Android aplikacija za kreiranje tjednog plana treninga

Zečević, Mato

Undergraduate thesis / Završni rad

2022

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

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:628989

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-07-27**

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

Preddiplomski sveučilišni studij

ANDROID APLIKACIJA ZA KREIRANJE TJEDNOG PLANA TRENINGA

Završni rad

Mato Zečević

Osijek, 2022.

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

Obrazac Z1P - Obrazac za ocjenu završnog rada na preddiplomskom sveučilišnom studiju

Osijek, 02.09.2022.

Odboru za završne i diplomske ispite

Prijedlog ocjene završnog rada na preddiplomskom sveučilišnom studiju

Ime i prezime Pristupnika:	Mato Zečević
Studij, smjer:	Preddiplomski sveučilišni studij Računarstvo
Mat. br. Pristupnika, godina upisa:	R 4443, 22.07.2019.
OIB Pristupnika:	36542039913
Mentor:	Izv. prof. dr. sc. Alfonzo Baumgartner
Sumentor:	,
Sumentor iz tvrtke:	
Naslov završnog rada:	Android aplikacija za kreiranje tjednog plana treninga
Znanstvena grana rada:	**Obradba informacija (zn. polje računarstvo)**
Zadatak završnog rad:	[Rezervirno: Mato Zečević] Potrebno je napraviti Android aplikaciju koja omogućava upravljanje s podacima o treniranju mišića. Na osnovu tih podataka aplikacija bi savjetovala korisnika koje vježbe bi još trebao raditi i za koje grupacije mišića. Aplikacija bi trebala imati mogućnost prikaza podataka o vježbanju za određene vremenske periode.
Prijedlog ocjene završnog rada:	Vrlo dobar (4)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 2 bod/boda Jasnoća pismenog izražavanja: 1 bod/boda Razina samostalnosti: 3 razina
Datum prijedloga ocjene od strane mentora:	02.09.2022.
Datum potvrde ocjene od strane Odbora:	07.09.2022.

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, 08.09.2022.

Ime i prezime studenta:	Mato Zečević
Studij:	Preddiplomski sveučilišni studij Računarstvo
Mat. br. studenta, godina upisa:	R 4443, 22.07.2019.
Turnitin podudaranje [%]:	15

Ovom izjavom izjavljujem da je rad pod nazivom**: Android aplikacija za kreiranje tjednog plana treninga**

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:

IZJAVA

o odobrenju za pohranu i objavu ocjenskog rada

kojom ja __Mato Zečević_____, OIB: __36542039913_____, student/ica Fakulteta elektrotehnike, računarstva i informacijskih tehnologija Osijek na studiju _Preddiplomski sveučilišni studij Računarstvo_, kao autor/ica ocjenskog rada pod naslovom: _Android aplikacija za kreiranje tjednog plana treninga**,**

dajem odobrenje da se, bez naknade, trajno pohrani moj ocjenski rad u javno dostupnom digitalnom repozitoriju ustanove Fakulteta elektrotehnike, računarstva i informacijskih tehnologija Osijek i Sveučilišta te u javnoj internetskoj bazi radova Nacionalne i sveučilišne knjižnice u Zagrebu, sukladno obvezi iz odredbe članka 83. stavka 11. *Zakona o znanstvenoj djelatnosti i visokom obrazovanju* (NN 123/03, 198/03, 105/04, 174/04, 02/07, 46/07, 45/09, 63/11, 94/13, 139/13, 101/14, 60/15).

Potvrđujem da je za pohranu dostavljena završna verzija obranjenog i dovršenog ocjenskog rada. Ovom izjavom, kao autor/ica ocjenskog rada dajem odobrenje i da se moj ocjenski rad, bez naknade, trajno javno objavi i besplatno učini dostupnim:

a) široj javnosti

b) studentima/icama i djelatnicima/ama ustanove

c) široj javnosti, ali nakon proteka 6 / 12 / 24 mjeseci (zaokružite odgovarajući broj mjeseci).

**U slučaju potrebe dodatnog ograničavanja pristupa Vašem ocjenskom radu, podnosi se obrazloženi zahtjev nadležnom tijelu Ustanove.*

__Osijek, 08.09.2022._____ _____

(mjesto i datum) (vlastoručni potpis studenta/ice)

SADRŽAJ

1. UVOD ... 1
1.1. Zadatak završnog rada ... 1
2. USPOREDBA SLIČNIH RJEŠENJA ... 2
2.1. Fitness & Bodybuilding ... 2
2.2. Strong Workout Tracker Gym Log ... 3
2.3. Muscle Booster Workout Planner ... 4
2.4. Vježbe kod kuće – bez opreme ... 4
2.5. Freeletics: Fitness Workouts ... 5
3. TIJEK RAZVOJA APLIKACIJE ... 7
3.1. Korištene tehnologije ... 7
3.1.1. Android Studio ... 7
3.1.2. Programski jezik Kotlin ... 8
3.1.3. Opisni jezik XML ... 8
3.1.4. SQL baza podataka ... 9
3.2. Postupak razvoja aplikacije ... 10
3.2.1. AndroidManifest datoteka ... 10
3.2.2. Aktivnost MainActivity ... 11
3.2.3. Gradle ... 12
3.2.4. RecylerView ... 12
3.2.5. Registracija korisnika ... 15
3.2.6. Funkcionalnost datuma ... 16
3.2.7. Odabir vježbi za određenu grupaciju ... 17
3.2.8. Statistika treniranja ... 18
4. STRUKTURA APLIKACIJE ... 20
4.1. MainActivity ... 20
4.2. AppActivity ... 22
4.2.1. PlannerFragment ... 22
4.2.2. AddWorkoutFragment ... 23
4.2.3. HistoryFragment ... 28
4.2.4. ExercisesFragment ... 29
4.2.5. UserFragment ... 30
5. ZAKLJUČAK ... 32
LITERATURA ... 33
SAŽETAK ... 34
ABSTRACT ... 35
ŽIVOTOPIS ... 36

1. UVOD

U današnje vrijeme korištenje mobilmih uređaja, mobilnih aplikacija i usluga je dostupno svima. Postoje razne aplikacije koje služe u različite svrhe, neke nam omogućuju komunikaciju i zabavu, a određene aplikacije nam pomažu u svakodnevnom životu i obavljanju naših svakodnevnih navika. U tu vrstu aplikacija spadaju i aplikacije orijentirane praćenju napretka u teretani i može se reči da su takve aplikacije postale alat za postizanje određenih ciljeva u fizičkim aktivnostima. Takve aplikacije su vrlo korisne i dosta olakšavaju „posao“ osobama koje su početnici u treniranju sa utezima i planiranju treninga u teretani. Također, vrlo su korisne i osobama koje se već dugi niz godina bave dizanjem utega, jer na jednom mjestu mogu pratiti svoj proces treniranja i napredak.

Završni rad je na temu Android aplikacije za kreiranje tjednog plana treninga. Sam rad se sastoji od pet poglavlja. U poglavlju nakon uvoda prikazano je nekoliko aplikacija koje imaju implenetirane neke od funkcionalnosti koje su slične mojoj. Treće poglavlje je opis postupka izrade aplikacije, opis tehnologija korištenih u izradi, prikaz dijela koda i strukture baze podataka. Predzadnje poglavlje opisuje strukturu same aplikacije, te prolazak kroz zaslone aplikacije i opis funkcionalnosti. Peto poglavlje je namjenjeno za zaključak na osnovu cijeloga rada.

1.1. Zadatak završnog rada

Potrebno je izraditi Android aplikaciju koja omogućava upravljanje podacima o treniranju. Na osnovu tih podataka aplikacija nudi korisniku koje vježbe i grupacije mišića je potrebno raditi raspoređeno po danaima. Aplikacija bi trebala imati mogućnost praćenja podataka o vježbanju, odnosno koje vježbe i grupacije mišića su odrađene u određenom vremenskom periodu.

2. USPOREDBA SLIČNIH RJEŠENJA

Cilj ovog poglavlja je prikazati aplikacije koje su dostupne korisnicima, a imaju slične funkcionalnosti u usporedbi sa razvijenom aplikacijom. Sljedeće aplikacije su odabrane na osnovu popularnosti i ocjena korisnika, te na osnovu samih rješenja. Rješenja odabranih aplikacija su ukratko opisana i objašnjena, te su aktivnosti prikazane slikama.

2.1. Fitness & Bodybuilding

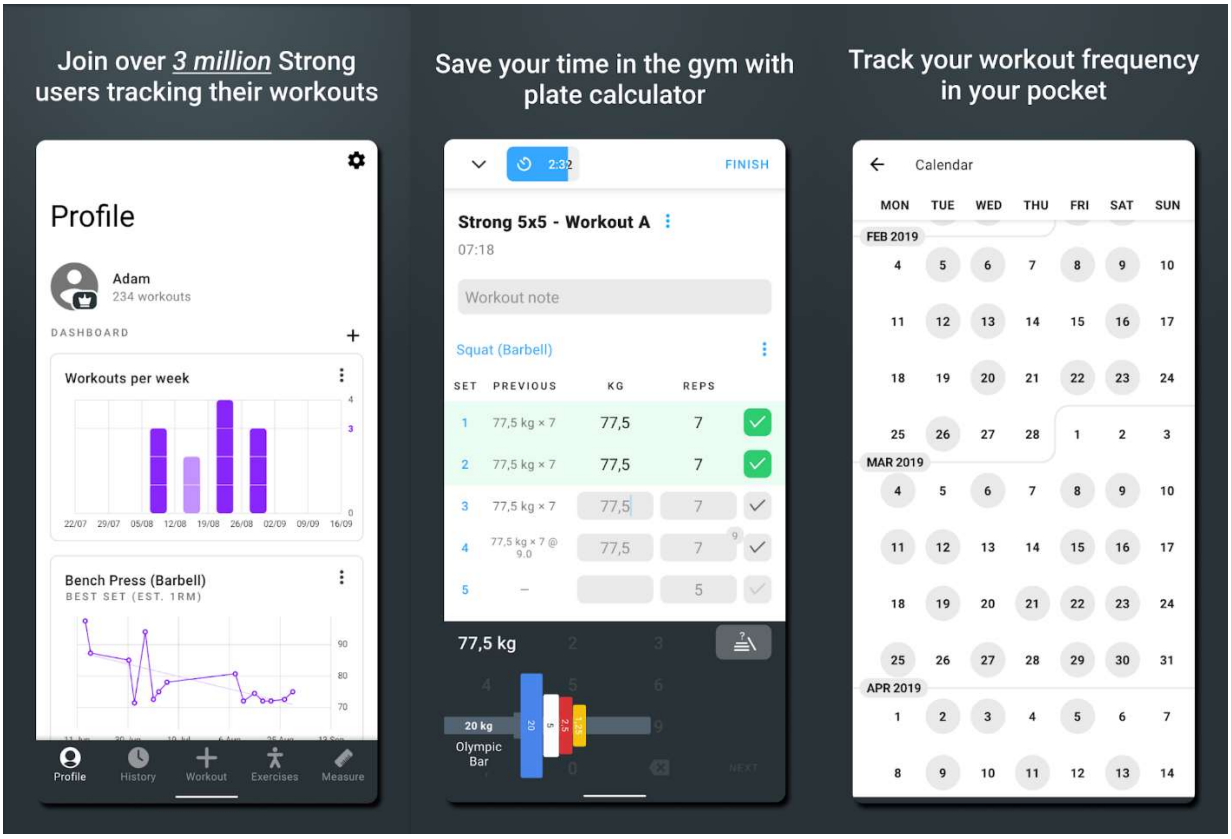
Prema [1] Fitness & Bodybuilding je jedna od najpopularnijih fitness aplikacija na Google Play-u. Prema dostupnim podacima preuzeta je preko 10 milijuna puta. Ocjena same aplikacije je 4.8/5, te je u ocjenjivanju sudjelovalo preko 194 000 korisnika. Aplikacija je dostupna samo na engleskom jeziku. Nadalje, najveća prednost Fitness & Bodybuilding aplikacije je ta što ima jako detaljno opisane i prikazane vježbe za pojedini dio tijela, što možemo vidjeti na slici 2.1. Također, za svaku vježbu daje mogućnost preuzimanja video zapisa koji prikazuje način izvođenja pokreta. Moguće je dodati težinu i broj ponavljanja, te nakon toga odabrano nam se sprema zajedno sa datumom kada je vježba obavljena.

Slika 2.1. Fitness & Bodybuilding – izgled korisničkog sučelja

(Preuzeto sa: https://play.google.com/store/apps/details?id=softin.my.fast.fitness&hl=hr&gl=US)

2.2. Strong Workout Tracker Gym Log

Prema [2] Strong Workout Tracker Gym Log je vrlo popularna fitnes aplikacija na Google Play-u. Prema podacima sa Google Play-a ima preko jednog milijuna preuzimanja. Ima ukupnu ocjenu 4.8 od 5, od preko 39 tisuća recenzija. Kao što se vidi na slici 2.2. ova aplikacija je puno više orjentirana ka praćenju statistika i podataka o vježbanju. Također, sadrži i vježbe za pojedini dio tijela koje su detaljno opisane.

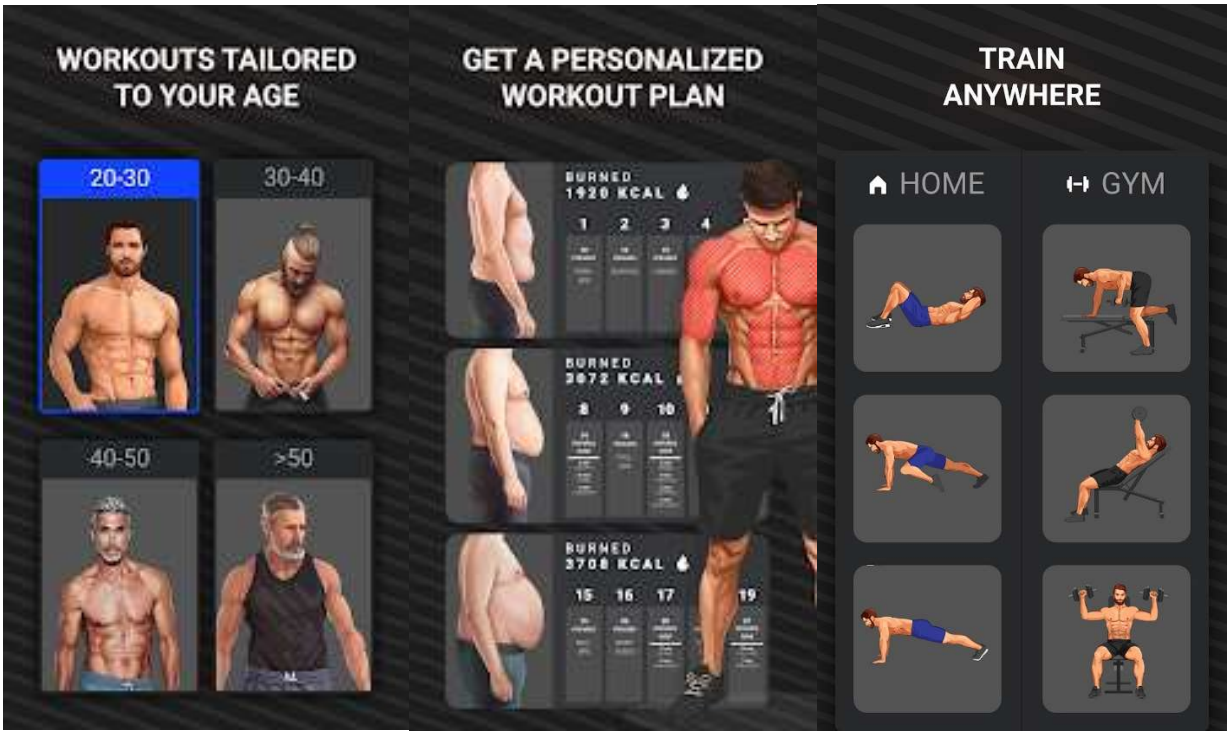


Slika 2.2. Strong Workout Tracker Gym Log – izgled korisničkog sučelja

(Preuzeto sa: https://play.google.com/store/apps/details?id=io.strongapp.strong&hl=hr&gl=US)

2.3. Muscle Booster Workout Planner

Pema [3] Muscle Booster Workout Planner je također jedna od najpopularnijih aplikacija na Google play-u, te je preuzeta preko 10 milijuna puta. Ocjena joj je 4.4 od 5, sa preko 143 tisuće recenzija. Aplikacija na osnovu odgovorenih pitanja pri prvom pokretanju kreira trening plan i racuna rok koji je potreban da se željeni cilj postigne. Međutim sam plan se plaća što može biti mana ove aplikacije.



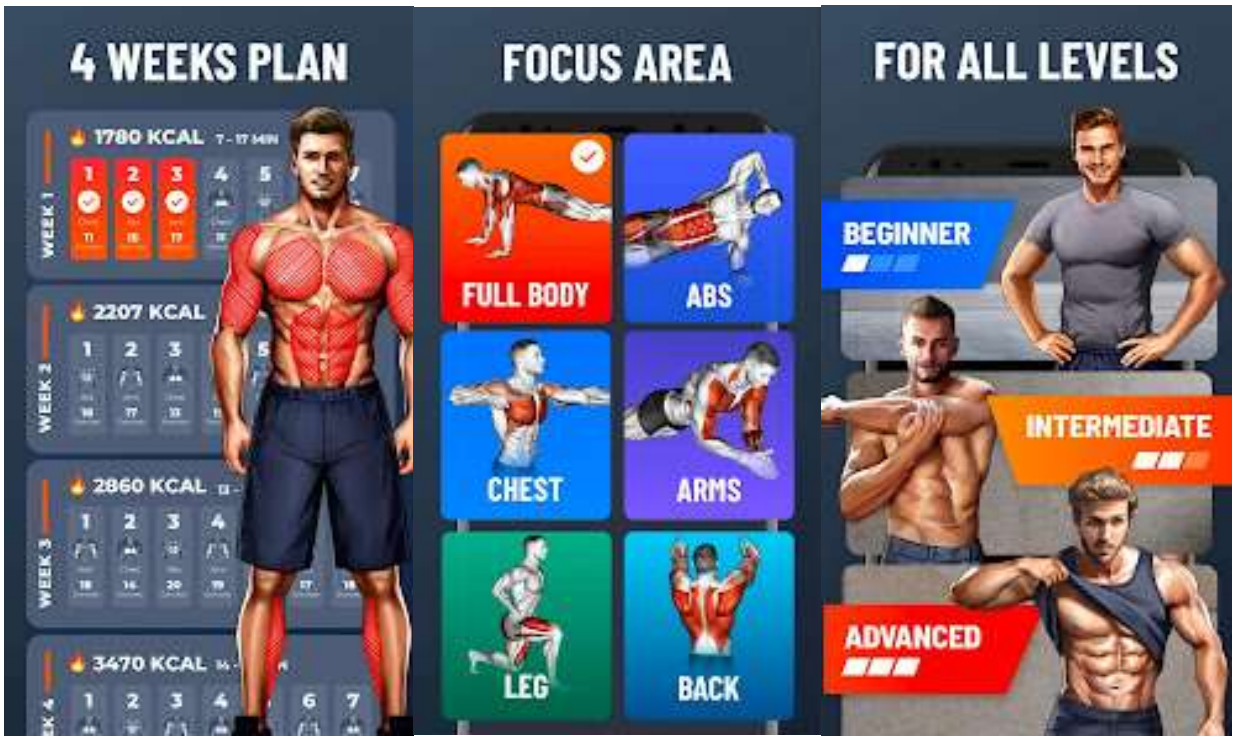
Slika 2.3. Muscle Booster Workout Planner – izgled korisničkog sučelja

(Preuzeto sa: https://play.google.com/store/apps/details?id=musclebooster.workout.home.gym.abs.loseweight&hl=hr&gl=US)

2.4. Vježbe kod kuće – bez opreme

Prema [4] Vježbe kod kuće – bez opreme je najpopularnija fitness aplikacija na Google Play-u. Prema prikazanim podacima ima preko 100 milijuna preuzimanja. Ocjena je 4.9 od mogućih 5, od preko 2.79 milijuna recenzija. Aplikacija je namjenjena za vježbanje kod kuće i samo sa bodyweight vježbama, ali sam koncept aplikacije je jako dobar te sadrži veliki broj vježbi. Najbolja stvar u vezi ove aplikacije je ta što ima jako veliki broj challenge-a koji potiču korisnike

na korištenje aplikacije i ostvarenje ciljeva. Također, aplikacije je prilagođena za sve korisnike, jer se kreira plan po korisnikovim odgovorima pri prvom paljenju aplikacije.

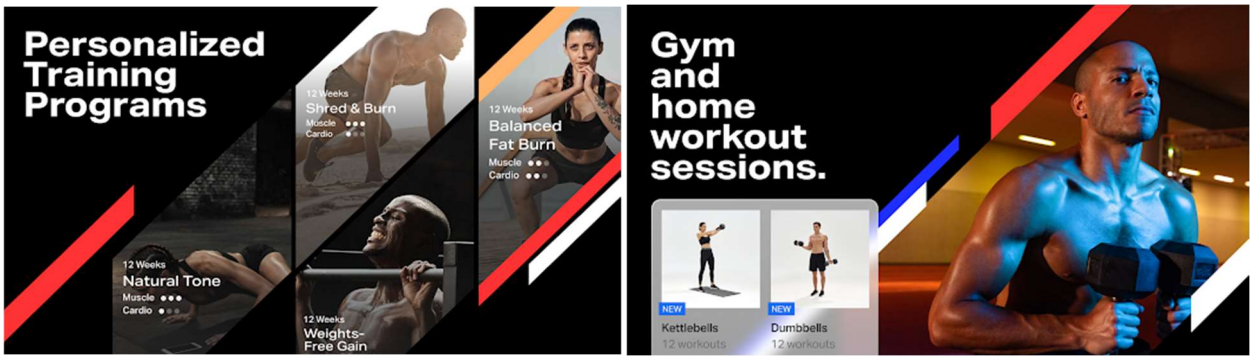


Slika 2.4. Vježbe bez kuće – bez opreme -izgled korisničkog sučelja

(Preuzeto sa: https://play.google.com/store/apps/details?id=homeworkout.homeworkouts.noequipment&hl=hr&gl=US)

2.5. Freeletics: Fitness Workouts

Prema [5] Freeletics: Fitness Workouts ima preko 10 milijuna preuzimanja. Ocjena joj je 4.4 od 5 sa preko 229 tisuća recenzija. Aplikacija ima veliki izbor vježbi i više načina vježbanja kao što je bodyweight, kettlebell i slično. Dio aplikacije gdje korisnik dobije plan treninga se plaća, međutim moguće je kreirati sam svoj trening ili pratiti određene challeng-e unutar aplikacije.



Slika 2.5. Freeletics: Fitness Workouts – izgled korisničkog sučelja

(Preuzeto sa: https://play.google.com/store/apps/details?id=com.freeletics.lite&hl=hr&gl=US)

3. TIJEK RAZVOJA APLIKACIJE

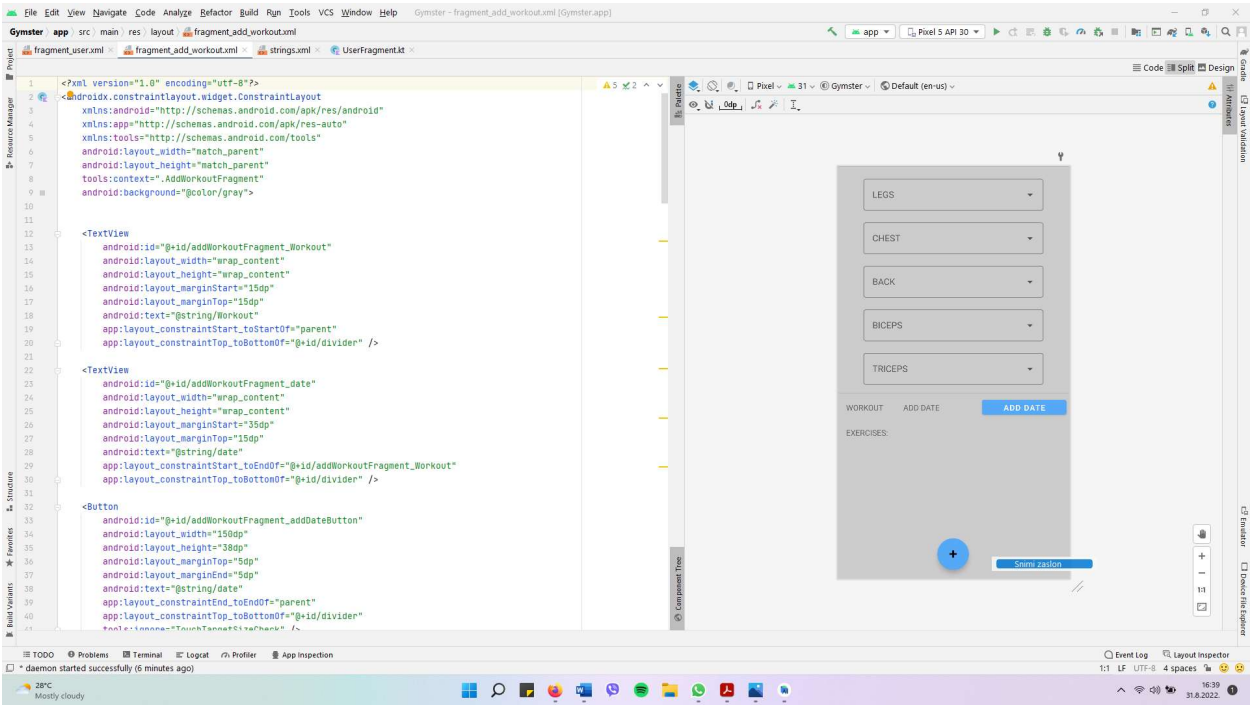
Cilj trećeg poglavlja je opisati tehnologije koje su korištene u razvoju same aplikacije. Također prikazati slike važnih dijelova koda te ostale alate korištene za ostvarenje funkcionalnosti aplikacije.

3.1. Korištene tehnologije

Aplikacija je u potpunosti razvijena u razvojnom okruženju Android Studio, korišten je programski jezi Kotlin, te XML opisni jezik za dizajn aplikacije. Za kreiranje baze podataka korištena je SQL baza podataka. Prethodno navedene tehnologije su ukratko opisane i pojašnjene u sljedećim potpoglavljima.

3.1.1. Android Studio

Prema [6] Android Studio je službeno integrirano razvojno okruženje (IDE, engl. Integrated Development Environment) za Googleov operacijski sustav Android. Samo okruženje je izgrađeno na JetBrains IntelliJ IDEA softveru i kao takvo dizajnirano je za razvoj isključivo aplikacija na operacijskom sustavu Android. Android Studio okruženje podržano je na četiri operacijska sustava: Windows, Linux, MacOS, te Chrome OS. Moguće je vršiti analize statičkog koda. U okruženju prevladavaju objektno orijentirani programski jezici. Prethodno je korišten objektno orijentirani programski jezik Java. Od 2019. godine programski jezik Kotlin postaje vodeći programski jezik za razvoj Android aplikacija.. Međutim, Java i programski jezik C++ su i dalje podržani. Na slici 3.1. je prikazan izgled korisničkog sučelja razvojnog okruženja Android Studio.



Slika 3.1. Prikaz sučelja Android Studia

3.1.2. Programski jezik Kotlin

Kotlin je objektno orijenirani programski jezik koji je prema [7] i [8] predstavljen u srpnju 2011. godine od strane JetBrains-a kao Project Kotlin. Međutim, 15. veljače 2016. godine objavljen je Kotlin 1.0 koji se smatra prvim službeno stabilnim izdanjem. Google je 2019. godine objavio da je programski jezik kotlin njegov preferirani jezik za programiranje Android aplikacija. Kotlin je dizajniran za potpunu interakciju sa Javom, te najbitnije značajke Kotlina su to što je intuitivan, te veliki dio ovog jezika je sličan onome što je već od prije poznato.

3.1.3. Opisni jezik XML

Prema [9] eXtensible Markup Language ili skraćeno XML programski je jezik standardiziran za označavanje podataka. Također sam jezik ima određena pravila koja služe za opisivanje dokumenata koji je u formatu čitljivom i čovjeku, ali i računalu. XML je jednostavan, generaliziran, te je upotrebljiv i preko interneta. Sintaksa XML-a je vrlo intuitivna i jako slična

sintaksi HTML-a, ali namjena im je različita. Na slici 3.2 je prikazan izgled XML koda u razvojnom okruženju Android Studio.

```
10
11
12      <TextView
13          android:id="@+id/addWorkoutFragment_Workout"
14          android:layout_width="wrap_content"
15          android:layout_height="wrap_content"
16          android:layout_marginStart="15dp"
17          android:layout_marginTop="15dp"
18          android:text="@string/Workout"
19          app:layout_constraintStart_toStartOf="parent"
20          app:layout_constraintTop_toBottomOf="@+id/divider" />
21
22      <TextView
23          android:id="@+id/addWorkoutFragment_date"
24          android:layout_width="wrap_content"
25          android:layout_height="wrap_content"
26          android:layout_marginStart="35dp"
27          android:layout_marginTop="15dp"
28          android:text="@string/date"
29          app:layout_constraintStart_toEndOf="@+id/addWorkoutFragment_Workout"
30          app:layout_constraintTop_toBottomOf="@+id/divider" />
31
```

Slika 3.2. Prikaz izgleda XML koda u Android Studiu

3.1.4. SQL baza podataka

SQL je jezik namjenjen za radom i upravljanjem baza podataka. Moguće ga je koristiti za stvaranje, uređivanje i pretraživanje podataka unutar SQL baze. Svaki složeniji projekt sadrži neki tip baze podataka te je ovdje korištena lokalna SQL baza podataka. Prikaz upravljanja tablicom i uređivanja iste prikazan je na slici 3.3.

```
lateinit var cursor:Cursor
companion object{
    const val DATABASE_VERSION = 1
    const val DATABASE_NAME = "USERS"
    const val TRAINING = "TRAINING"
}
override fun onCreate(db: SQLiteDatabase?) {
    db?.execSQL( sql: "CREATE TABLE USERSTable(_id integer primary key autoincrement,FULLNAME TEXT,HEIGHT INTEGER,WEIGHT DOUBLE)")
    db?.execSQL( sql: "CREATE TABLE TRAINING(_key integer primary key autoincrement,DATE TEXT,EXERCISES TEXT)")
    db?.execSQL( sql: "INSERT INTO USERSTable(FULLNAME,HEIGHT,WEIGHT) VALUES('Mato Zecevic','175','70')")
```

Slika 3.3. Prikaz kreiranja SQL baze podataka

3.2. Postupak razvoja aplikacije

U narednim potpoglavljima prikazan je sam postupak razvoja aplikacije, njezini najvažniji dijelovi koda popraćeni su slikom te opisom. Zbog složenosti aplikacije sastoji se od velikog broja klasa i više aktivnosti.

3.2.1. AndroidManifest datoteka

Svaki Android projekt sadrži *AndroidManifest.xml* datoteku. Ona predstavlja korijen projekta te prikazuje informacije i alate bitne samoj Google Play aplikaciji. Nadalje, datoteka brine o zaštiti aplikacije, odnosno u datoteku se upisuju određene dozvole koje pokreću određene akcije pri pokretanju aplikacije. Primjer dozvole može biti dozvola za pristup internetu koja je korištena u ovoj aplikaciji. Također, u ovoj datoteci se postavlja logo same aplikacije te se upravlja statusnom trakom. Na slici 3.4. je prikazan kod iz datoteke AndroidManifest.

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.gymster">

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

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="Gymster"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.Gymster">
        <activity
            android:name=".BeginnerActivity"
            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>
    </application>

</manifest>
```

Slika 3.4. Prikaz koda iz datoteke AndroidManifest

3.2.2. Aktivnost MainActivity

Ova aktivnost predstavlja prvu aktivnost koja se prikazuje prilikom ulaska u aplikaciju, što se također definira unutar *AndroidManifest* datoteke. U slučaju da korisnik prvi put ulazu u aplikaciju potrebno se registrirati na način da se unese ime te visina i težina. Međutim, ako je korisnik već prijavljen, potrebno je preskočiti početnu aktivnost jer više nema potrebe za registriranjem.

3.2.3. Gradle

Gradle je alat unutar Android projekta koji služi za automatizirano upravljanje procesa izgradnje te je sa Gradle alatom moguće definirati prilagođenu konfiguraciju potrebnu za izgradnju. Svaku konfiguraciju je potrebno implementirati unutar ovisnosti te se ona sastoji od seta koda i resursa. Sam kod i implementacija predstavljaju određene biblioteke potrebne za rad unutar projekta. Na slici 3.5. je prikazan kod sa implementacijama unutar projekta.

```
8
9       implementation 'androidx.core:core-ktx:1.7.0'
0       implementation 'androidx.appcompat:appcompat:1.4.1'
1       implementation 'com.google.android.material:material:1.5.0'
2       implementation 'androidx.constraintlayout:constraintlayout:2.1.3'
3       implementation 'androidx.appcompat:appcompat:1.2.0-alpha01'
4       implementation 'androidx.legacy:legacy-support-v4:1.0.0'
5       testImplementation 'junit:junit:4.+'
6       androidTestImplementation 'androidx.test.ext:junit:1.1.3'
7       androidTestImplementation 'androidx.test.espresso:espresso-core:3.4.0'
8
9       //meow navigation bar
0       implementation 'com.etebarian:meow-bottom-navigation:1.0.4'
1
2
3       implementation 'androidx.legacy:legacy-support-v4:1.0.0'
4
5       implementation("androidx.navigation:navigation-fragment-ktx:2.5.1")
6       implementation("androidx.navigation:navigation-ui-ktx:2.5.1")
```

Slika 3.5. Prikaz koda iz Gradle

3.2.4. RecylerView

RecyclerView je način prikazivanja podataka unutar aplikacije, najčešće je prikaz veće količine podataka prikazanih kao zasebni *item* jedan ispod drugog ili jedan pored drugog. Budući da se u aplikaciji RecyclerView koristi na tri mjesta važno ga je spoemuti te opisati njegovu funkcionalnost i implementaciju. U aplikaciji imamo dva načina na koji je korišten, odnosno prvi način je u ExerciseFragmentu gdje se podaci povlače iz *drawable* datoteke projekta. Unutar

drawable datoteke projekta nalaze se slike za pojedinu vježbu, a naziv pojedine vježbe nalazi se u polju stringova. Taj način prikazan je kodom na slici 3.5. i 3.6.

```
imageId= arrayOf(
    //noge
    R.drawable.bbl_back_squats,
    R.drawable.inverted_smith_machine_leg_press,
    R.drawable.single_leg_dumbell_squat,
    R.drawable.dumbell_lunge_step_up,
    R.drawable.seated_machine_squats,
    R.drawable.leg_extensions,
    R.drawable.barbell_hip_thrust,
    //triceps
    R.drawable.triceps_dips,
    R.drawable.triceps_dips_machine,
    R.drawable.cable_v_bar_pushdown,
    R.drawable.lyng_bbl_triceps_ext,
    //leđa
    R.drawable.wide_overhand_grip_lat,
    R.drawable.incline_bench_bbl_rows,
    R.drawable.bent_bar_cable_row,
    R.drawable.stand_cable_rope_push,
    R.drawable.hyperextensions,
    //prsa
```

Slika 3.5. Prikaz koda za povlačenje slika iz *drawable* datoteke

```
private fun getExerciseData() {
    for(i in imageId.indices){
        val exercise = Exercise(imageId[i],exerciseTitle[i])
        newArrayList.add(exercise)
    }
    newRecyclerView.adapter=ExerciseAdapter(newArrayList)

}
```

Slika 3.6. Prikaz koda za postavljanje imena i slike metodom *getExerciseData*

Kada se radi od RecyclerView-u potreban je i adapter koji će prilagoditi prikaz iz određene baze u prikaz na zaslonu, stoga imamo klasu sa metodama potrebnim za ostvarenje funkcionalnosti adaptera. Na slici 3.7 prikazan je kod ExerciseAdapter-a korištenog u aplikaciji.

```
class ExerciseAdapter (private val exercisesList : ArrayList<Exercise>):
    RecyclerView.Adapter<ExerciseAdapter.ViewHolder>() {

    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ViewHolder {
        val itemView = LayoutInflater.from(parent.context).inflate(R.layout.exercise_layout,parent, attachToRoot: false)
        return ViewHolder(itemView)
    }

    override fun onBindViewHolder(holder: ViewHolder, position: Int) {
        val currentItem = exercisesList[(position)]
        holder.titleImage.setImageResource(currentItem.titleImage)
        holder.exerciseName.text=currentItem.exerciseName
    }

    override fun getItemCount(): Int {
        return exercisesList.size
    }

    class ViewHolder(itemView : View) : RecyclerView.ViewHolder(itemView){
        val titleImage : ShapeableImageView = itemView.findViewById(R.id.title_image)
        val exerciseName :TextView = itemView.findViewById(R.id.workout)
    }
}
```

Slika 3.7. Prikaz koda ExerciseAdaptera

Drugi način korištenja RecyclerView-a u aplikaciji je povlačenje podataka iz baze podataka. U ovoj aplikaciji je korištena SQL baza podataka te na slici 3.8. prikazan je kod potreban za ostvarenje funkcionalnosti.

```
var rs = db.rawQuery( sql: "SELECT * FROM USERSTable", selectionArgs: null)
if(rs.moveToLast()){
    fullName.setText(rs.getString( columnIndex: 1))
    weight.setText(rs.getString( columnIndex: 3))
    height.setText(rs.getString( columnIndex: 2))
}
```

Slika 3.8. Kod za povlačenje podataka iz baze

Budući da je RecyclerView element potrebno ga je i opisati unutar XML koda, odnosno omogućeno je uređivati ga kao i sve ostale elemente AndroidStudio okruženja. Izgled RecyclerView-a u XML kodu prikazan je na slici 3.9.

```
12
13      <androidx.recyclerview.widget.RecyclerView
14          android:id="@+id/exercisesFragment_recyclerView"
15          android:layout_width="match_parent"
16          android:layout_height="wrap_content"
17          app:layout_constraintEnd_toEndOf="parent"
18          app:layout_constraintStart_toStartOf="parent"
19          app:layout_constraintTop_toTopOf="parent" />
20
```

Slika 3.9. XML kod RecyclerView-a

3.2.5. Registracija korisnika

Postupak registracije korisnika obavlja se unutar MainActivity-a ispunjavanjem ponuđenih polja, te pritiskom guma Save. Od korisnika se zatraži da unese podatke tipa string u prvo polje koje se odnose na nadimak ili puno ime i prezime, zatim u drugo polje podatak tipa double koji se odnosi na visinu te u treće polje podatak tipa double koji se odnosi na težinu. Za registraciju je potreban samo nadimak, međutim registracija je uspješna samo ako su sva 3 polja ispunjena, te se to potvrđuje pojavom *Toast* poruke na zaslonu. Budući da sama registracija ne igra veliku ulogu u ovoj aplikaciji nisu potrebne prijave kao što je email i lozinka , te se nakon unosa podataka isti spremaju u SQL bazu podataka. Isti podaci se povlače iz baze podataka u „UserFragment-u“ te je u istom omogućena promjena težine. Na slici ispod prikazan je dio koda sa unosom podataka u bazu, odnosno insert funkcija SQL-a. Na slici 3.5. je prikazan kod koji je korišten za registraciju korisnika odnosno unos podataka u bazu.

```
        var helper = MyHelper(applicationContext)
        var db:SQLiteDatabase = helper.readableDatabase
        var rs = db.rawQuery( sql: "SELECT * FROM USERSTable", selectionArgs: null)


        registered.setOnClickListener { SwitchActivity() }

        val saveButton = findViewById<Button>(R.id.mainActivity_saveButton)
        saveButton.setOnClickListener {  it: View!
            var cv = ContentValues()
            cv.put("FULLNAME",fullName.text.toString())
            cv.put("HEIGHT",height.text.toString())
            cv.put("WEIGHT",weight.text.toString())
            db.insert( table: "USERSTable", nullColumnHack: null,cv)
            rs.requery()
            SwitchActivity()
        }
    }
```

Slika 3.5. Prikaz koda za registraciju korisnika

3.2.6. Funkcionalnost datuma

Za spremanje i kreiranje plana treninga unaprijed potrebno je za svaki trening unijeti datum samog treninga, te je isto izvedeno pomoću AlertDialog-a koji se pojavljuje na pritisak gumba u samoj aplikaciji. Budući da u razvojnom okruženju imamo implementirani prikaz kalendara pomoću integrirane klase DatePickerDialog, istu klasu sam i koristio za ostvarenje funkcionalnosti datuma. Odabirom određenog datuma, isti se sprema u TextView unutar fragmenta te kasnije pritiskom na gumb i sprema u bazu podataka. Sam kod za ostvarenje funkcionalnosti datuma prikazan je na slici 3.6.

```
//////////calendar////////////////////////////////////////////////////////////////////////////////////////////////////
val date = view.findViewById<TextView>(R.id.addWorkoutFragment_date)
val chosenExercises=view.findViewById<TextView>(R.id.addWorkoutFragment_choosenExercises)

val c = Calendar.getInstance();
val year = c.get(Calendar.YEAR);
val month = c.get(Calendar.MONTH);
val day = c.get(Calendar.DAY_OF_MONTH);

val addWorkoutButton = view.findViewById<Button>(R.id.addWorkoutFragment_addDateButton);
addWorkoutButton.setOnClickListener {  it: View!
    val dpd = DatePickerDialog(requireActivity(),R.style.DatePicker,DatePickerDialog.OnDateSetListener
              { view, mYear, mMonth, mDay -> date.setText(""+mDay+"/"+mMonth+"/"+mYear)
    },year,month,day)
    dpd.show()
}
```

Slika 3.6. Prikaz koda za ostvarenje funkcionalnosti datuma

3.2.7. Odabir vježbi za određenu grupaciju

Glavna funkcionalnost aplikacije je samostalno kreiranje treninga i planiranje treninga unaprijed. Funkcionalnost kreiranja treninga je ostvarena pomoću 5 grupacija. Grupacije su noge, prsa, leđa, biceps i tricpes. Svaka od navedenih grupacija ima svoj izbornik te pritiskom na taj izbornik imamo AlertDialog sa vježbama za pojedinu grupaciju. Svaka vježba ima svoj CheckBox te se na taj način odabire određena vježba. U samom AlertDialogu imamo 2 gumba. Jedan je gumb je za spremanje odabranog, dok drugi služi za prekid radnje. Omogućeno je da korisnik odabere vježbe koje god on. Nakon spremanja odabranog vježbe se spremaju u TextView te kasnije u SQL bazu podataka.

3.2.8. Statistika treniranja

Jedna od bitnih funkcionalnosti aplikacije je funkcionalnost statistike obavljenih treninga. Statistika se vrši na način da se broje odrađene grupacije te se na osnovu ukupno odrađenih treninga računa postotak odrađenih treninga po pojedinoj grupaciji. Sam kod napisan za realizaciju statistike prikazan je na slikama 3.7. i 3.8.

```
cursor= db.rawQuery( sql: "SELECT * FROM TRAINING", selectionArgs: null)

if(cursor.moveToFirst()){
    do {
        workout= cursor.getString( columnIndex: 2)
        for (i in 0 until tricepsKeywords.size) {
            if (workout.contains(tricepsKeywords.get(i))) tricepsCounter++
        }
        for (i in tricepsKeywords.indices) {
            if (workout.contains(bicepsKeywords.get(i))) bicepsCounter++
        }
        for (i in tricepsKeywords.indices) {
            if (workout.contains(legsKeywords.get(i))) legsCounter++
        }
        for (i in tricepsKeywords.indices) {
            if (workout.contains(backKeyWords.get(i))) backCounter++
        }
        for (i in tricepsKeywords.indices) {
            if (workout.contains(chestKeywords.get(i))) chestCounter++
        }
    } while (cursor.moveToNext())}
```

Slika 3.7. Prikaz koda za ostvarenje funkcionalnosti *statistika*

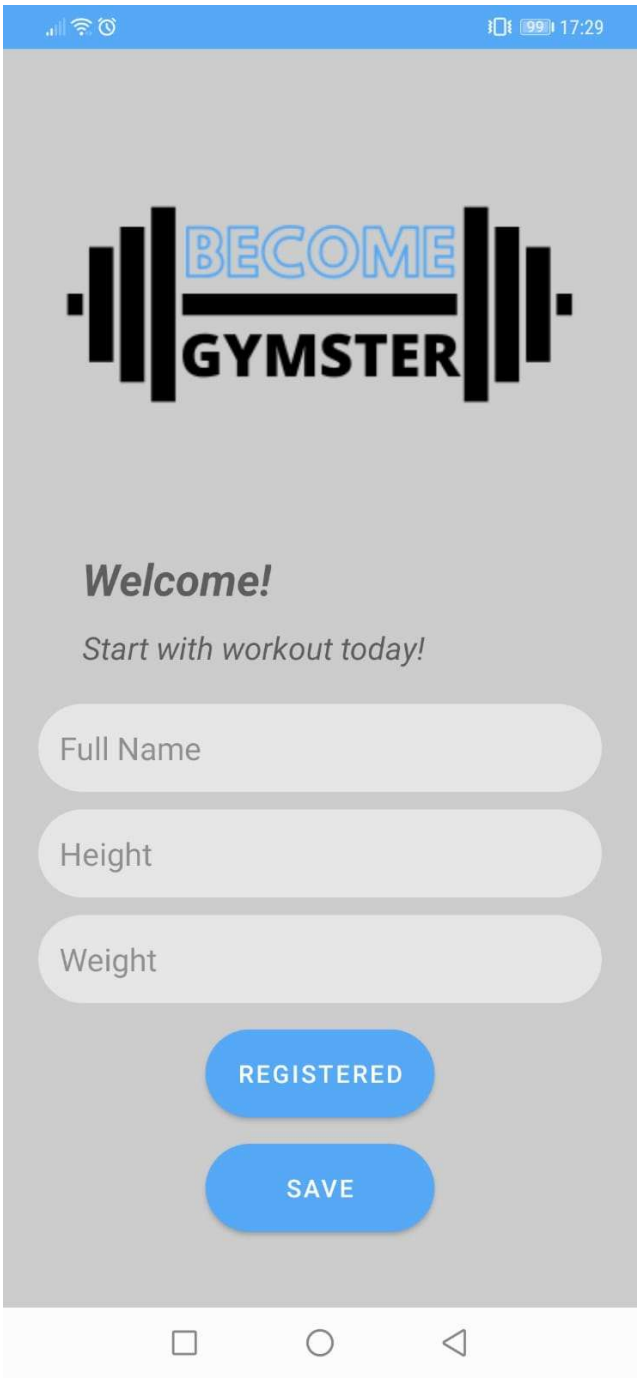
```
val suma = doubleArrayOf(tricepsCounter,bicepsCounter,legsCounter,backCounter,chestCounter)
val sum = suma.sum()

val tricepsStats=view.findViewById<TextView>(R.id.tricepsStats)
val bicepsStats=view.findViewById<TextView>(R.id.bicepsStats)
val backStats=view.findViewById<TextView>(R.id.backStats)
val chestStats=view.findViewById<TextView>(R.id.chestStats)
val legsStats=view.findViewById<TextView>(R.id.legsStats)


var triceps : Double

if(sum!=0.0){
    triceps = ((tricepsCounter/sum)*100).roundToInt().toDouble()
    tricepsStats.text= triceps.toString()+"%"

    val biceps =((bicepsCounter/sum)*100).roundToInt().toDouble()
    bicepsStats.text=biceps.toString()+"%"

    val back = ((backCounter/sum)*100).roundToInt().toDouble()
    backStats.text=back.toString()+"%"

    val legs = ((legsCounter/sum)*100).roundToInt().toDouble()
    legsStats.text=legs.toString()+"%"

    val chest = ((chestCounter/sum)*100).roundToInt().toDouble()
    chestStats.text=chest.toString()+"%"

} else {
    tricepsStats.text= "0%"
    bicepsStats.text="0%"
    chestStats.text="0%"
    legsStats.text="0%"
    backStats.text="0%"
}
```

Slika 3.8. Prikaz koda za ostvarenje funkcionalnosti *statistika*

4. STRUKTURA APLIKACIJE

4.1. MainActivity

MainActivity na samom vrhu prikazuje logo aplikacije, ispod njega se nalazi tekst koji poručuje dobrodošlicu korisniku te ispod teksta se nalaze tri polja za unos teksta. U prvi se upisuje *NickName*, u drugi *Height* te u treći *Weight.* Ispod polja nalazi se gumb *Save* te pritiskom na njega otvara se nova aktivnost. Svako polje ima hintove za onos, te ukoliko se ne unesu podaci u svako polje pojavljuje je „Toast“ poruka. Ispod polja za unos teksta nalaze se 2 gumba. Jedan gumb sprema uneseno i prebacuje nas na iduću aktivnost i PlannerFragment, dok drugi gumb se koristi u slučaju da ste već prijavljeni korisnik a iz nekog razloga vas aplikacija dovede na početak. Sam izgled početne aktivnosti prikazan je na slici 4.1.



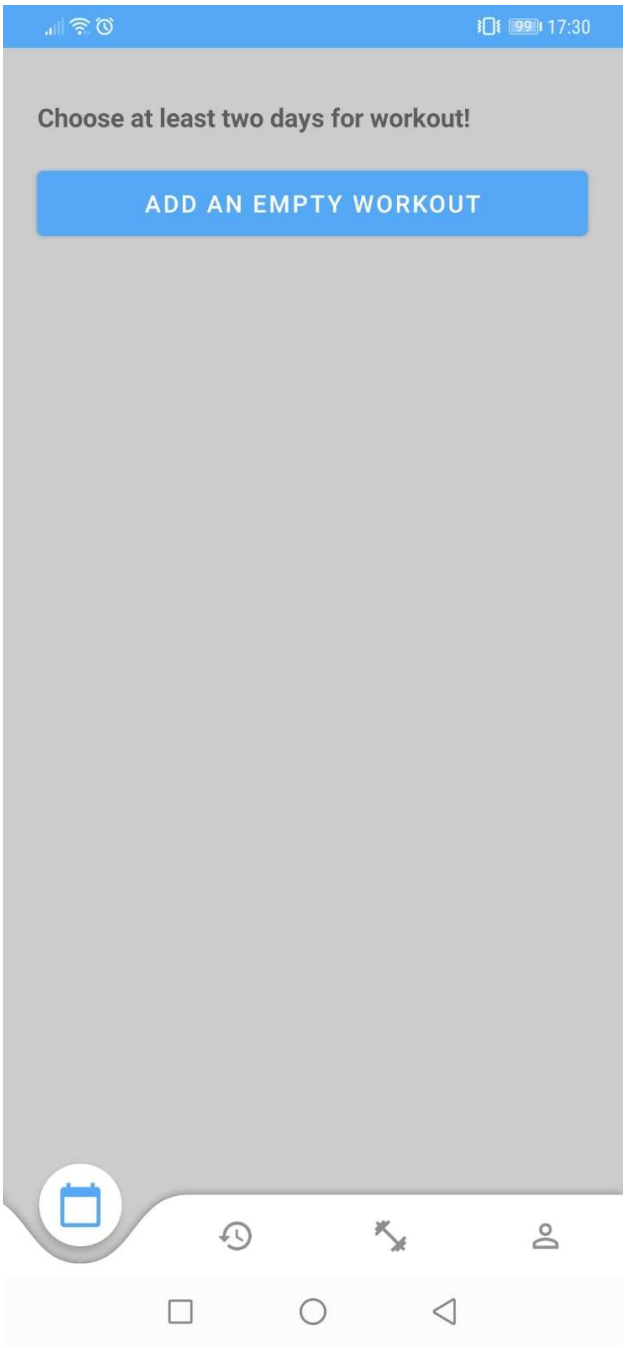
Slika 4.1. Prikaz početne aktivnosti

4.2. AppActivity

Druga aktivnost je AppActivity te ova aktivnost služi kao spremnik za fragmente koji su korišteni u projektu. U *spremniku* se nalaze 5 fragmenata: *PlannerFragment*, *AddWorkoutFragment, HistoryFragment, ExercisesFragment* te *UserFragment.*

4.2.1. PlannerFragment

Prvi fragment unutar AppActivity-a je *AddWorkoutFragment*. Na samom vrhu fragmenta nalazi se TextView u kojem se poručuje da se odabere najmanje 2 dana za trening. Neposredno ispod se nalazi gumb ADD AN EMPTY WORKOUT kojim se prelazi u novi fragment. Na dnu framgenta nalazi se navigacijska traka sa gumbima koji vode na ostala 3 glavna fragmenta. Na samoj sredini fragmenta nalazi se RecyclerView koji prikazuje kreirani trening. Na slici 4.2. se nalazi prikaz izgleda PlannerFragment-a.

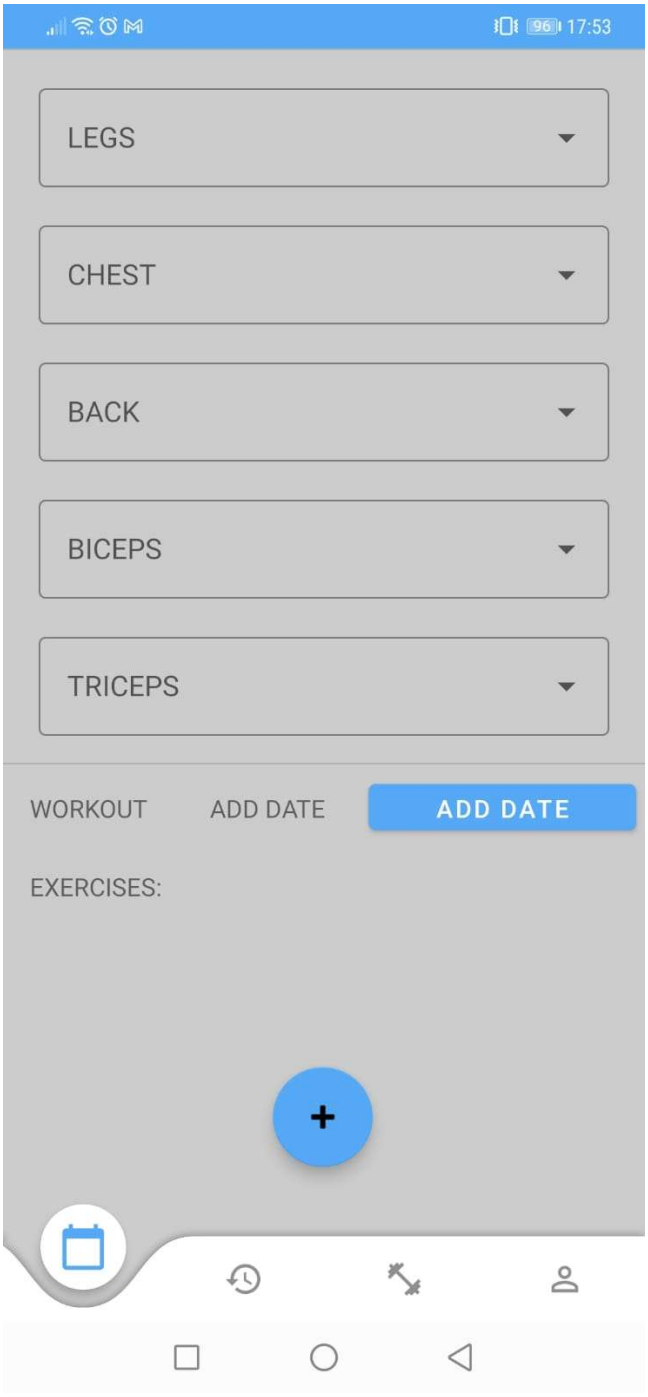


Slika 4.2. Prikaz PlannerFragment-a

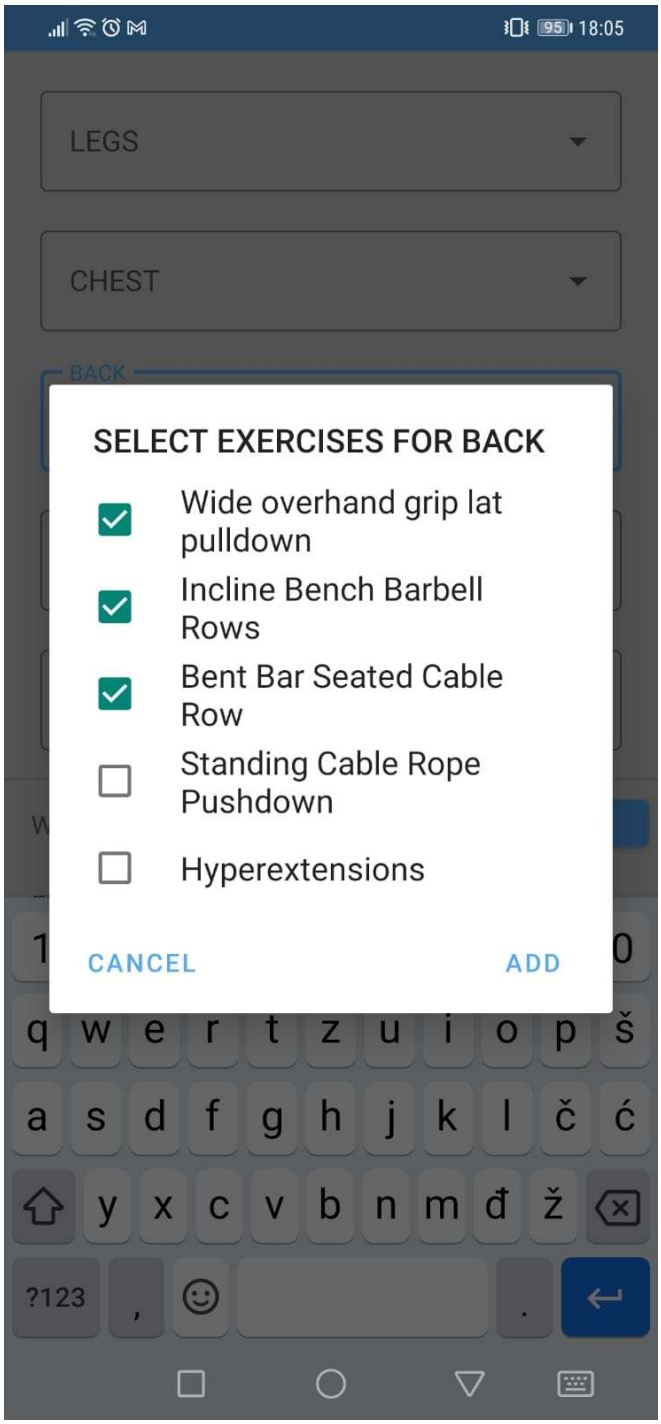
4.2.2. AddWorkoutFragment

AddWorkoutFragment je glavni fragment ove aplikacije, odnosno glavne funkcionalnosti aplikacije nalaze se u njemu te zbog toga ovom fragmentu treba posvetiti najviše pažnje. Na samom fragmentu imamo funkconalnosti za postavljanje datuma te funkcionalnosti za odabir željenih grupacija i vježbi za iste grupacije. Kao što je prikazano na slici 4.3. , svaka grupacija ima

svoj izbornik i klikom na određeni izbornik otvara se AlertDialog (slika 4.4.) sa „MultiChoice“ odabirom vježbi. Svaka vježba ima svoj CheckBox te se na taj način odabire pojedina vježba.

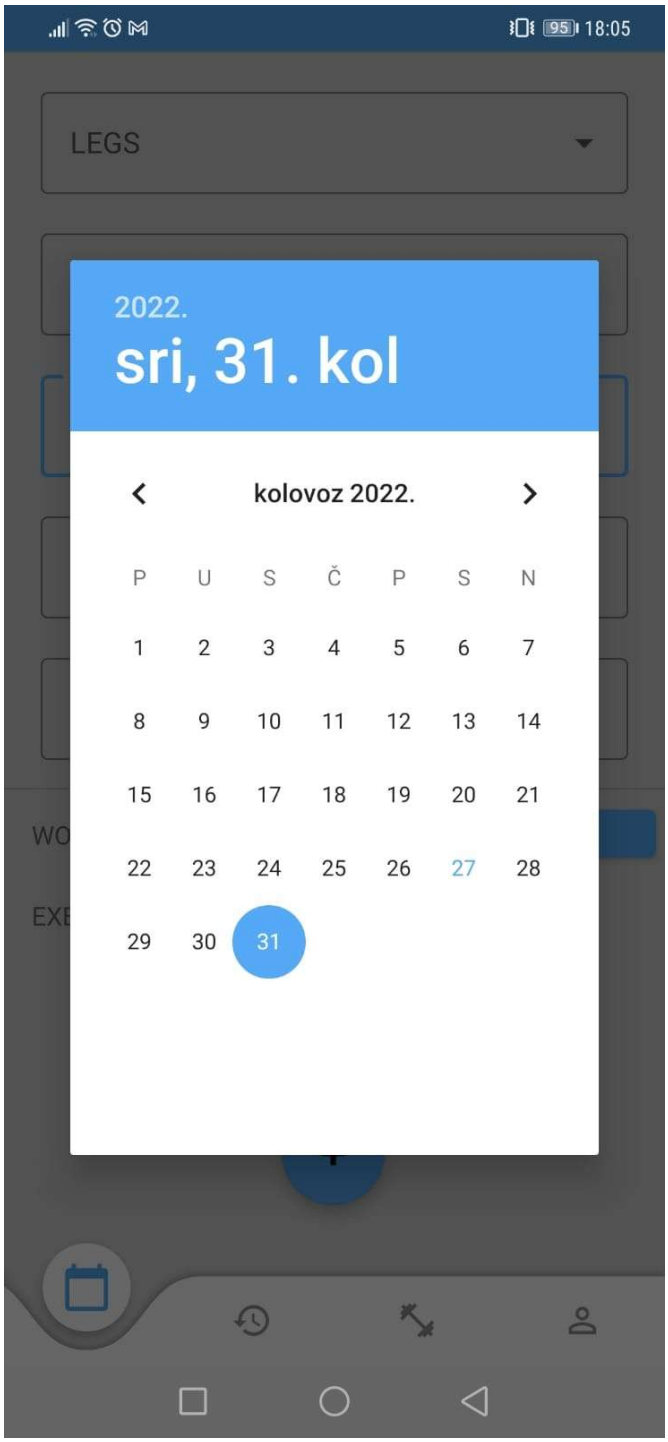


Slika 4.3. Prikaz AddWorkoutFragment-a

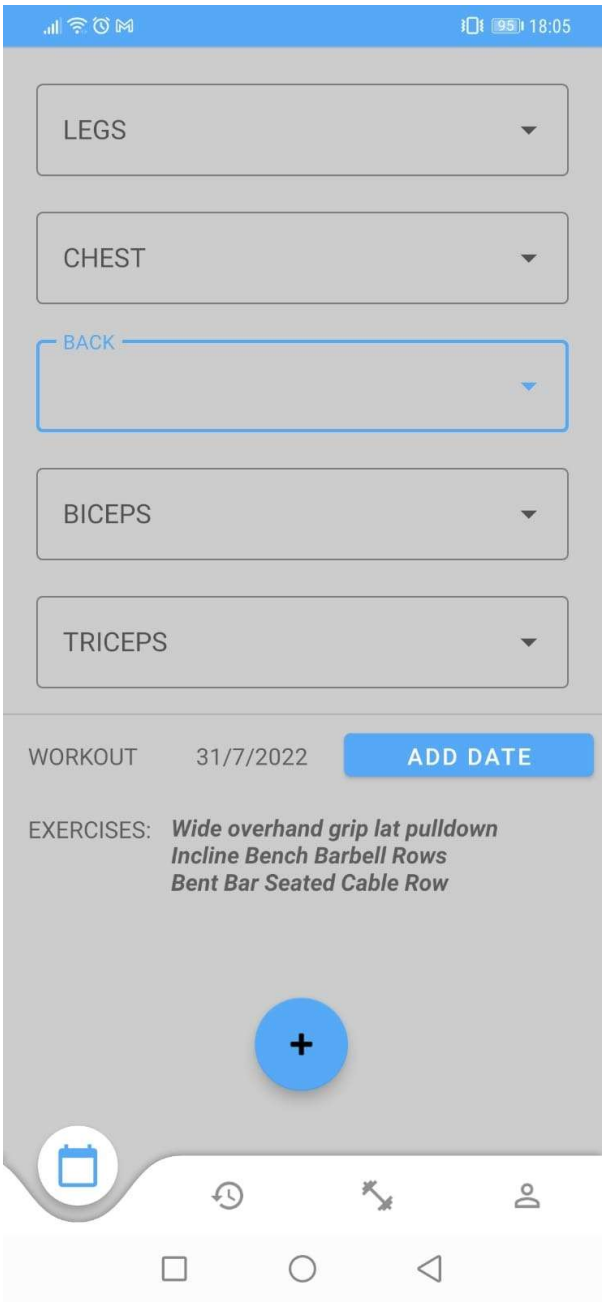


Slika 4.4. Prikaz izbornika za vježbe unutar AddWorkoutFragment-a

DatePicker se otvara pritiskom na gumb *ADD DATE* te izgled samog dialoga prikazan je na slici 4.5. U konačnici na slici 4.6. je prikazano kako se odabir datuma i odabir vježbi sprema u 2 TextView-a te pritiskom na FloatingButton na dnu fragmenta isti odabiri se spremaju u bazu podataka.



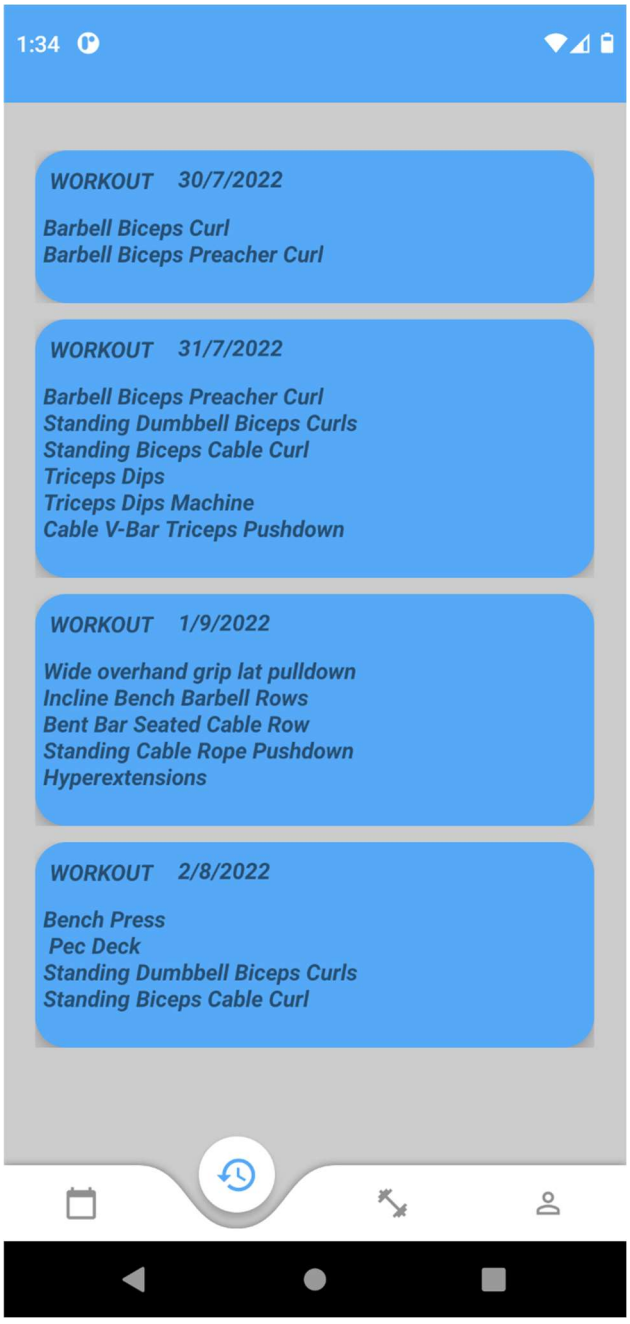
Slika 4.5. Prikaz dialog-a za datum unutar AddWorkoutFragment-a



Slika 4.6. Prikaz spremljenih odabira unutar AddWorkoutFragment-a

4.2.3. HistoryFragment

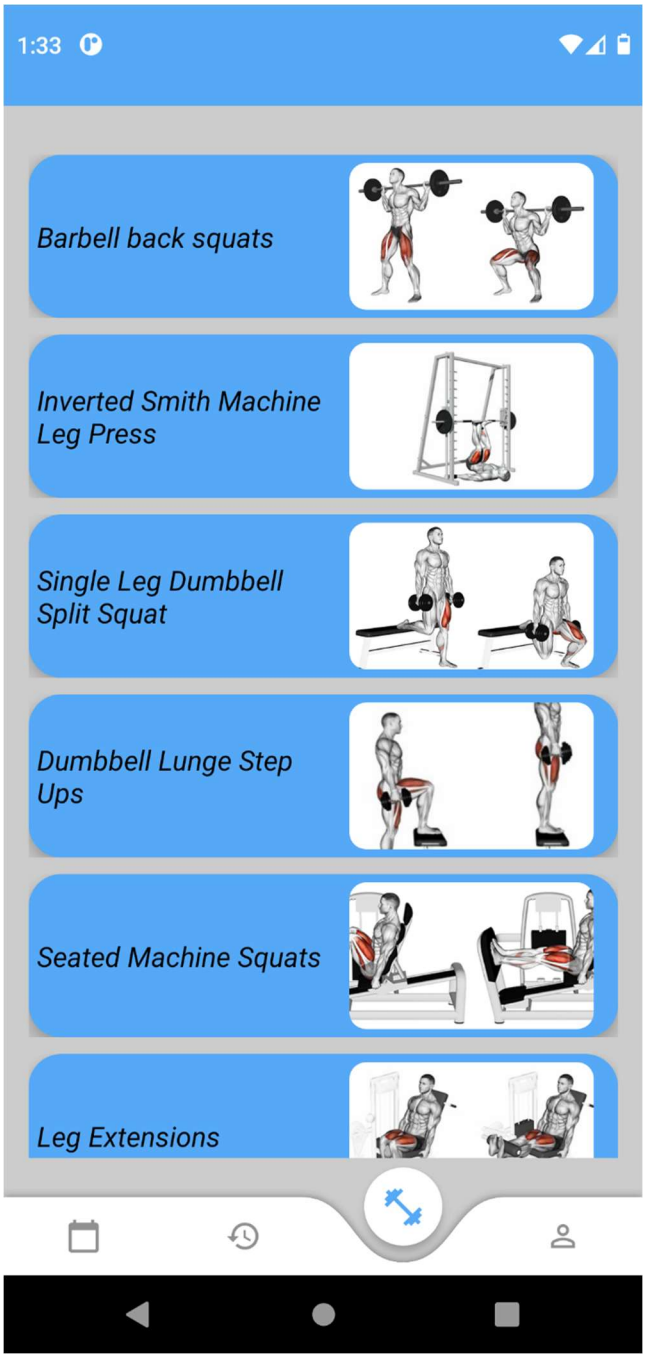
Treći fragment unutar AppActivity-a je *HistoryFragment.* Svrha ovog fragmenta je prikazati povijest svih obavljenih treninga. Navedeno je postignuto pomoću SQL baze podataka te RecyclerViewa u kojem su *povućeni* podaci spremljeni u SQL bazu podataka. Na slici 4.7 je prikazan izgled HistoryFragmenta, te na dnu možemo vidjeti navigacijsku traku sa ikonicom fragmenta.



Slika 4.7. Prikaz povućenih podataka iz baze unutar HistoryFragmenta

4.2.4. ExercisesFragment

Četvrti fragment unutar AppActivity-a je *ExercisesFragment.* Svrha ovog fragmenta je prikazati listu svih vježbi koje su ponuđene u aplikaciji, te su vježbe popraćene njihovim nazivom i slikom koja prikazuje način izvođenja. Navedeno je ostvareno pomoću RecyclerView-a, odnosno svaka vježba je jedan *item* unutar RecyclerView-a.

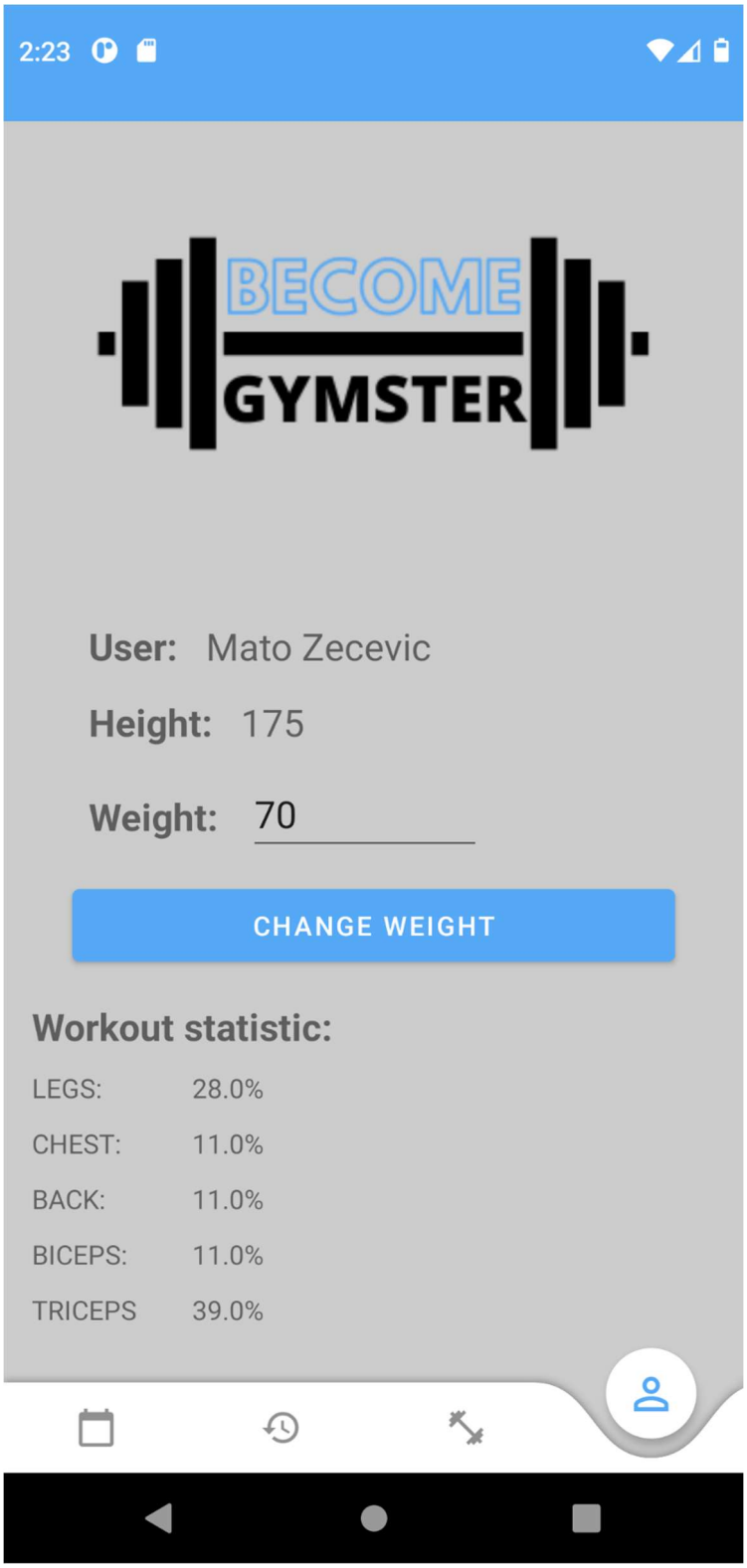


Slika 4.8. Prikaz ExercisesFramgent-a

Na dnu fragmenta nalazi se i navigacijska traka koja prikazuje da se nalazimo u ExerciseFragmentu te je ikonica odabrana kako bi bilo što intuitivnije za korisnika. Na slici 4.8. je prikazan izgleda samog fragmenta sa nazivom i slikom izvođenja pojedine vježbe.

4.2.5. UserFragment

Četvrti fragment unutar AppActivity-a je *UserFragment*. Svrha ovog fragmenta je prikazati unesene podatke pri samoj registraciji, te omogućiti promjenu težine. Promjena težine mora biti spremljena u SQL bazu i to na način da postojeća težina bude promijenjena, odnosno pomoću „*update*“ funkcije izmjenjena. Ovo je postignuto sa dva TextView-a i jednim EditText-om za težinu te gumbom za promjenu težine, odnosno spremanje „update-ane“ težine. Ispod toga nalazi se statistika obavljenih vježbi, te na osnovu statistike korisnik može zaključiti koliko određenu grupaciju trenira i što bi trebao popraviti u samom treningu. Na dnu se nalazi navigacijska traka sa intuitivnom ikonicom koja prikazuje da se radi o UserFragmentu. Izgled samog fragmenta prikazan je na slici 4.8.



Slika 4.9. Prikaz UserFramgent-a

5. ZAKLJUČAK

U današnje vrijeme aplikacije orijentirane praćenju napretka u teretani su postale jako često korištene. Kako bi ostvarivali i pratili napredak u teretani i sportu više ne moramo plaćati privatne trenere nego je moguće preuzeti aplikacije slične ovoj i pratiti svoj napredak. Aplikacija je vrlo intuitivna i prilagođena je tako da je mogu koristiti svi uzrasti.

Cilj završnog rada je bila realizacija aplikacije za praćenje napretka u treningu i samostalno kreiranje treninga namjenjene za korištenje na operacijskom sustavu Android. Aplikacija je razvijena u razvojnom okruženju Android Studio. Korišten je programski jezik Kotlin te opisni jezik XML za dizajn same aplikacije.Za pohranu i upravljanje podacima korištena je lokalna SQL baza podataka u koju su unošeni podaci o korisniku i podaci o treninzima. Tokom razvijanja aplikacije susreo sam se sa raznim poteškoćama zbog manjka poznavanja koncepata za razvoj ovog projekta.

Sama aplikacija ima još dosta prostora za nadogradnju i poboljšanje funkcionalnosti. Neke od mogućih nadogradnji bi bile detaljniji opisi vježbi, dodavanje štoperice, dodavanje broja ponavljanja, dodavanje već gotovih programa treniranja te prilagođavanje njih svojim potrebama, dodavanje određenih ciljeva kojim bi se korisnik potaknuo na obavljanje treninga i slično.

LITERATURA

[1] Fitness & Bodybuilding, Trgovina Google Play, dostupno na:

https://play.google.com/store/apps/details?id=softin.my.fast.fitness&hl=hr&gl=US

(Datum pristupa web sadržaju: 28.6.2021.)

[2] Strong Workout Tracker Gym Log, Trgovina Google Play, dostupno na:

https://play.google.com/store/apps/details?id=io.strongapp.strong&hl=hr&gl=US

(Datum pristupa web sadržaju: 28.6.2021.)

[3] Muscle Booster Workout Planner, Trgovina Google Play, dostupno na:

https://play.google.com/store/apps/details?id=musclebooster.workout.home.gym.abs.loseweight&hl=hr&gl=US

(Datum pristupa web sadržaju: 28.6.2021.)

[4] Vježbe kod kuće – bez opreme, Trgovina Google Play, dostupno na:

https://play.google.com/store/apps/details?id=homeworkout.homeworkouts.noequipment&hl=hr&gl=US

(Datum pristupa web sadržaju: 28.6.2021.)

[5] Freeletics: Fitness Workouts, Trgovina Google Play, dostupno na:

https://play.google.com/store/apps/details?id=com.freeletics.lite&hl=hr&gl=US

(Datum pristupa web sadržaju: 28.6.2021.)

[6] T. Norbye, X. Ducrohet, K. Chou, Android Studio: An IDE built for Android, dostupno na:

https://android-developers.googleblog.com/2013/05/android-studio-ide-built-for-android.html

(Datum pristupa web sadržaju: 8.7.2021.)

[7] Službena stranica, Kotlin, dostupno na:

https://kotlinlang.org/

(Datum pristupa web sadržaju: 11.7.2021.)

[8] Wikipedia, Kotlin (programming language), dostupno na:

https://en.wikipedia.org/wiki/Kotlin_(programming_language)

(Datum pristupa web sadržaju: 11.7.2021.)

[9] Liam R. E. Quin, XML Essentials, dostupno na:

https://www.w3.org/standards/xml/core

(Datum pristupa web sadržaju: 11.7.2021.)

SAŽETAK

U ovom završnom radu prikazan je razvoj Android aplikacije za praćenje napretka u treningu i samostalno kreiranje treninga. Aplikacija ima mogućnost kreiranja treninga s velikim izborom vježbi za sve grupacije mišića tijela također aplikacija na osnovu odrađenih treninga vrši proračun statistike te na osnovu nje korisnik može zaključiti koje grupacije treninga treba poboljšati i težiti određenoj jednakosti ili se prilagoditi proporcijama svoga tijela. Aplikacija je u potpunosti kreirana u Android Studio okruženju, dizajnirana je u XML opisnom jeziku te programirana u programskom jeziku Kotlin. Korištena je lokalna SQL baza podataka za unos i pohranu potrebnih podataka. Nakon uvoda prikazano je pet sličnih rješenja koja se mogu usporediti s rješenjem unutar ove aplikacije također kroz rad su opisane i glavne funkcionalnosti aplikacije popraćene s dijelom koda.

Ključne riječi: Android,Android Studio, Kotlin, sport, trening, teretana, XML

ABSTRACT

Android application for creating a weekly training plan

This final paper shows the development of an Android application for tracking improvement in training and for creating the training schedule on your own. Users of the application can create training plans with a large selection of exercises for major muscle groups of the body. Also, the application calculates statistics based on performed training, and by taking that into consideration, users can find out which groups of training should be improved, strive for a balanced schedule, or adapt to the proportions of the body. The application is completely created in the Android Studio environment, and it is designed in the XML markup language and programmed in the Kotlin programming language. A local SQL database was used for entering and storing the necessary data. After the introduction, five similar solutions are shown that can be compared with the solution within this application. Also, the main functionalities of the application are described throughout the paper, accompanied by a part of the code.

Keywords: Android, Android Studio, gym, Kotlin, sport, training, XML

ŽIVOTOPIS

Mato Zečević rođen je 19.01.2001. u gradu Vinkovci na istoku Hrvatske. U selu Cerna pohađao je Osnovnu školu Matija Antun Reljković. Tijekom svog osnovnoškolskog obrazovanja sudjeluje na školsim i županijskim natjecanjima iz predmeta matematika i fizika. Nakon završetka osnovne škole upisuje Tehničku školu Ruđera Boškovića Vinkovci smjer Tehničar za mehatroniku te 2019. završava svoje srednjoškolsko školovanje. 2019. godine upisuje preddiplomski sveučilišni studij Računarstvo na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija u Osijeku.

_____

Potpis