

Android aplikacija za praćenje učinka u teretani

Forjan, Filip

Undergraduate thesis / Završni rad

2020

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

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:200:661270>

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

Download date / Datum preuzimanja: **2024-11-19**

Repository / Repozitorij:

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



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

Sveučilišni preddiplomski studij računarstva

**ANDROID APLIKACIJA ZA PRAĆENJE UČINKA U
TERETANI**

Završni rad

Filip Forjan

Osijek, 2020.

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

Osijek, 08.09.2020.

Odboru za završne i diplomske ispite

Prijedlog ocjene završnog rada na preddiplomskom sveučilišnom studiju

Ime i prezime studenta:	Filip Forjan
Studij, smjer:	Preddiplomski sveučilišni studij Računarstvo
Mat. br. studenta, godina upisa:	R4059, 25.09.2019.
OIB studenta:	50832823967
Mentor:	Izv. prof. dr. sc. Ivica Lukić
Sumentor:	
Sumentor iz tvrtke:	
Naslov završnog rada:	Android aplikacija za praćenje učinka u teretani
Znanstvena grana rada:	Informacijski sustavi (zn. polje računarstvo)
Predložena ocjena završnog rada:	Izvrstan (5)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 2 bod/boda Jasnoća pismenog izražavanja: 3 bod/boda Razina samostalnosti: 3 razina
Datum prijedloga ocjene mentora:	08.09.2020.
Datum potvrde ocjene Odbora:	11.09.2020.
Potpis mentora za predaju konačne verzije rada u Studentsku službu pri završetku studija:	Potpis:
	Datum:



FERIT

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

IZJAVA O ORIGINALNOSTI RADA

Osijek, 16.09.2020.

Ime i prezime studenta:

Filip Forjan

Studij:

Preddiplomski sveučilišni studij Računarstvo

Mat. br. studenta, godina upisa:

R4059, 25.09.2019.

Turnitin podudaranje [%]:

7

Ovom izjavom izjavljujem da je rad pod nazivom: **Android aplikacija za praćenje učinka u teretani**

izrađen pod vodstvom mentora Izv. prof. dr. sc. Ivica Lukić

i sumentora

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

Potpis studenta:

SADRŽAJ

1. Uvod.....	1
1.1. Zadatak završnog rada	2
2. Usporedba s postojećim rješenjima	3
2.1. Prikaz postojećih rješenja za praćenje učinka	3
2.1.1. <i>Strong – Workout Tracker Gym Log</i>	3
2.1.2. <i>GymRun – Workout Log & Fitness Tracker</i>	4
2.1.3. <i>RepCount -Gym Log</i>	5
2.2. Vlastito rješenje za praćenje učinka	6
3. Opis korištenih tehnologija	7
3.1. Android.....	7
3.2. Visual Studio Code.....	7
3.3. Flutter.....	8
3.4. SQL.....	9
3.5. Dart	9
4. Razvoj aplikacije.....	10
4.1. Koncept aplikacije	10
4.2. Implementacija aplikacije.....	10
5. Korištenje aplikacijom	23
6. Zaključak.....	30
Literatura	31
Sažetak	33
Abstract	34
Životopis.....	35
Prilozi	36

1. UVOD

Tema ovog završnog rada je praćenje učinka u teretani. Ovom mobilnom Android aplikacijom omogućen je unos osobnih podataka korisnika poput imena, prezimena, visine i tjelesne težine i praćenje napretka u tjelesnim aktivnostima kod vježbi s tegovima. Aplikacija računa korisnikov indeks tjelesne mase (engl. *body mass index* – BMI). Izračunavanjem indeksa tjelesne mase korisnik dobiva okvirnu informaciju o vlastitoj uhranjenosti jer index tjelesne mase daje relativno točnu informaciju o količini prekomjernog masnoga tkiva kod korisnika. Jedna od primjena aplikacije je dobivanje uvida u promjene kod njegove tjelesne težine. Druga primjena i mogućnost aplikacije je unošenje informacija vezanih uz same vježbe koje korisnik izvodi. Aplikacija korisniku omogućuje odabir pojedinih vježbi te upisivanje maksimalne težine s kojom korisnik može pravilno izvesti tu vježbu te broj ponavljanja koliko je puta vježba izvedena s tom težinom. Aktivnim korištenjem korisnik dobiva uvid u napredak kod svake vježbe pojedinačno. Osim samog unošenja brojki korisnik ima mogućnost priložiti video izvođenja vježbi. Tjelovježba je vrlo bitna u svakodnevnom životu kako bi osoba održala ne samo zdravo tijelo već i zdrav duh. Glavni problem dolazi kod održavanja motivacije i kontinuiranog pristupa tjelovježbi. Korištenjem ove aplikacije korisnik će postepeno imati mogućnost praćenja svog učinka što će korisnika konstantno držati motiviranim. Upisivanjem i praćenjem podataka korisnik će stvoriti naviku vježbanja i nakon nekog perioda dobiti uvid u njegov napredak koji ostavlja osjećaj zadovoljstva i sreće te dodatno motivira.

Drugo poglavlje daje uvid u trenutno stanje sličnih aplikacija ovoj i govori o sličnostima i razlikama te prednostima i nedostacima ove aplikacije u usporedbi s postojećim na tržištu.

Treće poglavlje daje uvid u tehnologije korištene za izradu mobilne aplikacije. Teorijski se objašnjava Android operacijski sustav, njegova povijest i osnovna funkcionalnost. Također se opisuju glavne korištene komponente razvoja Android mobilne aplikacije odnosno Flutter. Ukratko se opisuju korišteni programski jezici korišteni u razvitku aplikacije, a to su Dart i SQL (engl. *Structured Query Language*).

Četvrto poglavlje daje točne informacije o razvitku mobilne aplikacije. Prikazuje nam početnu vizualizaciju dizajna aplikacije te postupnu realizaciju čitavog korisničkog sučelja i prikazana je potpuna implementacija koda.

Peto poglavlje objašnjava funkcionalnost aplikacije na konkretnim primjerima, odnosno prikazuje sve mogućnosti aplikacije i koje sve korisničke zahtjeve aplikacija ispunjava i također detaljizira i objašnjava načine korištenja aplikacijom .

1.1. Zadatak završnog rada

Aplikacije omogućuje korisnicima unos podataka o napravljenim vježbama i praćenje učinka tijekom vremena. Omogućuje dodavanje slika i videa koji prikazuju faze treninga. Također omogućuje praćenje statistike uz pojedine vježbe.

2. USPOREDBA S POSTOJEĆIM RJEŠENJIMA

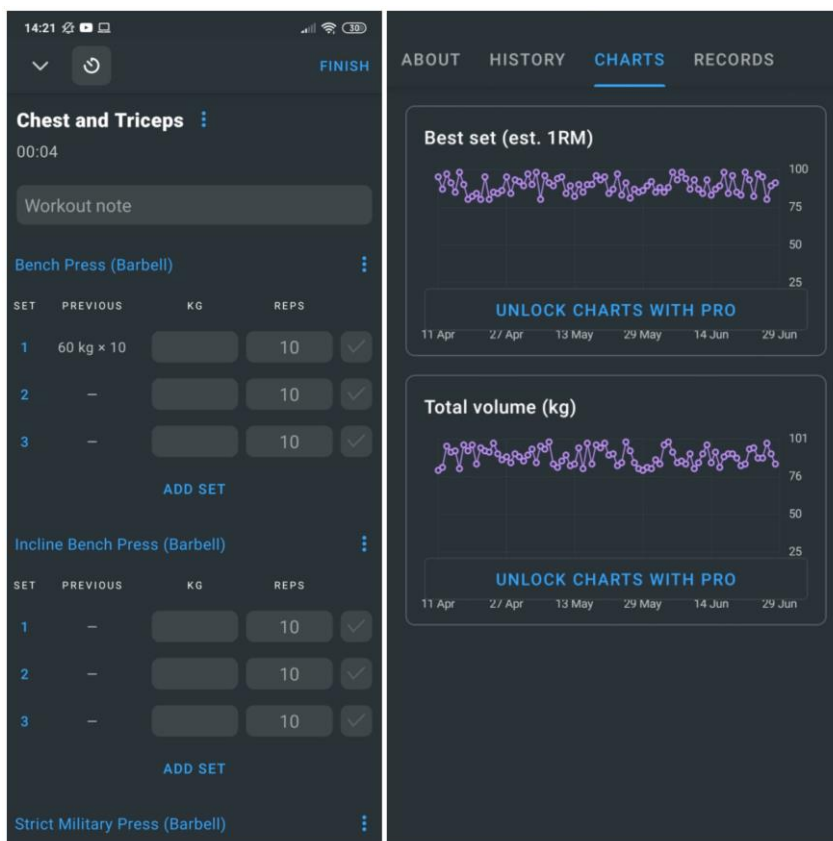
Postoje razna rješenja slična ovome, ali niti jedno ne obuhvaća i ne implementira svu funkcionalnost kao ovaj završni rad. Postoji raznoliki izbor aplikacija usmjerenih prema tjeleovježbi, ali većina se bavi ili praćenjem unesenih kalorija ili sastavljanje prikladnog plana za tjeleovježbu što pravi razliku od aplikacije ovog završnog rada.

2.1. Prikaz postojećih rješenja za praćenje učinka

Postoje brojne aplikacije zadužene za praćenje učinka u teretani, ali na tržištu niti jedna aplikacija nema mogućnost praćenja učinka uz mogućnost dodavanja fotografija ili video zapisa pomoću kojih se omogućuje praćenje promjena fizičkog izgleda korisnika.

2.1.1. *Strong – Workout Tracker Gym Log*

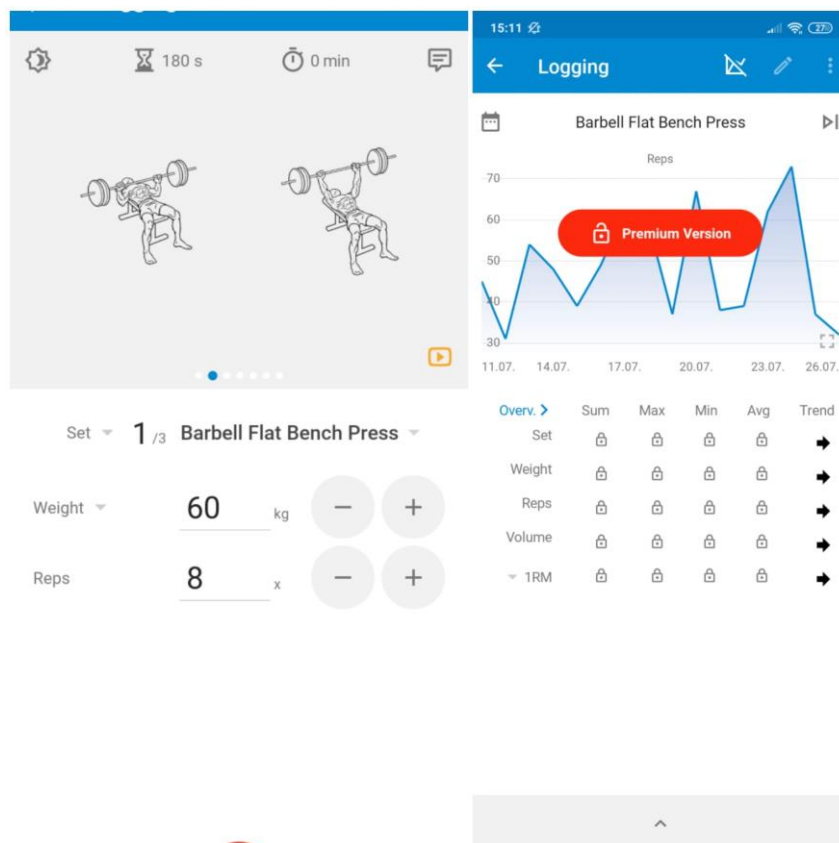
Strong je mobilna aplikacija koja omogućuje unošenje obavljenog treninga i praćenje učinka kroz određeni period vremena. Aplikacija prati broj odrađenih tjeleovježbi u tjednu, napretke u smanjenoj ili povećanoj tjelesnoj težni, postotak masnog tkiva. Pruža uvid u grafove koji prikazuju napredak te omogućava unošenje plana treninga unutar kojeg se treba definirati broj ponavljanja odrađene vježbe i težina s kojom je vježba rađena.



Slika 2.1. Sučelje aplikacije Strong

2.1.2. *GymRun – Workout Log & Fitness Tracker*

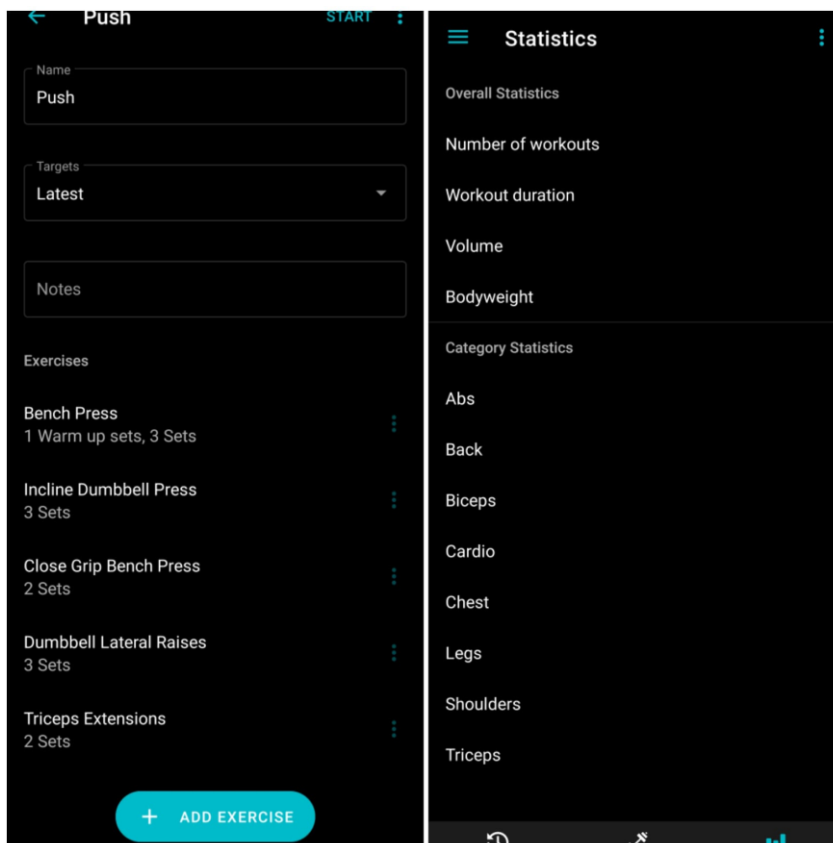
GymRun je mobilna aplikacija unutar koje korisnik sastavlja sebi prilagođen program za tjelovježbu. Aplikacija bilježi informacije o korisniku i odrađenim tjelovježbama i skupljanjem informacija pruža uvid u progresivni napredak. Aplikacija ujedno pruža bilježenje omjera dijelova tijela i pruža podršku za pametne satove.



Slika 2.2. Sučelje aplikacije GymRun

2.1.3. RepCount -Gym Log

RepCount je mobilna aplikacija koja omogućuje brz i jednostavan unos podataka o odrađenoj tjelovježbi. Uz postavljene štoperice aplikacija slijedno pruža pristup prethodno odabranim vježbama gdje korisnik upisuje svoje postignute uspjehe u odrađenoj tjelovježbi. Aplikacija bilježi osobne rekorde i pruža raznolik odabir programa za tjelovježbu.



Slika 2.3. Sučelje aplikacije RepCount

2.2. Vlastito rješenje za praćenje učinka

Problematika kod prethodno navedenih postojećih aplikacija su mikrotransakcije unutar njih samih. Aplikacije su besplatne za preuzeti, ali kako bi se omogućila potpuna njihova funkcionalnost potrebno je izdvojiti određeni iznos u obliku mikrotransakcije. Unutar svih aplikacija uvid u grafove se tek dobiva nakon kupljenja pune verzije aplikacije. Za razliku od postojećih aplikacija, ovaj završni rad je u potpunosti besplatan. Ima mogućnost bilježenja tjelesne težine i maksimalne opterećenosti tokom pojedine vježbe kroz određeni period vremena te uvid u vizualni prikaz napretka pomoću grafikona. Aplikacija pruža mogućnost pohranjivanja videa gdje korisnik bilježi proces izvođenja pojedinih vježbi i pohranjivanje slika fizičkog izgleda korisnika. Uvidom u slike korisnik može lakše pratiti vlastiti napredak što doprinosi motivaciji korisnika.

3. OPIS KORIŠTENIH TEHNOLOGIJA

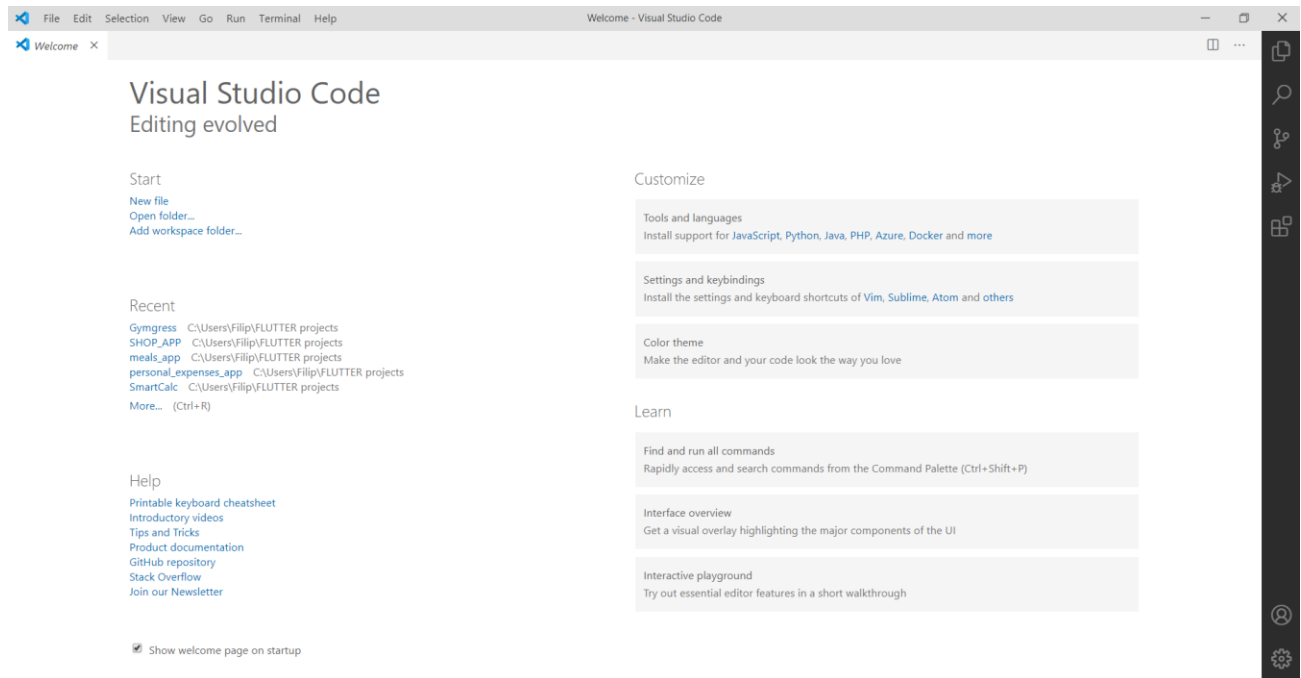
Izrada ovog završnog rada izvršena je pomoću mješavine modernih tehnologija za izradu mobilnih aplikacija kao što su Flutter i programski jezik Dart te SQL (engl. *Structured Query Language*). Aplikacija je pisana unutar Visual Studio Code razvojnog okruženja. Unutar ovog poglavlja detaljnije su opisane korištene tehnologije.

3.1. Android

Android operacijski sustav je jedan od najkorištenijih operacijskih sustava diljem svijeta. Osnovan je na bazi Linux jezgre (engl. Kernel). Sastoji se od četiri glavne cjeline, a to su jezgra, knjižnice, aplikacijski okviri i aplikacije. [1] Osnivači *Android Inc* tvrtke su započeli razvijanje operacijskog sustava koji je na kraju završio namijenjen za mobilne uređaje. Njihovu tvrtku otkupio je Google 2005 godine. Nastavkom razvijanja Androida 2007. godine prvi put je predstavljen operacijski sustav za mobilne uređaje na bazi Linux jezgre. Android je operacijski sistem konstruiran za mobitele, to je mjesto gdje se nalaze sve mobilne funkcije i aplikacije. [2] Glavni programski jezik na kojima je zasnovan Android su C/C++ pomoću kojeg su napisane knjižnice. Neke od glavnih knjižnica su *Surface Manager*, *OpenGL*, *Media Framework*, *SQLite* i brojne druge. Postoje brojni jezici za razvitak Android aplikacija, a neki od njih su *Java*, *Kotlin*, *C++*, *C#* i *Flutter*. [3] Neka od glavnih obilježja Androida su *Google Play* trgovina koja sadrži aplikacije podržane na Android uređajima i brojne podrške za očvrsja (engl. *hardware*) kao što su zaslon osjetljiv na dodir te podršku za mnoge senzore. Posljednja inačica Androida je Android 10, a neke od prethodnih su *Marshmallow*, *Nougat*, *Oreo* i *Pie*.

3.2. Visual Studio Code

Visual Studio Code je razvojno okruženje za izradu aplikacija. Pruža brz uređivač izvornog koda koji je jednostavan za upotrebu. Visual Studio Code ima podršku za stotine jezika te ima mogućnost isticanja sintakse koda, pridruživanja odgovarajućih zagrada, automatsko uvlačenje redaka, odabir okvira, isječcima i još mnogo toga. Podržava mnoštvo korisniku prilagodljivih tipkovničkih prečaca. [4]



Slika 3.1. Razvojno okruženje Visual Studio Code

Visual Studio Code ima mogućnost pokretanja virtualnog uređaja (engl. *Emulator*) koji uveliko olakšava testiranje aplikacije. Virtualni uređaj je zamjena za fizički uređaj na kojem je moguće pokrenuti i testirati aplikaciju tokom pisanja njezinog koda. Osim virtualnog uređaja, Visual Studio Code omogućava traženje pogrešaka (engl. *debugging*) [5]. Pomoću traženja pogrešaka razvojni inženjer ima mogućnost prolaska kroz kod korak po korak kako bi se lakše pronašla, a zatim otklonila pogreška. Razvojnem inženjeru Visual Studio Code nudi mogućnost automatskog nadopunjavanja teksta što ubrzava proces pisanja koda.

3.3. Flutter

Flutter je Googleov alat za izradu aplikacija namijenjenih za mobilne uređaje, web i stolna računala. Stvorio ga je Google 2015. godine, a objavljen je 2018. godine. [6] Flutter omogućuje stvaranje mobilne aplikacije sa samo jednom bazom koda. Moguće je koristiti jedan programski jezik i jednu bazu koda za izradu dvije aplikacije (za iOS i Android). [7] Flutter se sastoji od dvije komponente, SDK (engl. *Software Development Kit*) i okvira (engl. *Framework*). Za izradu aplikacija Flutter koristi zaseban programski jezik Dart. Glavna razlika između Fluttera i većine ostalih programskih jezika je što se njegov kod može pokrenuti na iOS i Android uređajima te nije potrebno pisanje dva različita koda u dva različita jezika, jedan za Android i jedan za iOS. Naziva se

višeplatformnim (engl. *Crossplatform*) alatom. Flutter je jednostavan, a efikasan razvojni okvir. Sadrži brojne nove značajke koje olakšavaju izradu mobilnih aplikacija, a glavna je brzo osvježavanje aplikacije (engl. *Hot-Reload*) [8] pomoću koje se na zaslonu virtualnog uređaja ili uređaja spojenog na računalo prikazuju promjene u stvarnom vremenu (engl. *Real-Time*). Svaki element je prikazan u obliku kontrole (engl. *Widget*) [9]

3.4. SQL

Strukturni upitni jezik odnosno SQL (engl. *Structured Query Language*) je programski jezik za upravljanje bazama podataka. Omogućuje stvaranje, brisanje i uređivanje tablica i podataka koji se nalaze unutar njih. [10] Prvi puta je objavljen 1978. godine. Naredbe za korištenje SQL bazom podataka se razdvajaju na naredbe za upravljanje bazom i naredbe za upravljanje podacima. Neke od glavnih naredbi su *CREATE*, *SELECT*, *DELETE*, *INSERT*, *ALTER*, *UPDATE*. [11] Postoje brojne mogućnosti na koje sve načine je moguće dohvatiti, sortirati ili grupirati podatke. SQL se smatra standardom za upravljanje bazama podataka.

3.5. Dart

Dart je programski jezik koji se prvobitno koristio u *Googlu* za izgradnju web stranica, poslužitelja (engl. *server*) i mobilnih aplikacija. *Dart* postoji od 2011. godine. *Dart* sastavlja izvorni kod slično kao drugi programski jezici. Glavna razlika je što *Dart* vlastitog upravitelja paketa zvanog *Pub*. *Dart* je počeo privlačiti više pažnje 2017. godine kada je *Google* službeno najavio *Flutter beta* za razvoj različitih platformi za mobilne aplikacije. [12]

4. RAZVOJ APLIKACIJE

Unutar ovog poglavlja prikazana je cjelokupna izrada aplikacije. Objasnjene su sve komponente, odnosno njihova funkcionalnost i implementacija unutar ovog završnog rada.

4.1. Koncept aplikacije

Osmišljeni koncept aplikacije ima funkcionalnost praćenja tjelesne težine i maksimalne težine ostvarene za pojedine vježbe. Početni zaslon sadrži dvije kartice (engl. *tab*) Na svakom od njih treba biti prikazan zasebni zaslon, pomoću jednog se bilježe podatci vezani uz tjelesnu težinu, a drugi zaslon omogućava izbor vježbi. Početni zaslon za praćenje tjelesne težine sadrži graf pomoću kojeg je prikazano mijenjanje tjelesne težine u ovisnosti o vremenu, prikaz trenutne tjelesne težine, gumb pomoću kojeg se otvara zaslon za unos novih podataka odnosno fotografija i gumb koje otvara galeriju svih unesenih fotografija. Drugi zaslon u kartici prikazuje listu vježbi čijim odabirom se otvara novi zaslon koji prikazuje graf pridružen toj vježbi kojim se prati učinak te vježbe u ovisnosti o vremenu, maksimalnu podignutu težinu za odabranu vježbu, video izvršavanja vježbe i gumb pomoću kojeg se otvara novi zaslon. Svi unosi se pohranjuju u bazu podataka iz koje se uzimaju podatci kako bi se mogli prikazati na zaslonu.

4.2. Implementacija aplikacije

Prije samog početka pisanja aplikacije potrebno je napraviti novu mapu te unutar Visual Studio Codea u terminal upisati „flutter create IME_APLIKACIJE“ kako bi stvorili novi Flutter projekt, a zatim je potrebno napraviti dizajn i logiku unutar aplikacije. Prvim pokretanjem aplikacije izvršavaju se funkcije unutar „main.dart“ datoteke. Pozivaju se „_createImagesDir()“ i „_createVideosDir()“ pomoću kojih se unutar datoteka aplikacije stvaraju nova dva direktorija s imenima „ImageGallery“ i „VideoGallery“. (slika 4.1.)

```

void _createImagesDir() async {
  final io.Directory _appDocDir = await getApplicationDocumentsDirectory();
  final io.Directory _appDocDirImages =
  | | io.Directory('${_appDocDir.path}/ImageGallery/');
  if (await _appDocDirImages.exists()) {
  | return;
  } else {
  | await _appDocDirImages.create(recursive: true);
  }
}

void _createVideosDir() async {
  final io.Directory _appDocDir = await getApplicationDocumentsDirectory();
  final io.Directory _appDocDirImages =
  | | io.Directory('${_appDocDir.path}/VideoGallery/');
  if (await _appDocDirImages.exists()) {
  | return;
  } else {
  | await _appDocDirImages.create(recursive: true);
  }
}

@override
void initState() {
  super.initState();
  _createImagesDir();
  _createVideosDir();
}

```

Slika 4.1. Stvaranje novih direktorija

Uz stvaranje novih direktorija, unutar „main.dart“ datoteke postavljena je željena tema aplikacije i potrebne rute pomoću kojih se vrši navigacija među zaslonima. (slika 4.2.)


```

theme: ThemeData(
  scaffoldBackgroundColor: Color(0xffff2f2f),
  primaryColor: Color(0xffcccc),
  accentColor: Color(0xff404040),
  errorColor: Color(0xffff0000),
  primarySwatch: Colors.blue,
  textSelectionColor: Color(0xff1919ff),
  textTheme: TextTheme(
    headline6: TextStyle(
      fontSize: 25.0,
      fontWeight: FontWeight.bold,
      color: Color(0xff1919ff),
    ), // TextStyle
  ), // TextTheme
), // ThemeData
routes: {
  '/': (ctx) => TabsScreen(),
  MyBodyScreen.routeName: (ctx) => MyBodyScreen(),
  ExercisesListScreen.routeName: (ctx) => ExercisesListScreen(),
  NewPhotoScreen.nameRoute: (ctx) => NewPhotoScreen(),
  GalleryScreen.routeName: (ctx) => GalleryScreen(),
  ExerciseScreen.routeName: (ctx) => ExerciseScreen(),
  NewVideoScreen.nameRoute: (ctx) => NewVideoScreen(),
},
); // MaterialApp

```

Slika 4.2. Postavljanje teme i ruta aplikacije

Unutar zasebne datoteke nazvane „*DBHelper.dart*“ napisane su sve funkcije za stvaranje baze podataka, tablica, pohranjivanje i dohvaćanje podataka iz postojeće baze. Pozivanjem baze podataka izvršava se funkcija „*_onCreate*“. Ona stvara dvije tablice unutar kojih će se moći pohranjivati jedinstveni identifikator, datum i težina. (slika 4.3.)

```

_onCreate(Database db, int version) async {
  await db.execute(
    'CREATE TABLE $TABLE1($ID INTEGER,$DATE DATETIME, $WEIGHT INTEGER)');
  await db.execute(
    'CREATE TABLE $TABLE2($ID INTEGER,$DATE DATETIME, $WEIGHT INTEGER)');
}

```

Slika 4.3. Stvaranje tablica baze podataka

„*DBHelper.dart*“ sadrži funkcije za pohranjivanje tjelesnih težina i vježbi, dohvaćanje tjelesnih težina i vježbi te dohvaćanje podataka za grafikone. Unutar funkcija za dohvaćanje grafikona korišten je SQL jezik. Njime se postavljaju uvjeti za dohvaćanje podataka iz baze podataka, kao na primjer dohvaćanje samo odabiranje prosječne tjelesne težine i datuma gdje je izraz grupiran po datumima. Kod pohranjivanja i dohvaćanja podataka iz baze za „*ExercisesScreen.dart*“ potrebno je dodatno postaviti uvjet za dohvaćanje podataka samo za traženu vježbu pomoću „*WHERE*“ upita. (slika 4.4.)

```

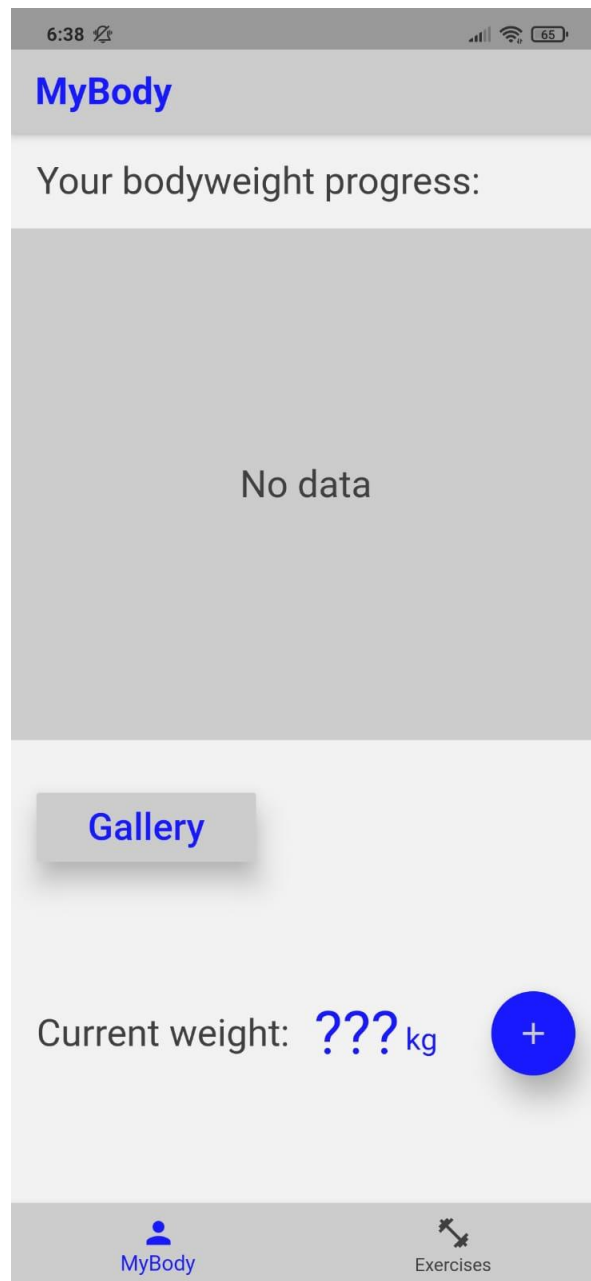
Future<List<ExerciseInfo>> getExerciseInfos(int id) async {
  var dbClient = await database;
  List<Map<String, dynamic>> maps =
  |   await dbClient.query(TABLE2, where: '$ID = ?', whereArgs: [id]);
  return maps.map((e) => ExerciseInfo.fromMap(e)).toList();
}

Future<List<ChartInfo>> getExerciseChartInfos(int id) async {
  var dbClient = await database;
  List<Map<String, dynamic>> maps = await dbClient.rawQuery('''
  SELECT AVG(weight) as weight, date
  FROM $TABLE2
  WHERE $ID = ?
  GROUP BY date
  |   ''', [id]);
  return maps.map((e) => ChartInfo.fromMap(e)).toList();
}

```

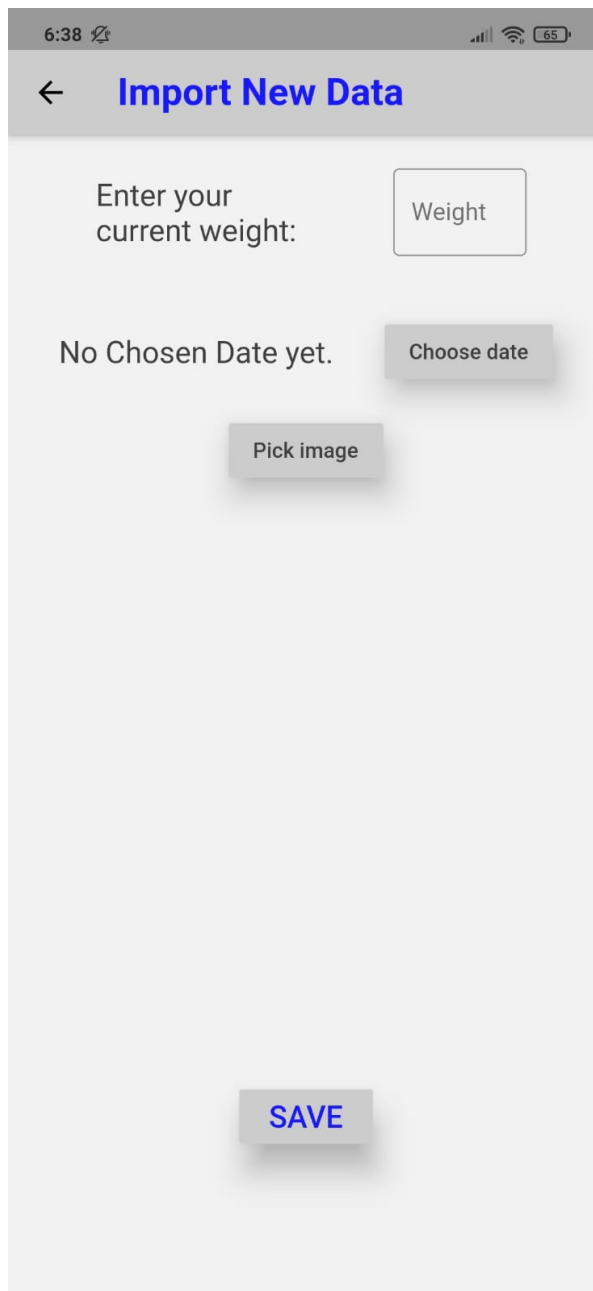
Slika 4.4. Dohvaćanje podataka za grafikon

Početnom rutom određeno je kako se prvo pokreće „TabsScreen.dart“ kojim se stvara „BottomNavigationBar“. Pritiskom na ikonu na dnu zaslona odabire se između „MyBody.dart“ zaslona koji je ujedno i početni zaslon aplikacije te „Exercises.dart“ zaslona. (Slika 4.5.) „MyBody.dart“ zaslon raspodijeljen je na elemente smještene unutar stupca. Prvi element je tekst „Your bodyweight progress:“. Svaki tekst unutar aplikacije je posebno uređen, odnosno izmijenjeni su mu veličina i boja za razliku od zadanog teksta. Drugi element u stupcu je grafikon (engl. *chart*) koji će se prikazati ukoliko baza podataka nije prazna, odnosno nakon što korisnik unese podatke.



Slika 4.5. „MyBody.dart“ zaslon

Zatim slijedi gumb (engl. *button*) koji otvara galeriju unutar koje se pohranjuju sve slike spremljene za „MyBody.dart“ zaslon. Mogućnost unošenja i pohranjivanja novih podataka nalazi se unutar zadnjeg elementa na zaslonu. Zadnji element na zaslonu je redak koji u prvom dijelu ispisuje korisnikovu trenutnu tjelesnu težinu, a u nastavku se nalazi „*FloatingActionButton*“ . Pritiskom na taj gumb otvara se novi zaslon „*NewPhoto.dart*“ unutar kojeg se unose podatci trenutne tjelesna težina, odabir datuma i odabir slike. (Slika 4.6.)



Slika 4.6. Unos podataka o tjelesnoj težini

„*NewPhoto.dart*“ zaslon sastoji se od pet elemenata unutar stupca. Prvi element je redak unutar kojeg se nalaze tekst „*Enter your current weight:*“ i tekstualno polje (engl. *textField*). Tekstualno polje korisniku omogućuje unos teksta preko tipkovnice, a pomoću „*keyboardType: TextInputType.number*“ ga ograničavamo samo na unos brojeva. Sljedeći element je ponovno redak, no umjesto tekstualnog polja elementa nalazi se gumb na koje pritiskom poziva funkciju „*pickDate*“.

(slika 4.7.) Prikazuje se na zaslonu „*DatePicker*“ pomoću kojeg se odabire datum. Zatim se nalazi gumb za odabir slike koja se nakon što je odabrana prikaže ispod tog gumba na ekranu.

```
void _pickImage() async {
  final PickedFile pickedImage =
    | await ImagePicker().getImage(source: ImageSource.gallery);
  if (pickedImage == null) return;
  _imageName = DateTime.now().toIso8601String();
  setState(() {
    | _image = File(pickedImage.path);
  });
}

void _saveImage() async {
  imagesPath = await _getImagesDirPath();
  if (_image == null) return;
  _image = await _image.copy('$imagesPath/$_imageName');
  Navigator.of(context).pop();
}

void _pickDate() async {
  DateTime start = DateTime(DateTime.now().year, 1, 1);
  DateTime end = DateTime.now();
  showDatePicker(
    context: context,
    initialDate: end,
    firstDate: start,
    lastDate: end,
  ).then((pickedDate) {
    if (pickedDate == null) return;
    setState(() {
      | _pickedDate = pickedDate;
    });
  });
}
```

Slika 4.7. NewPhoto funkcije

Prilikom odabira gumba „*Save*“ otvara se baza podataka i stvara se „*BodyweightInfo*“ objekt. Objekt se zatim pohranjuje u bazu podataka pomoću funkcije „*saveBodyweight*“ (slika 4.8.)

```

RaisedButton(
  onPressed: () {
    if (_image == null || _weight == null || _pickedDate == null)
      return;
    DBHelper dbHelper = DBHelper();
    BodyweightInfo bodyweightInfo =
      BodyweightInfo(id: 0, date: _pickedDate, weight: _weight);
    dbHelper.saveBodyweight(bodyweightInfo);
    _saveImage();
  },
  ...
)

```

Slika 4.8. Pohranjivanje u bazu podataka

Druga kartica prikazuje „*Exercises.dart*“ zaslon na kojem se nalazi popis vježba. Vježbe su prikazane korištenjem popisnog pregleda (engl. *ListView*). (slika 3.8.)

```

Widget build(BuildContext context) {
  MediaQueryData mediaQuery;
  mediaQuery = MediaQuery.of(context);

  return ListView.separated(
    itemCount: exercises_list.length,
    padding: EdgeInsets.all(mediaQuery.size.height * 0.02),
    separatorBuilder: (context, index) => Divider(
      color: Colors.black,
    ), // Divider
    itemBuilder: (context, index) => ExerciseItem(
      id: exercises_list[index].id,
      name: exercises_list[index].name,
    )); // ExerciseItem // ListView.separated
}

```

Slika 4.9. Kod popisnog pregleda vježbi

Svaki element liste vježbi je objekt „*ExerciseItem*“ i on sadrži jedinstveni identifikator pojedine vježbe te njezin naziv.

```

class ExerciseItem extends StatelessWidget {
  final int id;
  final String name;

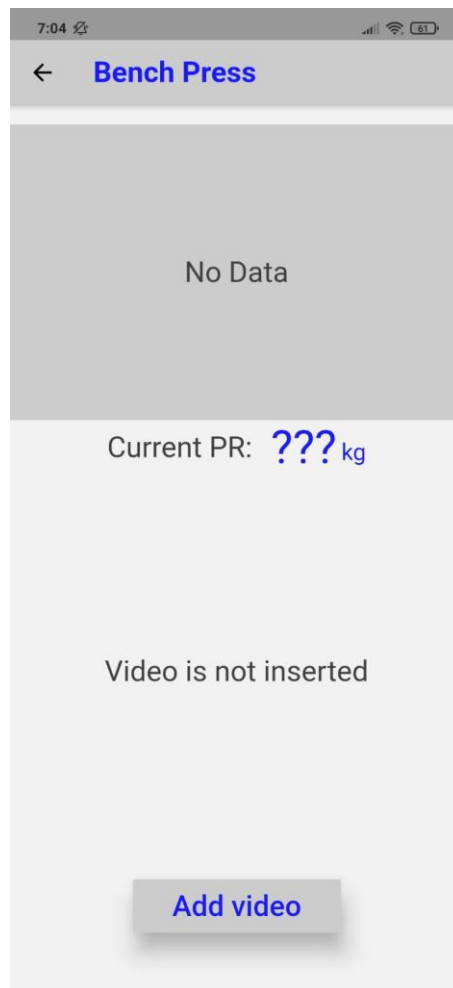
  ExerciseItem({this.id, this.name});

  void _selectCategory(BuildContext context) {
    Navigator.of(context).pushNamed(ExerciseScreen.routeName, arguments: {
      'id': id,
      'name': name,
    });
  }
}

```

Slika 4.10. Element liste vježbi

Odabirom jedne od ponuđenih vježbi otvara se novi zaslon na kojem su prikazani podatci za odabranu vježbu. Prikazani podatci nalik su podacima „*MyBody.dart*“ zaslonu. (slika 4.11.)



Slika 4.11. "Exercise.dart" zaslon odabrane vježbe

Na vrhu se prikazuje ime odabrane vježbe. Unutar stupca prikazuju se grafikon podataka unesenih o odabranoj vježbi ukoliko baza podataka nije prazna. Prikazuje se trenutna maksimalna podignuta težina, zadnje spremljeni videozapis i dugme pomoću kojeg korisnik dodaje nove podatke za odabranu vježbu. Videozapis je prikazan pomoću *Chewiea* kojeg je potrebno nadodati pomoću njegove knjižnice. (slika 4.12.)

```

child: (exerciseInfos.length > 0 && _chewieController != null)
  ? FittedBox(
    fit: BoxFit.contain,
    child: Chewie(
      controller: _chewieController,
    ), // Chewie
  ) // FittedBox
  : Center(
    child: Text(
      'Video is not inserted',
      style: TextStyle(
        color: Theme.of(context).accentColor,
        fontSize: 25.0,
      ), // TextStyle
    ), // Text
  ), // Center
);

ChewieController _setChewieController(videoController) {
  VideoPlayerController controller = videoController;
  return ChewieController(
    videoPlayerController: controller,
    aspectRatio: controller.value.aspectRatio,
    autoplay: false,
    looping: false,
    autoInitialize: true,
  );
}

```

Slika 4.12. Prikaz videozapisa

Odabirom gumba „Add video“ otvara se zaslon identičan zaslonu za unos podataka o tjelesnoj težini. (slika 4.14.) Odabirom gumba „Pick video“ poziva se funkcija pomoću koje korisnik odabire video iz galerije. Odabrani video se zatim prikazuje na ekranu. Nakon što korisnik unese sve podatke odabirom na gumb „Save“ podatci se pohranjuju u bazu podataka. (slika 4.13.)

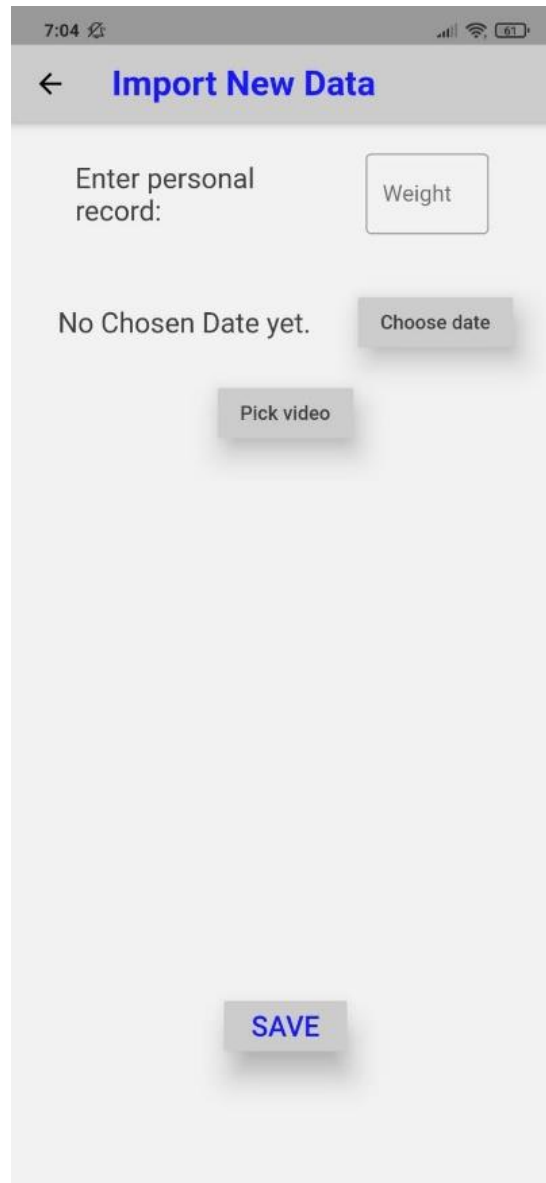
```

void _pickVideo() async {
  final PickedFile pickedVideo =
    await ImagePicker().getVideo(source: ImageSource.gallery);
  if (pickedVideo == null) return;
  if (_videoName != null) {
    videosPath = await _getVideosDirPath();
    Directory('${videosPath}/${_videoName}').delete(recursive: true);
  }
  _newVideoFile(pickedVideo);
  _videoPlayerController = VideoPlayerController.file(_video);
  _chewieController = _setChewieController(_videoPlayerController);
}

RaisedButton(
  onPressed: () {
    if (_video == null || _weight == null || _pickedDate == null)
      return;
    DBHelper dbHelper = DBHelper();
    ExerciseInfo exerciseInfo = ExerciseInfo(
      id: this.id, date: _pickedDate, weight: _weight); // ExerciseInfo
    dbHelper.saveExercise(exerciseInfo);
    _saveVideo();
    if (_chewieController != null) {
      _chewieController.pause();
      _chewieController.seekTo(Duration(seconds: 0));
    }
  },
);

```

Slika 4.13. Odabir i pohranjivanje videozapisa



Slika 4.14. Unos podataka za odabranu vježbu

Podatci vezani uz tjelesnu težinu korisnika pohranjuju se u obliku „*BodyweightInfo*“ objekta, a podatci za pojedine vježbe se spremaju u obliku „*ExerciseInfo*“ objekta. Unutar zasebnih datoteke definirani su objekti i njihove metode „*toMap*“ i „*fromMap*“ pomoću kojih je omogućeno upravljanje objektima tih klasa. (slika 4.15.)

```

Map<String, dynamic> toMap() {
  var map = <String, dynamic>{
    'id': id,
    'weight': weight,
    'date': date.toIso8601String(),
  };
  return map;
}

ExerciseInfo.fromMap(Map<String, dynamic> map) {
  id = map['id'];
  weight = map['weight'];
  date = DateTime.parse(map['date']);
}

```

Slika 4.15. Metode za upravljanje „*BodyweighInfo*“ i „*ExerciseInfo*“ objektima

Grafikoni se unutar „*MyBody.dart*“ zaslona i „*Exercise.dart*“ zaslona prikazuju pozivom „*Charta*“ kojem je potrebno predati listu objekata koje prikazuje na zaslonu.

```

class _ChartState extends State<Chart> {
  @override
  Widget build(BuildContext context) {
    List<charts.Series<ChartInfo, DateTime>> series = [
      charts.Series(
        id: 'chart',
        data: this.widget.data,
        domainFn: (series, _) => series.date,
        measureFn: (series, _) => series.weight,
        colorFn: (_, _) => charts.MaterialPalette.blue.shadeDefault,
      ),
    ];
    return charts.TimeSeriesChart(series, animate: true);
  }
}

child: (exercisesChartInfos.length > 0)
  ? Chart(
    data: exercisesChartInfos,
  ) // Chart
  : Center(
    child: Text(
      'No Data',
      style: TextStyle(
        color: Theme.of(context).accentColor,
        fontSize: 25.0,
      ), // TextStyle
    ), // Text
  ), // Center

```

Slika 4.16. Klasa „*Chart*“ i njena uporaba

Jedna od glavnih funkcija korištenih unutar „*MyBody.dart*“ zaslona i „*Exercise.dart*“ zaslona je funkcija „*_refreshInfos()*“ koja je zadužena za dohvaćanje podataka iz baze podataka i njihovo spremanje u liste. Naredbom „*setState()*“ funkcija novo prikupljene podatke prikazuje na zaslonu. (slika 4.17.)

```

void _refreshInfos() {
  dbBodyweightInfo.getBodyweightInfos().then((weightInfo) {
    setState(() {
      bodyweightInfos.clear();
      bodyweightInfos.addAll(weightInfo);
      if (bodyweightInfos.length > 0) {
        bodyweightInfos.sort((a, b) {
          return a.date.compareTo(b.date);
        });
        _weight = bodyweightInfos.last.weight;
      }
    });
    dbBodyweightInfo.getBodyweightChartInfos().then((chartInfo) {
      setState(() {
        bodyweightChartInfos.clear();
        bodyweightChartInfos.addAll(chartInfo);
      });
    });
  });
}

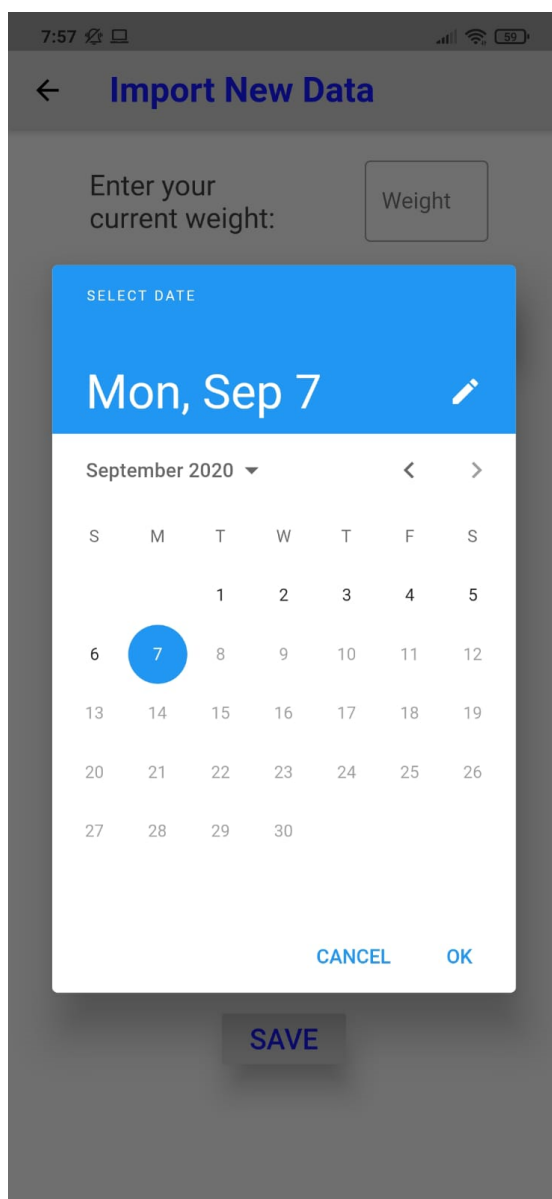
void _refreshInfos() {
  dbExercisesInfo.getExerciseInfos(id).then((exerciseInfo) {
    setState(() {
      exerciseInfos.clear();
      exerciseInfos.addAll(exerciseInfo);
      if (exerciseInfos.length > 0) {
        exerciseInfos.sort((a, b) {
          return a.date.compareTo(b.date);
        });
        _weight = exerciseInfos.last.weight;
      }
    });
    dbExercisesInfo.getExerciseChartInfos(id).then((chartInfo) {
      setState(() {
        exercisesChartInfos.clear();
        exercisesChartInfos.addAll(chartInfo);
      });
    });
    if (exerciseInfos.length > 0) {
      _getVideosDirPath().then((path) {
        setState(() {
          _videoPlayerController =
            VideoPlayerController.file(File('$path/videosid'));
          _chewieController = _setChewieController(_videoPlayerController);
        });
      });
    }
  });
}

```

Slika 4.17. Kod „*_refreshInfos()*“ funkcije

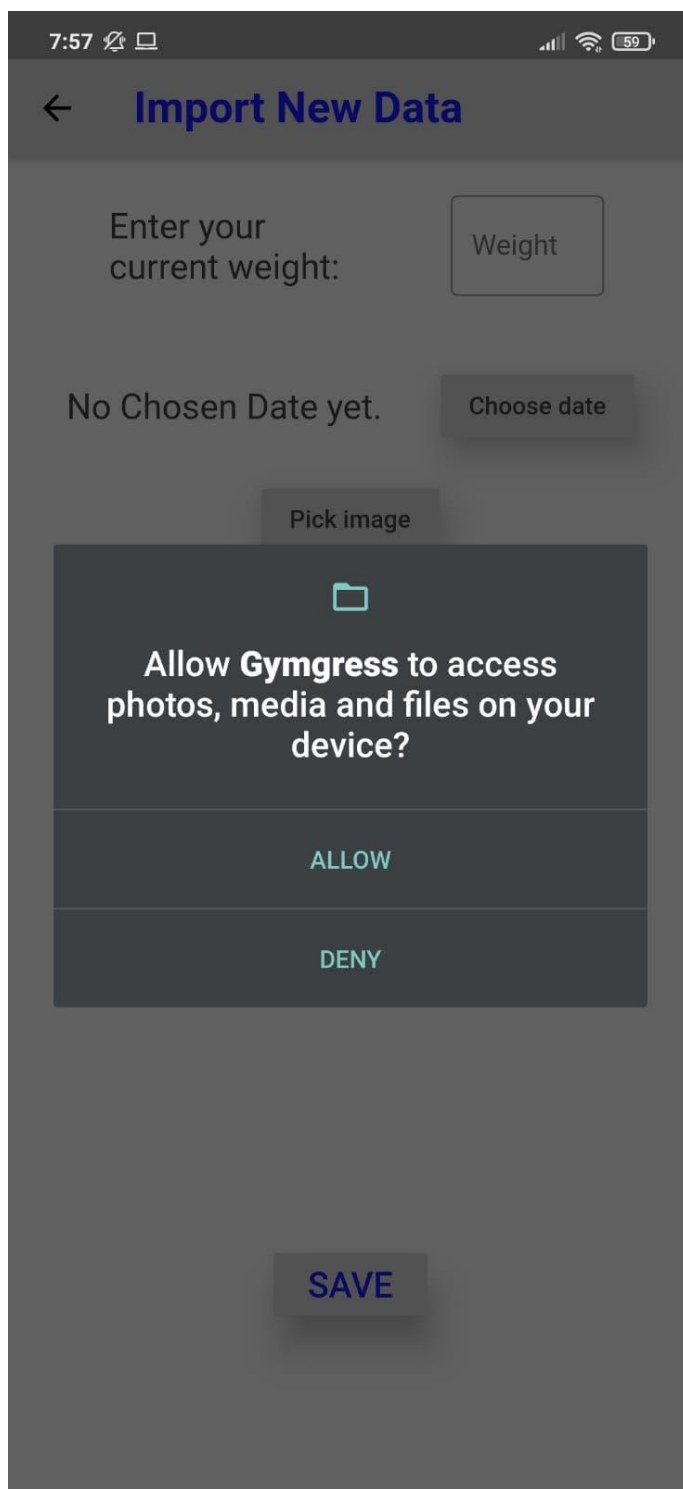
5. KORIŠTENJE APLIKACIJOM

Prilikom ulaska u aplikaciju korisniku se prikazuje „*MyBody*“ zaslon. Pritiskom na gumb za dodavanje novih fotografija korisniku se otvara „*NewPhoto*“ zaslon unutar kojeg korisnik upisuje podatke. Prvo korisnik upisuje izmjerenu tjelesnu težinu. Odabiranjem gumba „Pick date“ prikazuje se *DatePicker* pomoću kojeg korisnik odabire datum kojeg je tjelesna težina izmjerena. (slika 5.1.)



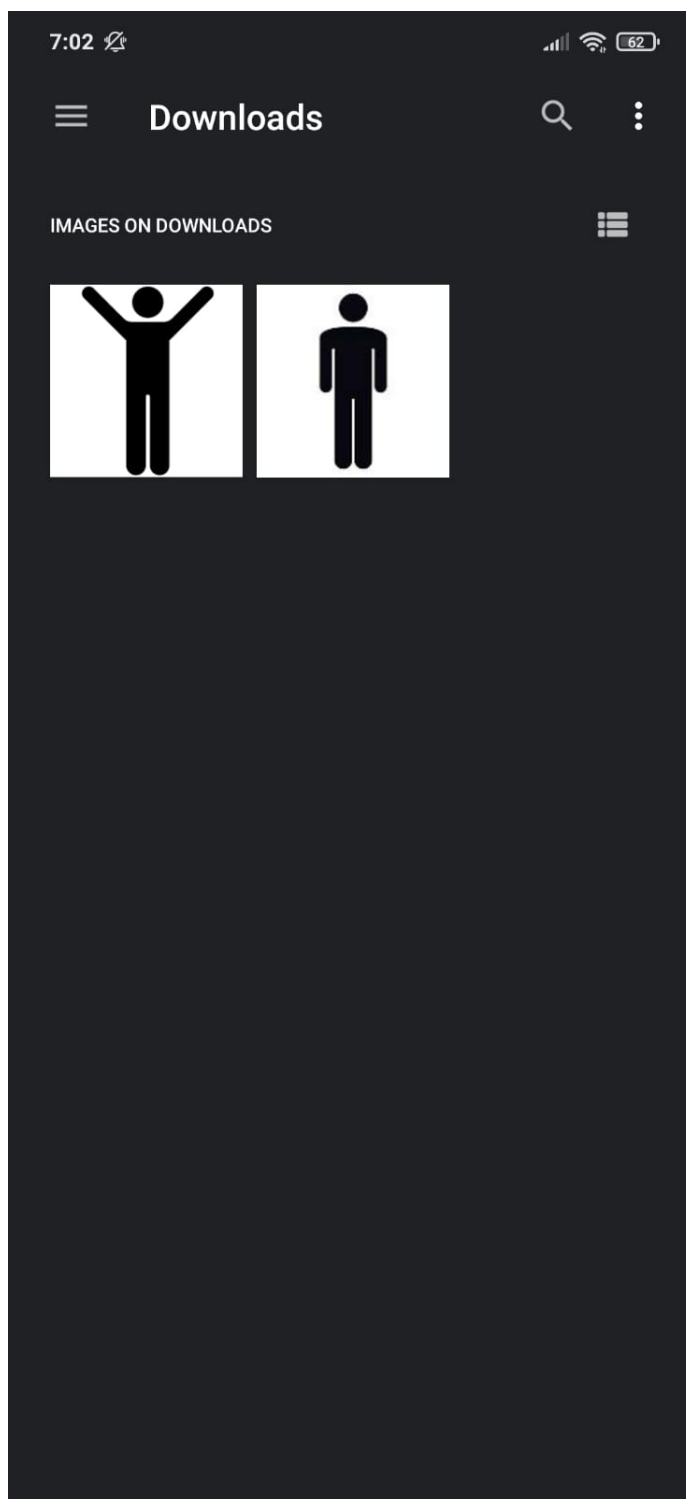
Slika 5.1. Odabir datuma pomoću *DatePickera*

Nakon što korisnik odabere željeni datum, prilikom pritiska na gumb „*Pick image*“ korisnik mora dopustiti „*Gymgress*“ aplikaciji pristup podacima na uređaju. (slika 5.2.)



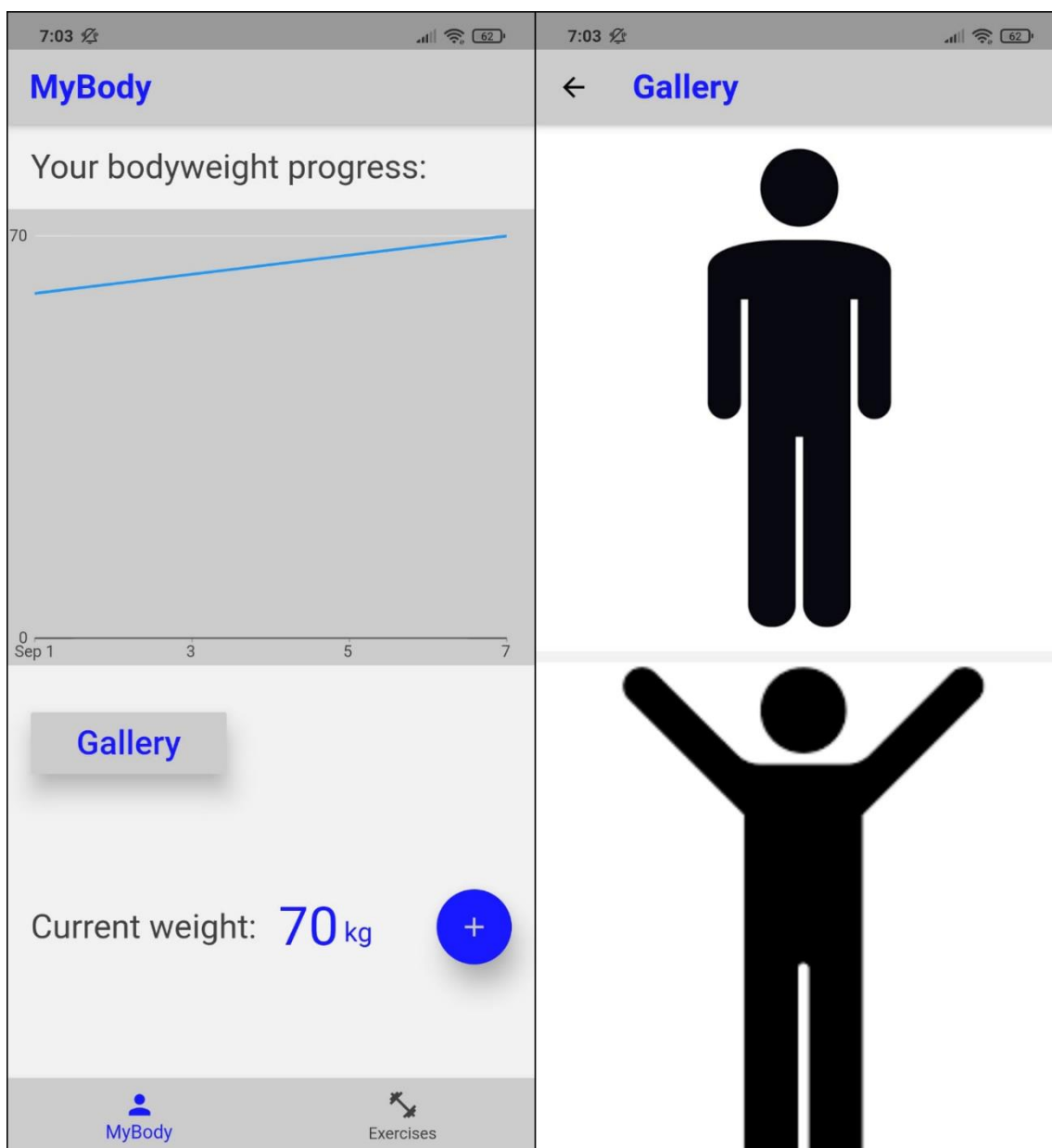
Slika 5.2. Dopuštenja za „*Gymgress*“ aplikaciju

Dopuštanjem pristupa korisniku se otvara novi zaslon na kojem odabire željenu fotografiju koju želi pohraniti u galeriju. (slika 5.3.)



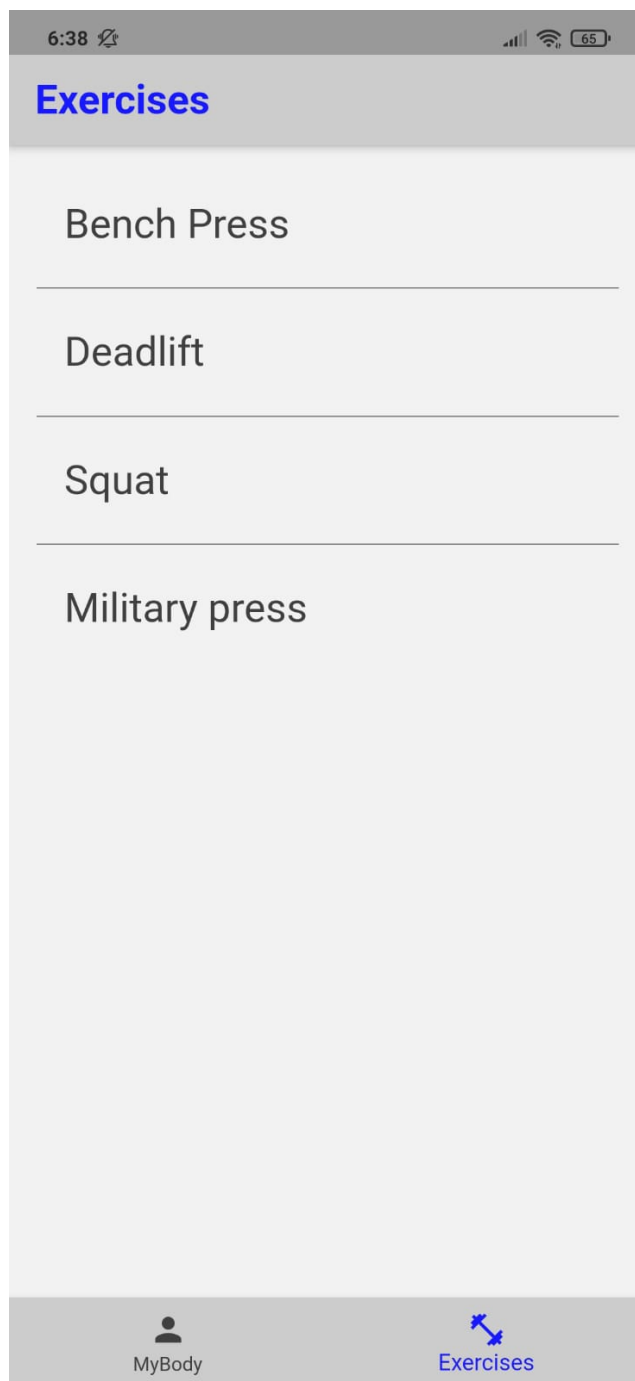
Slika 5.3. Odabir fotografije

Nakon pohranjivanja unesenih podataka, podatci se prikazuju na „MyBody.dart“ zaslonu. Prikazuje se grafikon na kojem su prikazani svi prethodno pohranjeni podatci korisnika. Grafikon prikazuje promjenu tjelesne težine u ovisnosti o vremenu. Prilikom odabira gumba „Gallery“ korisniku se prikazuju spremljene fotografije. (slika 5.4.)



Slika 5.4. Prikaz podataka unesenih u „NewPhoto.dart“ zaslonu

Promjenom kartice odabirom „Exercises“ korisniku se prikazuje lista postojećih vježbi. (slika 5.5.) Odabirom jedne od ponuđenih vježbi korisniku se otvara novi zaslon s podatcima za odabranu vježbu.



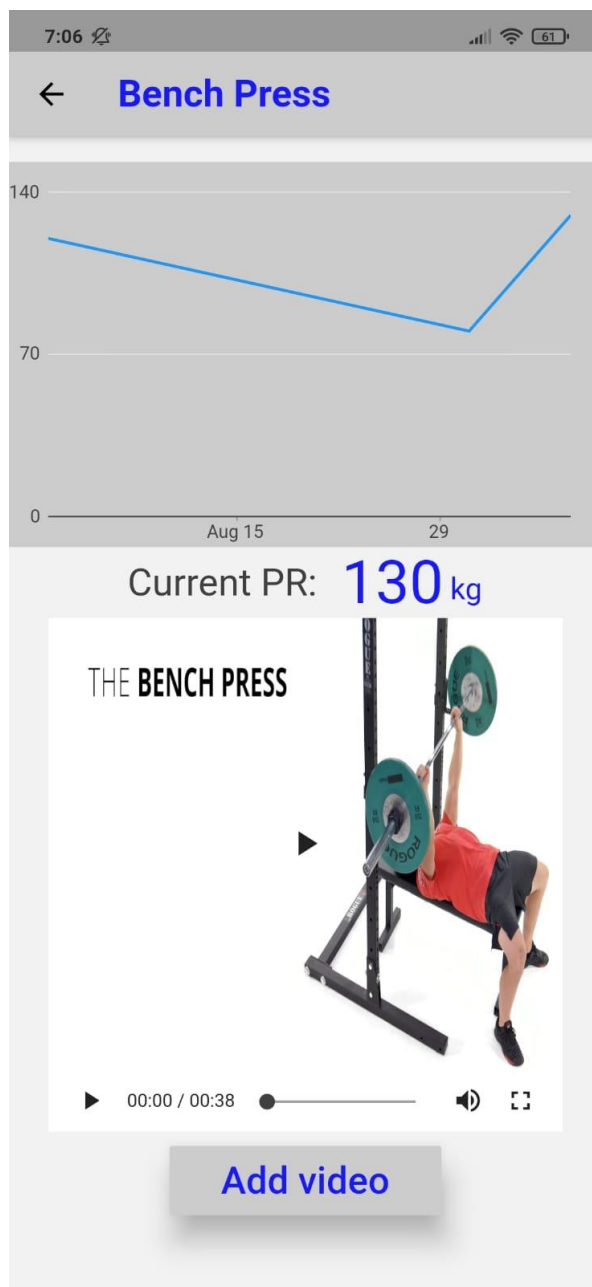
Slika 5.5. Lista postojećih vježbi

Odabirom na gumb „Add video“ korisniku se otvara zaslon za unošenje podataka o odabranoj vježbi. Nakon upisivanja podignute težine i odabira datuma, korisnik pritiskom na gumb „Pick video“ otvara novi zaslon na kojem odabire željeni videozapis koji želi pohraniti. (slika 5.6.)



Slika 5.6. Odabir videozapisa

Nakon pohranjivanja unesenih podataka, podatci se prikazuju na „*Exercises.dart*“ zaslonu. Prikazuje se grafikon koji prikazuje promjenu tjelesne težine u ovisnosti o vremenu od svih prethodno unesenih podataka za odabranu vježbu. (slika 5.7.)



Slika 5.7. Prikaz podataka odabrane vježbe

Ispod grafikona prikazuje se posljednje pohranjeni video korisnikovog izvođenja odabrane vježbe.

6. ZAKLJUČAK

Zadatak ovog završnog rada bio je napraviti mobilnu aplikaciju za praćenje učinka u teretani. Aplikacija je napravljena pomoću relativno novog *Googleovog* alata Flutter, a kod je napisan pomoću programskog jezika *Dart*. Prvi dio završnog rada ukratko opisuje ideju ovog završnog rada i uspoređuje ga s tri postojeća rješenja. Prikazane su sličnosti i razlike između ovog završnog rada i rješenja već postojećih na tržištu. Drugi dio ovog završnog rada daje uvid u korištene tehnologije za pisanje ovog završnog rada. Unutar trećeg djela ovog završnog rada opisan je u potpunosti razvoj aplikacije. Prikazan je koncept odnosno ideja stvaranja aplikacije i osmišljena funkcionalnost pojedinih dijelova. Nakon koncepta detaljno je opisana potpuna implementacija svih dijelova aplikacije ovog završnog rada. Objašnjene su sve datoteke i njihov sadržaj i prikazan je opći izgled aplikacije uz opis svakog elementa prikazanog na aplikaciji. Na kraju ovog završnog rada pojašnjeno je korištenje aplikacijom i njena ukupna funkcionalnost. Prikazano je upravljanje aplikacijom i funkcionalnost cjelokupnog završnog rada. Aplikacija bi se mogla jednostavno nadograditi dodavanjem autorizacije i prijave korisnika. Tom nadogradnjom informacije pohranjene unutar aplikacije mogle bi se spremati na internetsku bazu podataka te ne bi bile spremljene samo na korisnikov uređaj. Moguće je dodavanje funkcionalnosti za upravljanje prethodno unesenih podataka, odnosno mogućnost brisanja ili promjene prethodno pohranjenih podataka u aplikaciju.

LITERATURA

- [1] J. J. A. K.Chinetha, »An Evolution of Android Operating System and Its,« 2 2015.
- [2] A. Authority, »Android Authority,« 1. 8. 2020. [Mrežno]. Available: <https://www.androidauthority.com/what-is-android-328076/>.
- [3] A. Sinicki, Android Authority, 10. 8. 2019. [Mrežno]. Available: <https://www.androidauthority.com/develop-android-apps-languages-learn-391008/>.
- [4] Microsoft, »Why did we build Visual Studio Code?,« 13. 8. 2020. [Mrežno]. Available: <https://code.visualstudio.com/docs/editor/whyvscode>.
- [5] D. Bolton, »5 Reasons to Use Visual Studio Code,« Dnsight Dice, 26. 5. 2017. [Mrežno]. Available: <https://insights.dice.com/2017/05/26/5-reasons-use-visual-studio-code/>.
- [6] C. Software, »What is Flutter? Here is everything you should know,« Medium, 26. 8. 2019. [Mrežno].
- [7] freeCodeCamp, »What is Flutter and Why You Should Learn it in 2020,« 12. 12. 2019. [Mrežno]. Available: <https://www.freecodecamp.org/news/what-is-flutter-and-why-you-should-learn-it-in-2020/>.
- [8] F. dev, »Hot reload,« Flutter, [Mrežno]. Available: <https://flutter.dev/docs/development/tools/hot-reload>.
- [9] T. Omotunde, »Flutter : Everything is a Widget Series — Part 1,« Medium, 1. 5. 2019. [Mrežno]. Available: <https://medium.com/@topeomot/flutter-everything-is-widgets-series-e40fffe0d44e>.
- [10] tutorialspoint, »tutorialspoint,« [Mrežno]. Available: https://www.tutorialspoint.com/sql/sql_tutorial.pdf.

[11] w3schools, »Introduction to SQL,« w3schools, [Mrežno]. Available:
https://www.w3schools.com/sql/sql_intro.asp.

[12] codemagic, »Dart vs JavaScript,« codemagic, 5. 5. 2019. [Mrežno].

SAŽETAK

Zadatak ovog završnog rada je napraviti mobilnu aplikaciju za olakšano kontinuirano praćenje učinka u teretani. Za razvitak mobilne aplikacije u Visual Studio Code i Flutter razvojnom okruženju korišteni su Dart programski jezik pomoću kojeg je napisan kod aplikacije te SQL baza podataka pomoću koje se pohranjuju uneseni podatci. Aplikacija omogućuje praćenje promjene tjelesne težine korisnika i maksimalne podignute težine pojedinih vježbi u ovisnosti o vremenu. Kod pohrane novih podataka potreban je korisnikov unos tjelesne težine ili podignute težine, odabir datuma te umetanje fotografije ili videozapisa ovisno o zaslonu kojeg korisnik ispunjava. Podatci pohranjeni unutar baze podataka korisniku se prikazuju u obliku grafikona te prikaza pohranjenih videozapisa i galerije fotografija. Prikazivanjem grafikona olakšava se praćenje učinka, a pohranjivanje fotografija i videozapisa korisniku pruža uvid u ostvareni napredak što doprinosi motivaciji korisnika.

Ključne riječi: Android, Baza podataka, Dart, Flutter, SQL

ABSTRACT

Android application for tracking gym progress

The task of this final thesis was to develop a mobile application for easy and continuous track of gym progress. Dart programming language was used to write application code along with SQL database which is used to store user data while developing the application in Visual Studio Code and Flutter developer environments. Application enables its users to track changes in their bodyweight along with their maximum exercise weight in relation to time. When adding new data, the user has to input their current weight or their exercise weight along with a date and photo or a video, depending on which screen is being filled. Data which is stored in the database is displayed to the user as a chart and also as a gallery of stored photos and videos. Progress tracking is facilitated by displaying a chart and storing photos and videos enables the user to track their progress which helps to motivate the user.

Keywords: Android, database, Dart, Flutter, SQL

ŽIVOTOPIS

Filip Forjan rođen je u Osijeku 25.5.1998. godine. Pohađao je osnovnu školu Ivana Filipovića u Osijeku. Nakon završetka osnovne škole upisuje III. Gimnaziju Osijek, smjer prirodoslovna matematička gimnazija. Nakon završetka srednjoškolskog školovanja upisuje se na preddiplomski sveučilišni studij Računarstvo na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija u nastavku FERIT. Zainteresiran je za razvoj i održavanje mobilnih aplikacija. Član je studentskog zbora FERIT-a.

Filip Forjan

PRILOZI

Na priloženom DVD disku nazali se:

- Kopija aplikacije *Gymgress* u .apk formatu namijenjena za *Android* uređaje
- Izvorni kod aplikacije
- Orginalna kopija završnog rada u DOCX i PDF formatu