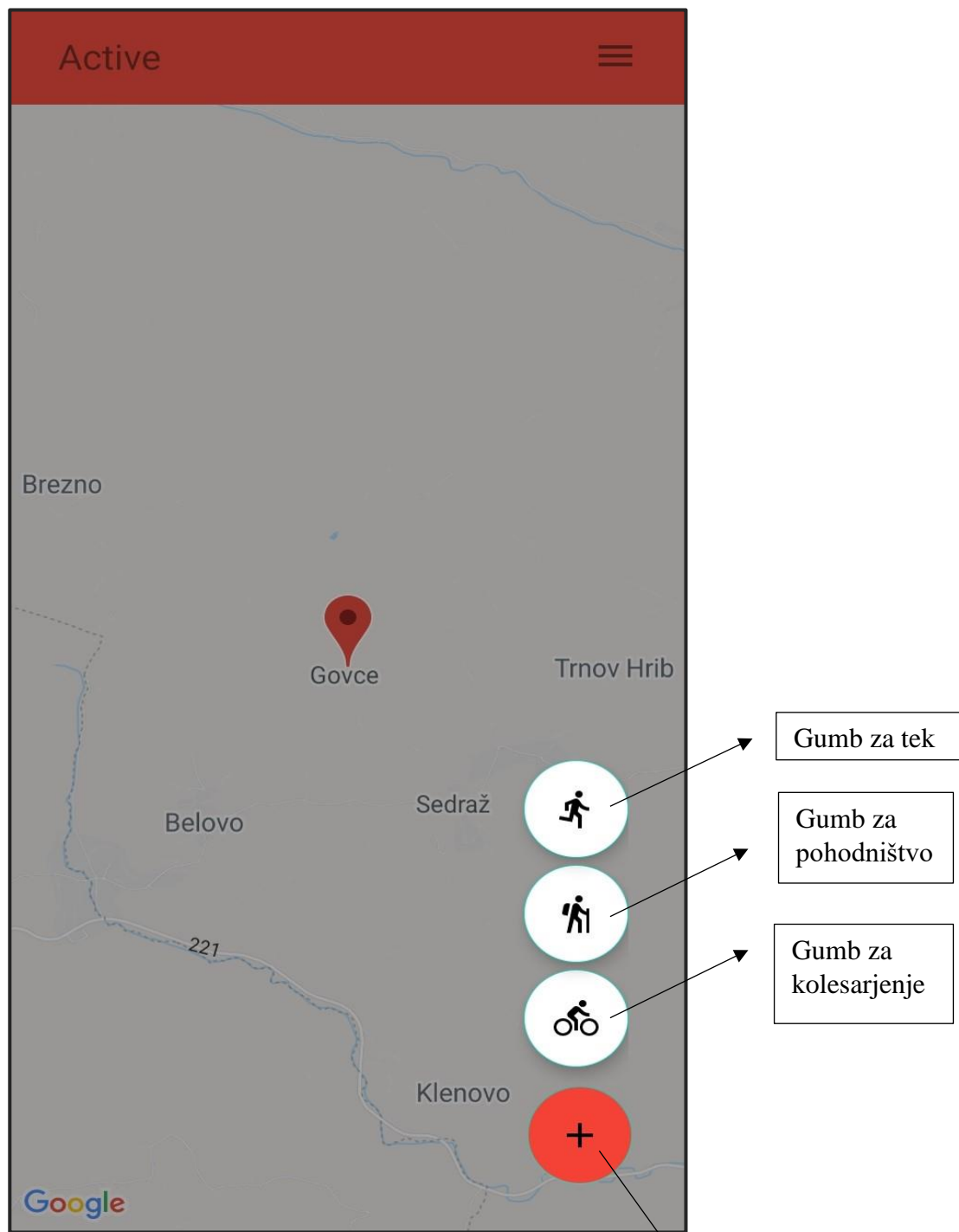
Slika 22: Implementacija API ključa v kodo.



Če lokacijske in internetne storitve niso vklopljene, aplikacija od uporabnika zahteva, da jih omogoči, saj so bistvenega pomena za delovanje aplikacije (slika 23). Uporabnik nato izbere vrsto športne dejavnosti, ki jo želi opravljati. Izbira lahko med tekom, pohodništvom in kolesarjenjem (slika 24). Ob kliku na izbrani gumb se sproži tako imenovan dogodek. Koda je vidna na sliki 25.



Slika 23: Zahteva za omogočanje lokacijske storitve.



Slika 24: Izbiranje športne dejavnosti.

Gumb za prikaz izbire dejavnosti

```

gumb_kolesarjenje.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View v)
    {
        String textRead = "Kolesarjenje";
        Intent intent = new Intent(getApplicationContext(), NovaAktivnost.class);
        intent.putExtra( name: "EXTRA_STRING", textRead);
        startActivityForResult(intent, requestCode: 1);
        closeFABMenu();
    }
});

gumb_pohodnistvo.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View v)
    {
        String textRead = "Pohodništvo";
        Intent intent = new Intent(getApplicationContext(), NovaAktivnost.class);
        intent.putExtra( name: "EXTRA_STRING", textRead);
        startActivityForResult(intent, requestCode: 1);
        closeFABMenu();
    }
});

gumb_tek.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View v)
    {
        String textRead = "Tek";
        Intent intent = new Intent(getApplicationContext(), NovaAktivnost.class);
        intent.putExtra( name: "EXTRA_STRING", textRead);
        startActivityForResult(intent, requestCode: 1);
        closeFABMenu();
    }
});

```

Slika 25: Koda za premik na novo stran.

Ta dogodek odpre novo stran za sledenje športne dejavnosti, ki jo je uporabnik izbral. Glede na izbrano športno dejavnost se v navigacijski vrstici izpiše njeno ime. Koda je na sliki 26.

```

Intent intent = getIntent();

if(intent.hasExtra( name: "EXTRA_STRING"))
{
    String textRead = intent.getStringExtra( name: "EXTRA_STRING");
    TextView text = findViewById(R.id.izbira);
    text.setText(textRead);
}

```

Slika 26: Koda za izpis imena izbrane športne dejavnosti.

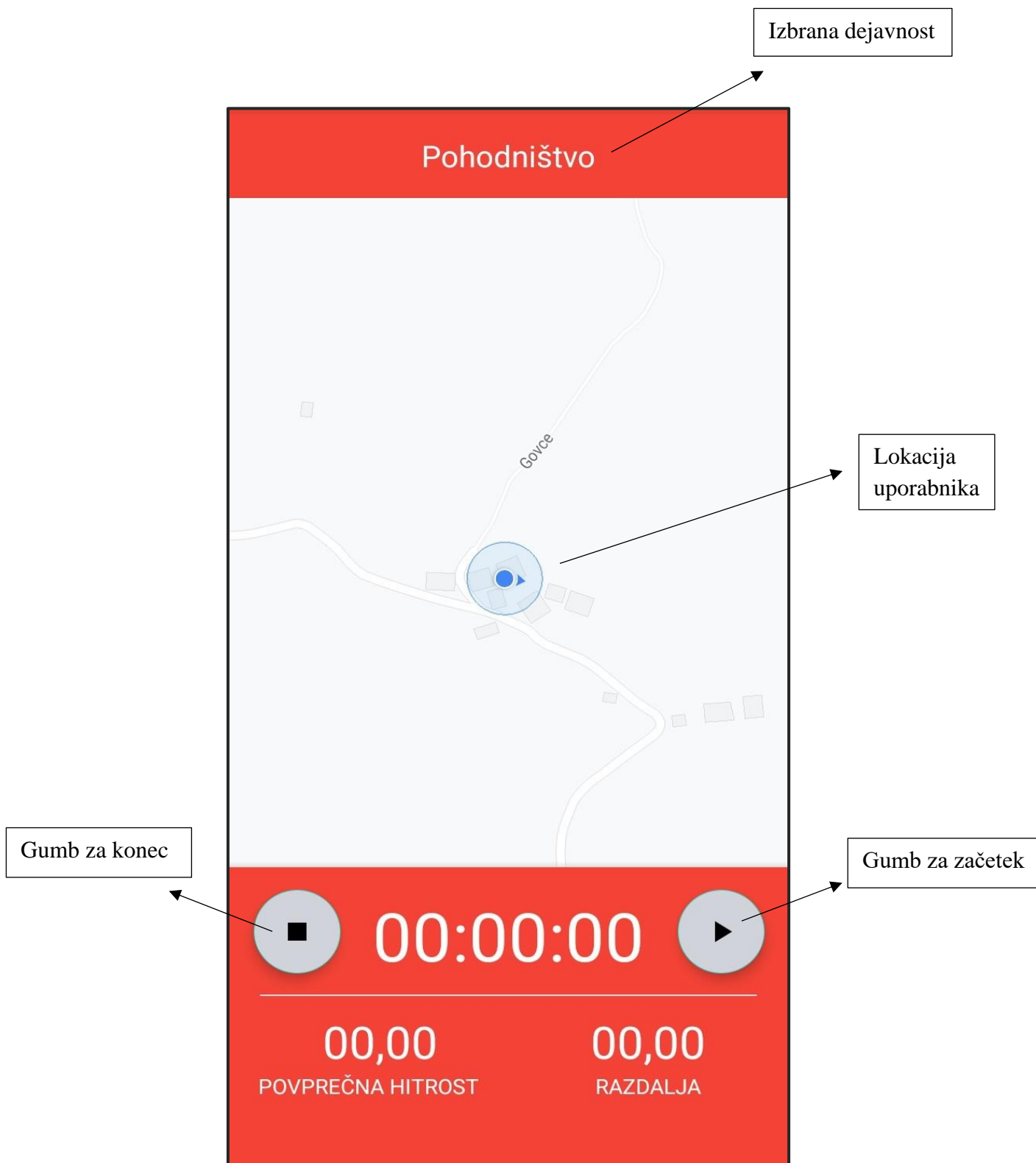
Na sliki 28 je prikazan primer za izbiro dejavnosti pohodništvo. Ob kliku na gumb za začetek, ki je na sliki 28 na desni strani, aplikacija začne s sledenjem. Ob dejavnosti sledi, preračunava in prikazuje čas, povprečno hitrost in opravljeno razdaljo športne dejavnosti uporabnika. Povprečna hitrosti in opravljena razdalja sta merjeni v kilometrih na uro. Poleg tega na zemljevid sproti izrisuje tudi pot, ki jo uporabnik opravi. Vse te podatke pridobiva na podlagi uporabnikove lokacije. Povprečna hitrost se preračunava tako, da delimo skupno razdaljo s časom, ki ga je uporabnik dosegel na tej razdalji. Ko se uporabnik med športno dejavnostjo premika, se glede na njegovo spremenjeno lokacijo posodablja razdalja, povprečna hitrost in izrisana pot na zemljevidu. Koda za risanje poti je prikazana na sliki 27. Gumb za začetek se ob kliku spremeni v gumb, ki omogoča uporabniku začasno prekinitvev sledenja. Če želi uporabnik končati s sledenjem, pa ima na voljo gumb, ki je na sliki 28 na levi strani.

```

private static void RisanjePoti(final GoogleMap map, final Location zadnja_lokacija, final Location nova_lokacija)
{
    Polyline polyline = map.addPolyline(new PolylineOptions()
        .add(new LatLng(zadnja_lokacija.getLatitude(), zadnja_lokacija.getLongitude()),
            new LatLng(nova_lokacija.getLatitude(), nova_lokacija.getLongitude()))
        .width(20)
        .color(Color.argb( alpha: 90, red: 0, green: 130, blue: 255))
    );
    polylines.add(polyline);
}

```

Slika 27: Koda za izrisovanje uporabnikove poti.



Slika 28: Sledenje za športno dejavnost pohodništvo.

Po končani aktivnosti se podatki shranijo. To smo dosegli z metodo `saveData`, ki je prikazana na sliki 29. Tako so podatki s pomočjo PHP kode `dataSave` (slika 30) vstavljeni v podatkovno bazo. Vsak podatek je vstavljen na točno določeno mesto v tabeli.

```

void saveData() {
    String[] field = new String[6];
    field[0] = "username";
    field[1] = "password";
    field[2] = "kategorija";
    field[3] = "cas";
    field[4] = "pot";
    field[5] = "hitrost";
    String[] data = new String[6];
    data[0] = MainActivity.preferences.getString( key: "USERNAME", defValue: "");
    data[1] = MainActivity.preferences.getString( key: "PASSWORD", defValue: "");
    data[2] = MainActivity.kategorija;
    data[3] = timeInMinutes();
    data[4] = distanceSaveString();
    data[5] = speedToFormatString();
    PutData putData=new PutData( url: "http://192.168.0.31/Active/php/dataSave.php", method: "POST", field, data);
    if(putData.startPut()) {
        if (putData.onComplete()) {
            Log.d( tag: "sql return", putData.getResult());
        }
    }
}
}

```

Slika 29: Koda za vnašanje aktivnosti v bazo podatkov.

```

<?php
require "DataBase.php";
$db = new DataBase();
if (isset($_POST['username']) && isset($_POST['password']) && isset($_POST['kategorija']) && isset($_POST['cas']) && isset($_POST['pot']) && isset($_POST['hitrost']))
{
    if ($db->dbConnect())
    {
        echo $db->dataSave("aktivnosti", $_POST['username'], $_POST['password'], $_POST['kategorija'], $_POST['cas'], $_POST['pot'], $_POST['hitrost']);
    }
}
else echo "All fields are required";
?>

```

Slika 30: Koda za povezavo med bazo podatkov in `saveData` metodo.

Za prikaz teh podatkov smo ustvarili PHP kodo `dataRequest` za pridobivanje podatkov iz podatkovne baze (slika 31). Ti podatki so nato vneseni v polja, kot je prikazano na sliki 32.

```

<?php
require "DataBase.php";
$db = new DataBase();
if (isset($_POST['username']) && isset($_POST['password']))
{
    if ($db->dbConnect())
    {
        if ($db->dataRequest("aktivnosti", $_POST['username'], $_POST['password']))
        {
            echo $db->dataRequest("aktivnosti", $_POST['username'], $_POST['password']);
        }
    }
}
else echo "All fields are required";
?>

```

Slika 31: Koda za povezavo med bazo podatkov in PutData metodo.

```

PutData putData=new PutData( url: "http://192.168.0.31/Active/php/dataRequest.php", method: "POST", field, data);
if(putData.startPut())
{
    if(putData.onComplete())
    {
        String result = putData.getResult();
        int size = Character.getNumericValue(result.charAt(0));
        if(size > 0)
        {
            String[] split = new String[size * 4];

            String[] s1 = new String[size];
            String[] s2 = new String[size];
            String[] s3 = new String[size];
            String[] s4 = new String[size];

            int counterWords = 0;
            int counterArray = 1;
            String CurrentWord = "";
            for (int i = 2; i < result.length(); i = i + 1) {
                if (result.charAt(i) == ',') {
                    if (counterArray == 1) {
                        s1[counterWords] = CurrentWord;
                    }
                    if (counterArray == 2) {
                        s2[counterWords] = CurrentWord;
                    }
                    if (counterArray == 3) {
                        s3[counterWords] = CurrentWord;
                    }
                    if (counterArray == 4) {
                        s4[counterWords] = CurrentWord;
                    }
                    CurrentWord = "";
                    counterArray++;
                } else {
                    CurrentWord += result.charAt(i);
                }
            }
            if (counterArray == 5) {
                counterArray = 1;
                counterWords++;
            }
        }
    }
}

```

Slika 32: Koda za pridobitev podatkov o aktivnostih iz baze podatkov.

Nato smo s kodo za Adapter, ki je na slikah 33, 34 in 35, povezali podatke iz polja z njihovim mestom v uporabniškem vmesniku. S tem si lahko uporabniki vse opravljene aktivnosti ponovno ogledajo s klikom na možnost za pretekle aktivnosti, ki se nahaja v desnem zgornjem kotu začetne strani aplikacije. Zaradi uporabe podatkovne baze, v kateri so te aktivnosti shranjene in razvrščene po uporabnikih, omogočimo, da se pri prijavi več uporabnikov na eni napravi prikažejo samo aktivnosti trenutno prijavljenega uporabnika.

```
recyclerView.setLayoutManager(new LinearLayoutManager( context: this));
Adapter Adapter = new Adapter( ct: this, activity: PretekleAktivnosti.this, s1, s2, s3, s4);
recyclerView.setAdapter(Adapter);
```

Slika 33: Koda za klicanje metode Adapter.

```
public Adapter (Context ct, Activity activity,String s1[],String s2[],String s3[],String s4[]){
    this.context = ct;
    this.activity = activity;
    data1=s1;
    data2=s2;
    data3=s3;
    data4=s4;
}
```

Slika 34: Koda za deklariranje metode Adapter.

```
@Override
public void onBindViewHolder(@NonNull MyViewHolder holder, int position) {

    holder.text1.setText(data1[position]);
    holder.text2.setText(data2[position]);
    holder.text3.setText(data3[position]);
    holder.text4.setText(data4[position]);

    Intent intent = new Intent(context, PretekleAktivnosti.class);
}

@Override
public int getItemCount() { return data1.length; }
public class MyViewHolder extends RecyclerView.ViewHolder{

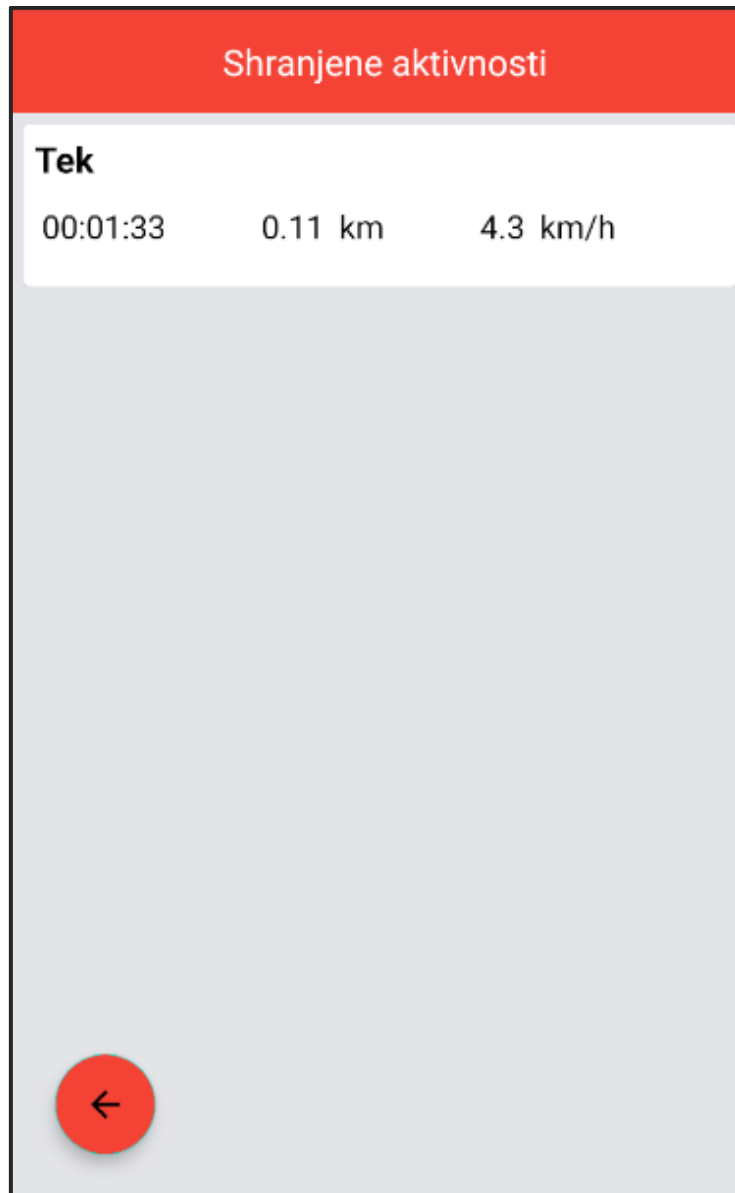
    TextView text1,text2,text3,text4;

    public MyViewHolder(@NonNull View itemView) {
        super(itemView);
        text1=itemView.findViewById(R.id.tv_kategorija);
        text2=itemView.findViewById(R.id.tv_cas);
        text3=itemView.findViewById(R.id.tv_povpHitrost);
        text4=itemView.findViewById(R.id.tv_pot);
    }
}
```

Slika 35: Koda za metodo Adapter, ki poskrbi za izpis podatkov.



S slike 36 je razvidno, kako so podatki o preteklih aktivnosti prikazani uporabniku. Po končani aktivnosti, ki jo uporabnik shrani, se izpišejo podatki o vrsti aktivnosti, ki jo je uporabnik opravljajal, njenem trajanju, številu kilometrov in povprečni hitrosti, ki jo je pri aktivnosti dosegel.



Slika 36: Shranjeni podatki o športni dejavnosti.

### 3.3 TEŽAVE PRI USTVARJANJU APLIKACIJE

Najprej smo se morali odločiti, katere športne dejavnosti in katere funkcije bomo vključili v našo aplikacijo. K tej odločitvi so bistveno prispevali odgovori anketirancev na vprašanja, ki smo jim jih zastavili v anketi. Na podlagi rezultatov ankete smo v aplikacijo vključili tiste športne dejavnosti, za katere so anketiranci odgovorili, da jih najpogosteje opravljajo, torej tek, pohodništvo in kolesarjenje. Ker smo iz rezultatov ankete razbrali, da je anketirancem pri športnih aplikacijah najpomembnejše beleženje športnih aktivnosti smo se osredotočili predvsem na to. Kar nekaj anketirancev, natančneje 25 % pa je izbralo, da je zanje najpomembnejše delovanje brez interneta. Zaradi tega smo poskušali narediti aplikacijo, ki bi delovala tudi brez internetne povezave, vendar smo ugotovili, da to ne bo mogoče. Aplikacija temelji na lokaciji uporabnika, ti podatki pa so najbolj zanesljivi z vzpostavljeno internetno povezavo.

Naslednja težava je bila izbrati platformo, ki bi nam ustrezala glede na znanje, ki smo ga pridobili v letih šolanja na srednji šoli za računalništvo. Po premisleku se nam je zdel najprimernejši program Android Studio, saj smo ga že obravnavali pri strokovnem predmetu. Želeli smo nadgraditi znanje, ki smo ga pridobili pri pouku.

Pri odločanju o imenu, ki ga mora imeti vsaka aplikacija, smo se sprva želeli domisliti posebnega, unikatnega imena. Ker se je takšnega imena, ki bi bil všečen in bi pritegnil ljudi, težko domisliti, smo to zamisel opustili. Odločili smo se, da bomo ubrali varnejšo pot in aplikaciji dali preprosto in neposredno ime. Zato smo jo poimenovali Active.

## 4 MOŽNOSTI NADGRADNJE V PRIHODNOSTI

V prihodnosti bi radi aplikacijo nadgradili. Radi bi dodali izbiranje med še več športnimi dejavnostmi, kot je na primer rolanje, trening z utežmi, fitness telovadba. Pri treningu z utežmi bi lahko uporabnik izbral med že vnaprej pripravljenimi vajami ali pa si naredil svoje in vpisal težo uteži ter število ponovitev. Nadgradili bi tudi shranjevanje športnih dejavnosti, da bi se poleg povprečne hitrosti, razdalje in časa shranila slika zemljevida poti, ki jo je uporabnik opravil. Poleg tega pa bi uporabnik svojo dejavnost lahko preimenoval in mogoče celo dodal kakšno sliko, ki jo je ustvaril med potjo.

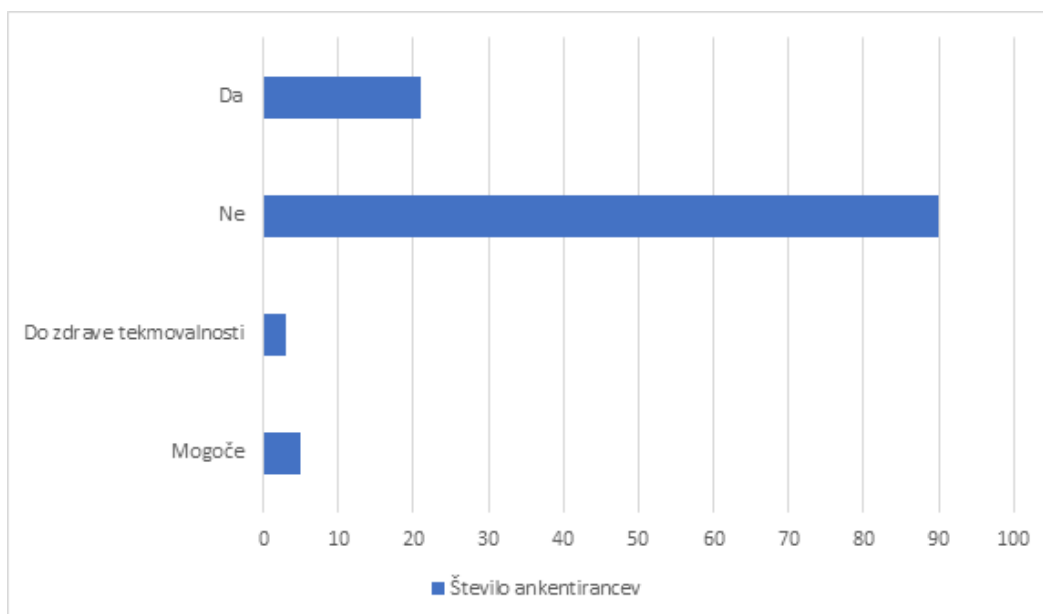
Aplikacija bi izpisala tudi približek porabljenih kalorij po vsaki aktivnosti. Uporabnik bi lahko dostopal do statistik, kjer bi bila prikazana povprečna razdalja in čas njegovih aktivnosti, pa tudi njegova najdaljša aktivnost (glede na čas in razdaljo). Za bolj personalizirano izkušnjo pa bi dodali možnost spreminjana teme aplikacije, kjer bi uporabnik lahko izbral med mnogo različnimi barvami grafičnega vmesnika aplikacije. Aplikacija bi omogočala tudi funkcijo, da uporabnik vnese cilje, ki jih želi doseči, kot je na primer preteči 20 kilometrov v enem tednu.

## 5 PREDSTAVITEV REZULTATOV RAZISKOVALNE NALOGE

Naša prva hipoteza je bila, da se bo Android Studio pokazal kot dobra izbira platforme za ustvarjanje aplikacije. Hipotezo smo potrdili, saj smo po preizkusu različnih razvojnih okolij ugotovili, da ima le Android Studio vse potrebne funkcije za kvaliteten in hiter razvoj naše aplikacije. Ponuja inteligentni urejevalnik kode, ki omogoča hitrejšo programiranje in skrbi, da nismo izgubljali časa pri dodatnem pisanju. Med samim razvojem z emulatorjem omogoča hitro izdelavo prototipov in s tem tudi sprotne preizkušanje pravilnosti delovanja aplikacije. Tako smo lahko pri vsaki negotovosti aplikacijo hitro testirali. Zelo prav nam je prišlo tudi orodje za nadzor različic, ki nam je omogočilo vrnitev na stare verzije aplikacije, ko nam kakšen končni izid delovanja ni ustrezal. Poleg vseh teh prednosti pa ponuja še možnost gradnje aplikacije za končno objavo z vso potrebno dokumentacijo. Edina težava, ki smo jo imeli s platformo, se je pojavila pri njeni posodobitvi, ko so se pri konfiguraciji naše aplikacije pojavile napake zaradi nove verzije sistema. Kljub temu smo te napake hitro odpravili.

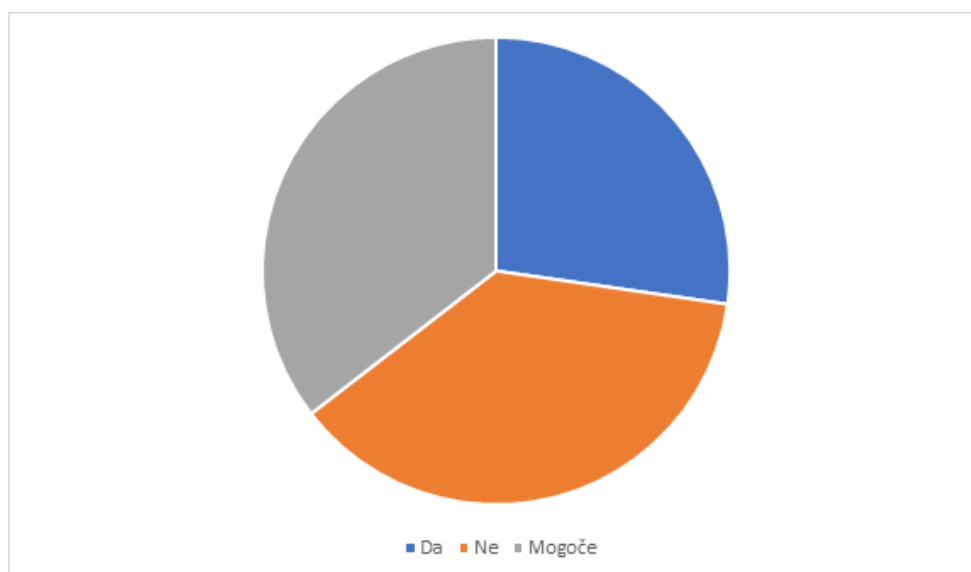
Druga hipoteza je bila, da nam bo programski del predstavljal večjo težavo kot grafična podoba aplikacije. Android Studio ima zelo preprost urejevalnik za grafično podobo aplikacije. Vse elemente lahko dodamo s tako imenovano drag and drop (povleci in spusti) metodo, ki zagotavlja hitro dodajane in premikanje elementov. Po drugi strani pa ima veliko bolj kompleksen način za programiranje. Kot primer lahko navedemo del, kjer smo želeli, da se uporabniku med samo aktivnostjo izrisuje pot. Za tem je bilo potrebno veliko povezav med različnimi deli kode. Za to smo porabili veliko več časa kakor za samo podobo aplikacije in s tem tudi potrdili našo hipotezo.

Zadnja hipoteza temelji na dejstvu, da se pri uporabi športnih aplikacij pojavlja nezdrava tekmovalnost, ki se je večina mladostnikov ne zaveda. To hipotezo smo potrdili z rezultati, pridobljenimi iz ankete. Anketirance smo vprašali, ali menijo, da pri športnih aplikacijah s spremljanjem napredka prijateljev prihaja do nezdrave tekmovalnosti. Iz grafa na sliki 36 lahko razberemo, da je kar 74 % odgovorilo z ne, le 19 % z da, 4 % pa z mogoče; ostali 3 % menijo, da prihaja do zdrave tekmovalnosti.



Slika 37: Graf o mnenju anketirancev glede nezdrave tekmovalnosti.

Na vprašanje, ali bi raje uporabljali aplikacijo, s katero bi spremljali le svoje športne aktivnosti je 27 % anketirancev odgovorilo z da, 37 % z ne in 35 % z mogoče. Kot argument za so poudarili, da bi se lahko s takšno aplikacijo osredotočili le nase, imeli bi več zasebnosti in da jim ni pomembno spremljanje dosežkov drugih. Kot argument proti pa so navedli, da športne dosežke radi primerjajo z drugimi in da s tem pridobijo dodatno motivacijo in tekmovalnost.



Slika 38: Graf o mnenju anketirancev glede spremljanja le svojih aktivnosti.

## 6 ZAKLJUČEK

Med raziskavo nam je največ preglavic povzročala funkcija shranjevanja športnih dejavnosti, saj smo imeli z njo precej več dela, kot smo sprva pričakovali. Anketo smo poslali mentorju, ki jo je posredoval med dijake, zato težav s premajhnim številom anketirancev nismo imeli. S tem smo pridobili vse potrebne odgovore na vprašanja, ki so nam pomagali tako pri izdelavi aplikacije kot tudi pri rezultatih hipotez. Raziskava v tej raziskovalni nalogi nam je predstavila malo drugačen zorni kot na tekmovalnost, ki ni tako nedolžna, kot mogoče izgleda na prvi pogled. Kljub težavam, na katere smo med nalogo naleteli, smo raziskovalno nalogo uspešno zaključili. Za izdelovanje aplikacije smo porabili veliko prostega časa, vendar smo veseli, da nam je uspelo, in že komaj čakamo, da se bomo v prihodnje lotili nadgradnje, s katero bomo aplikacijo še izboljšali.

Aplikacija je še vedno v razvojni fazi, vsak dan pa se število uporabnikov povečuje. V času zagovora na občinskem nivoju, ko smo aplikacijo uporabljali zgolj mi in člani naših družin, je bilo uporabnikov približno pet. Po zagovoru na občinskem nivoju smo aplikacijo razširili med sošolce, prijatelje, znance in jo trenutno uporablja približno petdeset oseb. Od dosedanjih uporabnikov smo dobili pozitiven odziv glede uporabe. Če bomo še naprej dobivali dobre povratne informacije in bo aplikacijo uporabljalo čedalje več ljudi, jo bomo z omenjenimi možnostmi nadgradnje naložili tudi v trgovino Google Play.

## 7 VIRI IN LITERATURA

- [1] Acronis. *Google Cloud Platform: What it is, how to use it, and how to compares*. Pridobljeno 04. 04. 2021 iz <https://www.acronis.com/en-us/articles/google-cloud-platform/>
- [2] Alen Ravbar. (2011). *Sistemi za upravljanje z vsebinami*. Pridobljeno 04. 04. 2021 iz <http://eprints.fri.uni-lj.si/1457/1/Ravbar1.pdf>
- [3] Android Developers. (24. 02. 2021). *Meet Android Studio*. Pridobljeno 04. 04. 2021 iz <https://developer.android.com/studio/intro>
- [4] eNSA. *Uvod v PHP*. Pridobljeno 04. 04. 2021 iz <https://nsa-splet.si/php/uvod/php-uvod-01.php>
- [5] Guru99. *SQL vs NoSQL: What's the Difference Between SQL and NoSQL*. Pridobljeno 04. 04. 2021 iz <https://www.guru99.com/sql-vs-nosql.html>
- [6] HeidiSQL. *What's this, News, Features*. Pridobljeno 04. 04. 2021 iz <https://www.heidisql.com/>
- [7] <https://blog.google/products/google-cloud/cloud-covered-what-was-new-with-google-cloud-in-october-2019/>. Pridobljeno 06. 04. 2021.
- [8] [https://en.wikipedia.org/wiki/HeidiSQL#/media/File:HeidiSQL\\_logo\\_image.png](https://en.wikipedia.org/wiki/HeidiSQL#/media/File:HeidiSQL_logo_image.png). Pridobljeno 06. 04. 2021.
- [9] [https://miro.medium.com/max/2024/1\\*Z5SpsmDvk67BIImwHvh\\_cQ.png](https://miro.medium.com/max/2024/1*Z5SpsmDvk67BIImwHvh_cQ.png). Pridobljeno 06. 04. 2021.
- [10] [https://miro.medium.com/max/454/1\\*A5KT219JQR\\_vg7F4OJ-Qow.png](https://miro.medium.com/max/454/1*A5KT219JQR_vg7F4OJ-Qow.png). Pridobljeno 06. 04. 2021.
- [11] <https://www.googlenewsapp.com/android-studio-4-1/>. Pridobljeno 06. 04. 2021.
- [12] [https://www.logo.wine/logo/Kotlin\\_\(programming\\_language\)](https://www.logo.wine/logo/Kotlin_(programming_language)). Pridobljeno 06. 04. 2021.
- [13] <https://www.stickpng.com/img/icons-logos-emojis/tech-companies/java-logo>. Pridobljeno 06. 04. 2021.

- [14] phpMyAdmin. *About, Features*. Pridobljeno 04. 04. 2021 iz <https://www.phpmyadmin.net/>
- [15] Revija Društva študentov psihologije Slovenija. (05. 04. 2019). *PANIKA*. Pridobljeno 04. 04. 2021 iz Motivacija v športu: <http://revijapanika.si/2019/04/05/motivacija-v-sportu/>
- [16] Techliance. *JAVA VS KOTLIN: WHICH IS BEST FOR ANDROID APP DEVELOPMENT*. Pridobljeno 04. 04. 2021 iz <https://blog.techliance.com/java-vs-kotlin/>
- [17] Uživajmo v zdravju. *Zakaj je telesna dejavnost tako pomembna za naše zdravje?* Pridobljeno 04. 04. 2021 iz <http://www.uzivajmovzdravju.si/index.php/zdravje/telesna-dejavnost/>
- [18] Wikipedia. *XAMPP*. Pridobljeno 04. 04. 2021 iz <https://en.wikipedia.org/wiki/XAMPP>



## 8 PRILOGE

### 8.1 PREDSTAVITEV OSEB

- Ime in priimek: Nuša Alja Kralj
  - Naslov: Cvetlična ulica 14, 3313 Polzela
  - Šola: Srednja šola za kemijo, elektrotehniko in računalništvo
  - Hobiji: risanje, branje
- 
- Ime in priimek: Maja Škorjanc
  - Naslov: Govce 4, 3270 Laško
  - Šola: Srednja šola za kemijo, elektrotehniko in računalništvo
  - Hobiji: pohodništvo, druženje z družino in prijatelji

### 8.2 ANKETNI VPRAŠALNIK

Uvod

Pozdravljeni. Sva dijakinji programa tehnik računalništva in v okviru raziskovalne naloge delava na izdelavi športne aplikacije. Prosili bi vas, da odgovorite na to anketo, katere vprašanja se navezujejo nanjo. Hvala vam in lep pozdrav!

Stran 1

---

Q1 Obvezno vprašanje

Izberite vaš spol.

Moški

Ženski

Q2 Obvezno vprašanje

V katero starostno skupino spadate?

do 15 let

16 do 25 let

26 do 35 let

36 do 45 let

Več kot 45

**Q3 Obvezno vprašanje**

Koliko ur na teden ste športno aktivni?

- Eno uro
- Dve uri
- Tri do štiri ure
- Vsak dan po eno uro
- Drugo:

**Q4 Obvezno vprašanje**

Ali ste že slišali za športne aplikacije, kot so na primer Strava, Sports Tracker, Runkeeper?

- Da
- Ne

**Q6 Obvezno vprašanje**

Ali uporabljate kakšno športno aplikacijo podobno zgoraj navedenim?

- Da
- Ne

**Q5**

Če ste na prejšnje vprašanje odgovorili z "da", odgovorite še na naslednje vprašanje:  
Za katero športno aktivnost jo največkrat uporabljate?

Možnih je več odgovorov

- Tek
- Hojo
- Kolesarjenje
- Drugo:

**Q7 Obvezno vprašanje**

Kaj se vam zdi pri takšnih aplikacijah najpomembnejše?

- Beleženje aktivnosti
- Delovanje brez interneta
- Beleženje vnosa vode / hrane
- Drugo:

**Q8 Obvezno vprašanje**

Kaj vas pri trenutnih športnih aplikacijah najbolj moti?

- Netočnost podatkov
- Zapleteno za uporabo
- Grafična podoba
- Drugo:

**Q9 Obvezno vprašanje**

Kako pomembno se vam zdi spremljanje športnega napredka prijateljev?

- Ni pomembno
- Srednje pomembno
- Zelo pomembno

**Q10 Obvezno vprašanje**

Ali menite, da pri športnih aplikacijah s spremljanje napredka prijateljev prihaja do nezdrave tekmovalnosti?

- Da
- Ne
- Drugo:

**Q11 Obvezno vprašanje**

Kako pogosto primerjate svoje športne dosežke z dosežki svojih prijateljev?

- Vedno
- Pogosto
- Včasih
- Redko
- Nikoli
- Drugo:

**Q12 Obvezno vprašanje**

Ali bi raje uporabljali aplikacijo, s katero bi spremljali le svoje športne aktivnosti?

- Da
- Ne
- Mogoče

**Q13 Obvezno vprašanje**

Zakaj da oziroma zakaj ne?

**Q14 Obvezno vprašanje**

Kako pomembna vam je preprostost uporabe aplikacije?

- Ni pomembna
- Srednje pomembna
- Zelo pomembna

**Q15 Obvezno vprašanje**

Ali se vam zdi moteče, kadar vam aplikacije pošiljajo obvestila?

- Da
- Ne
- Občasno

Zaključna stran

**Zaključek**

Odgovorili ste na vsa vprašanja v tej anketi. Hvala za sodelovanje.