

RAZVOJ APLIKACIJE ZA ORGANIZIRANJE VJEŽBANJA U GRUPI

Liker, Fran Urban

Undergraduate thesis / Završni rad

2023

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Algebra University College / Visoko učilište Algebra**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:225:584302>

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

Download date / Datum preuzimanja: **2024-12-21**



Repository / Repozitorij:

[Algebra University - Repository of Algebra University](#)



VISOKO UČILIŠTE ALGEBRA

ZAVRŠNI RAD

**RAZVOJ APLIKACIJE ZA ORGANIZIRANJE
VJEŽBANJA U GRUPI**

Fran Urban Liker

Zagreb, veljača 2023.

Prilikom uvezivanja rada, Umjesto ove stranice ne zaboravite umetnuti original potvrde o prihvaćanju teme završnog rada kojeg ste preuzeli u studentskoj referadi

Sažetak

U ovom završnom radu se opisuje izrada aplikacije za organiziranje vježbanja u grupi. Na osnovi ankete provedene od polagatelja i istraživanja će se proučavati izrada aplikacije koja pomaže ljudima da zajedno vježbaju. Kroz izradu aplikacije se iskazalo više tehnologija među kojima su *Flutter* i *C#*. Proći će se kroz slojeve izrade aplikacije, opisivanje implementiranog rješenja i svih tehnologija koje su korištene u izradi aplikacije. Dodatne tehnologije uključuju funkciju za derivaciju ključa bazirana na lozinki, mrežne utičnice, Hipertekstualni transferni protokol itd. Zaključno se izvlače podatci iz čitavog procesa i formira kritična misao o vježbi i korisnosti aplikacije za individue u budućnosti.

Ključne riječi: *flutter*, derivacija, protokol.

Summary

This final paper describes the creation of an application for organizing exercise in a group. Based on a survey conducted by the scholar and studies, the creation of an application that helps people to exercise together will be studied. Through the development of the application, a handfull of technologies will be demonstrated, including Flutter and C#. The analysis will go through layers of application development, describing the implemented solution and all the technologies that were used in the development of the application. Additional technologies include a password-based key derivation function,, web sockets, hypertext transfer protocol, etc. In the conclusions section, data is extracted from the entire process and forms a critical thought about the exercise and usefulness of the application for individuals in the future.

Key words: *flutter*, derivation, protocol.

Sadržaj

1. Uvod	1
2. Pregled postojećih rješenja	2
3. Analiza provedene <i>Google Forms</i> ankete koja analizira zainteresiranost i fizičko stanje ispitanika.....	5
3.1. Cilj	5
3.2. Metodologija.....	5
3.3. Anketni upitnik	6
3.4. Analiza rezultata	13
4. Arhitektura implementiranog rješenja	22
4.1. Sloj prezentacijske logike	22
4.2. Sloj poslovne logike	26
4.3. Sloj pristupa bazi podataka.....	33
4.4. Sigurnosni aspekti aplikacije	37
5. Korištene tehnologije.....	39
6. Opis implementiranog rješenja.....	39
6.1. Registracija i Prijava.....	39
6.2. Upravljanje postavkama	42
6.3. Korištenje glavne stranice i stranice za pretraživanje	44
6.4. Korištenje sobe	45
6.5. Stranica za razgovor	46
Zaključak	49
Literatura	Pogreška! Knjižna oznaka nije definirana.
Popis slika.....	54

Popis kodova **Pogreška! Knjižna oznaka nije definirana.**

1. Uvod

Živi se u svijetu koji se odaje socijalnim medijima, nezdravoj hranom, video igricama i manjkom želje za samopoboljšanjem. Iako je to realnost današnjice, to je također promjenjiv skup parametara koji počinje s zdravim tijelom i mrežom prijatelja koji te podržavaju u tvojim porivima. Zato je izabran ovaj tip aplikacije za ovaj završni rad. Kako bi se osiguralo mjesto gdje se ljudi bilo kojeg tipa tijela mogu povezati sa sličnima i postići svoje ciljeve u vježbanju. To se omogućilo tako što se kreirao skup soba potkrijepljenih *lock-in* sistemom gdje se osoba može pridružiti, pripremiti i, na kraju, prisustvovati konverzaciji u chat room-u koji je pretvoren iz te iste sobe. Sve to dolazi ukomponirano u funkcionalnu aplikaciju za organizaciju vježbe u grupi. Kroz završni rad se opisuje izrada dotične aplikacije, implementacija svih tehnologija koje idu uz nju, istraživanje sveučilišta u Wisconsin-u, istraživanje Norveškog Centra za Istraživanje Ovisnosti u Oslu, anketa provedena od autora, slojeve aplikacije. U poglavlju „Pregled postojećih rješenja“ će postojeća rješenja biti opisana i uspoređena s aplikacijom prezentiranom u ovom završnom radu. U poglavlju „

Analiza provedene *Google Forms* ankete koja analizira zainteresiranost i fizičko stanje ispitanika“ će se analizirati anketa provedena od autora kako bi se поближе upoznala situacija života ispitanika. U poglavlju „

Arhitektura implementiranog rješenja“ će se proći slojevi implementiranog rješenja. U poglavlju „Korištene tehnologije“ će se opisivati tehnologije korištene za implementaciju rješenja. U poglavlju „Opis implementiranog rješenja“ će se proći po svim zaslonima i objasniti kako implementirano rješenje radi.

2. Pregled postojećih rješenja

Postoji više rješenja sličnih aplikaciji koja se prezentira u ovom završnom radu. Iako jesu slični niti jedan primjer nije potpuno isti što čini aplikaciju jedinstvenom među sličnima. Pošto aplikacija ima socijalan aspekt u svojoj implementaciji, njezin je doprinos jedinstven u usporedbi s ostalim aplikacijama za vježbu i *fitness*:

- A) *Strava* je društveni medij za sve koji se bave bilo kojom vrstom vježbi. U *Stravi* se mogu mjeriti pretrčani kilometri, provjeriti putanje trčanja, dijeliti postignuti rezultati, odrediti tip vježbe kao naprimjer; vožnja bicikla, trčanje, rolanje... Kao sama aplikacija, *Strava* ne potiče sastajanje uživo kao aplikacija koja je prezentirana ovim završnim radom. Naprotiv *Strava* imitira ostale društvene medije po svojoj konstrukciji. Ta konstrukcija je sljedeća: Netko slijedi svog prijatelja ili poznatog utjecajnog individualca, odlazi na glavnu stranicu koja ima niz preporučenih postova koji sadrže rezultate vježba koje su odradili ljudi koje je taj netko slijedio. Glavna stvar kod takvog pristup jest da slično potiče individualca da pregledava kroz postignuća i uspoređuje se s drugim vježbačima koji imaju i bolje rezultate. Aplikacija ima i prozor za mjerenje napredovanja, grupe za izazove, karte s najoptrčanijim rutama i stranica sa svojim statistikama.
- B) *JeFit* je savršeno rješenje za sve koji žele vježbati kod kuće ili individualno. To je analitička aplikacija koja prikazuje sve statistike vježbe individualca. Dijeli se na *desktop* verziju, mobilnu aplikaciju i aplikaciju za pametan sat. U glavnoj stranici *desktop* verzije se vide statistike kao što su: omjer vježbi poredani po dijelovima tijela, ukupna težina koja se podizala kroz vrijeme, vrijeme svake sesije i apsolutni zbroj svih vremenskih statistika. Osim glavne stranice je moguće pristupiti i svojim bilješkama, svojim izvještajima, svojim slikama napretka, svojim modificiranim vježbama, svojim porukama, elitnoj podršci i svojim prijateljima. Generalno gledajući, *JeFit* ima sve potrebno za individualnu vježbu i povezivanje s drugim vježbačima na mreži, no isto kao i u prvom postojećem rješenju, nema poticaj da se ti prijatelji susretnu u svrhu grupnog vježbanja. Dotično se rješenje može uspoređivati s aplikacijom koja se prezentira u ovom završnom radu iz razloga što je

puna implementiranih funkcija koje pomažu individualcu da se razvija kroz zdravo tijelo, što je na kraju cilj obje aplikacije. Za razliku od aplikacije koja se prezentira u ovom završnom radu *JeFit* ima puno više implementacija, puno više opcija i puno više funkcionalnosti što ga čini naprednijim i oblikovanijim.

- C) *FitBit* je općenita aplikacija za mjerenje skoro svih aspekata života koji se vrti pri treningu. Od ciklusa spavanja do prijeđenih koraka, ova aplikacija nudi sve što je potrebno za zdrav život i vježbanje. Poput *Strave* i *JeFita*, aplikacija ima svoju zajednicu s mogućnošću objavljivanja objava na javan zid interakcije koji se mijenja na osnovi odabranih prijatelja i interesa. Aplikacija uključuje i zabavne igre vježbanja kao što je „Vježbački bingo“ gdje birate vježbe umjesto brojeva kako bi prvi pogodili ispravan niz vježbi i pobijedili. Takav je pristup vježbanju inovativan i zabavan, no dolazi uz svoju cijenu jer je zaključan iza *premium* članstva. Aplikacija dolazi uz svoj osobni *FitBit* pametni sat koji prati sve statistike vašeg tijela i na taj način omogućuje validan prikaz svih statistika. Iako su među skupljim modelima satovi imaju instaliranu podršku za aplikacije 3. strane. Na taj način je moguće pristupiti aplikacijama kao što su *Whatsapp*, *Strava*, *WeightWachers* itd. Gledajući ukupno *FitBit* je aplikacija s raznovrsnim implementacijama i vrlo je opširna, no još uvijek ne potiče grupno vježbanje kao aplikacija koja je prezentirana u ovom završnom radu. Dok se *FitBit* bazira na mjerenju tjelesnih statistika i opširnom mjerenju, aplikacija prezentirana u ovom završnom radu se bazira na povezivanju ljudi kroz sobe za zaključavanje i razgovor te mjerenju koje je primitivnije od *FitBit* mjerenja.
- D) *Stacked* rješenje je aplikacija za praćenje vježbi, s posebnim fokusom na dizanje utega. *Stacked* nudi široki raspon vježbi koje se mogu izvesti s utezima i ostalim spravama za vježbanje. Implementacije su vrlo specijalizirane za mjerenje setova pri vježbanju. To znači da je svaki set od određenog broja ponavljanja registriran pri dodiru zaslona nakon odrađivanja seta. Rješenje je mehaničko, ali je bolje od zasad nepouzdanog računalnog vida koji bi nepouzdanio odredio broj izvršenih vježbi. *Stacked* je, kao što je bio demonstrirano u prijašnjem paragrafu, vrlo specijalizirana aplikacija koja je zadužena za mjerenje setova vježbi. To je malo drugačije od prijašnjih primjera jer se ne fokusira na opširnu kvantitetu implementacija, već na kvalitetu jednog tipa implementacije. Niti ova aplikacija ne potiče vježbanje u grupi kao aplikacija prezentirana u ovom završnom radu.

Analiza u sljedećem poglavlju će se pokazati analiza *Google Forms* ankete provedene od autora.

3. Analiza provedene *Google Forms* ankete koja analizira zainteresiranost i fizičko stanje ispitanika

U analizi provedene *Google Forms* ankete će se proći kroz točno određene smjernice navedene u nastavku. U cilju će se obrađivati ciljevi ankete i objasniti što oni zapravo jesu. U metodologiji će se obrađivati znanost o metodi korištenoj u ovoj anketi. U anketnom upitniku će se obraditi sve što upitnik sadržava i biti će potkrijepljeno slikama iz same ankete kako bi se ustanovilo prisustvo svih odgovora. Analiza rezultata će dovesti na svijetlo sintezu obrađenih podataka i time oplemeniti ovu anketu odgovorom na pitanja.

3.1. Cilj

Cilj provedene *Google Forms* ankete jest ispitati što više ljudi o fizičkim karakteristikama, mišljenju, aktivnim hobijima, interesu, spolu i ciljevima. Fizičke karakteristike podrazumijevaju samo tip tijela, što je vrlo relevantno za aplikaciju koja se prezentira u ovom završnom radu, iz razloga što se u samoj aplikaciji pojavljuju tipovi tijela kao indikator sličnosti s ostalim sudionicima u aplikaciji. Aktivni hobiji podrazumijevaju fizičke aktivnosti koje ispitanik provodi, bilo da se radi o vremenu u teretani, borilačkom klubu ili u nekoj drugoj vanjskoj aktivnosti. Interes podrazumijeva interes za aplikaciju, to jest ocjenu opće zainteresiranosti od „Nisam uopće zainteresiran/a“ do „Izrazito sam zainteresiran/a“. Spol podrazumijeva izjašnjava li se osoba kao muško, žensko ili nešto treće. Ciljevi ispitanika podrazumijevaju ciljeve u njihovom putovanju kroz fitness, primjerice: smršaviti, estetski oblikovati tijelo, postići fleksibilnost, itd. Svi ovi parametri reprezentiraju cilj ankete koji je bio prikupiti što više informacija od što više ispitanika kako bi se utvrdila potreba za aplikacijom i njezinim funkcionalnostima, realističnost opcija u aplikaciji te ciljevi individualaca koji bi koristili aplikaciju.

3.2. Metodologija

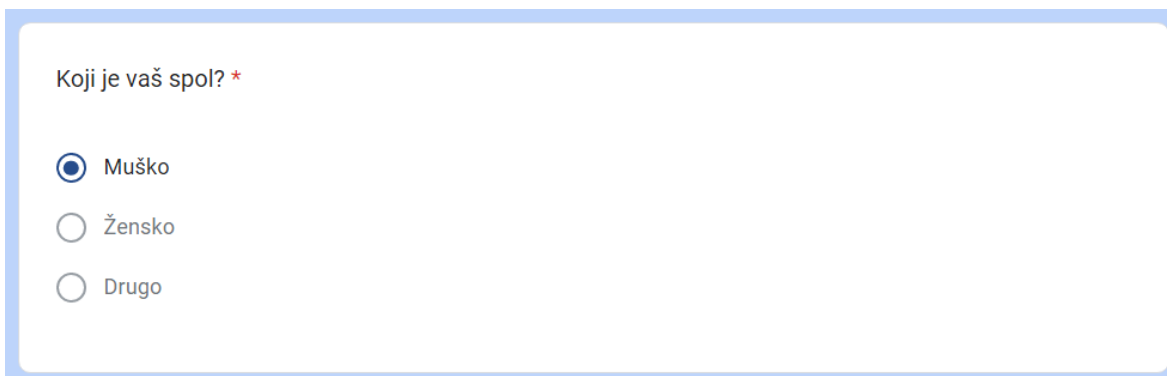
Metodologija korištena u ispitivanju je metodologija pet odgovora pri čemu odgovori konvergiraju najboljem odgovoru. Primjerice u pitanju „Koliko ste zainteresirani za

izvođenje vaših aktivnosti u grupi?“ odgovori na raspolaganju su: “Izrazito sam nezainteresiran/a”, “Lagano sam nezainteresiran/a”, “niti jedno niti drugo”, “Lagano sam zainteresiran”, “Izrazito sam zainteresiran/a” te redosljedom od najgoreg mogućeg do najboljeg. Ta metodologija je učinkovita iz razloga što se može lakše analizirati razine zadovoljstva ili nezadovoljstva individualca koji je ispitan za razliku od binarnog odgovora kao što su da ili ne. Također, ta metodologija nije nužno i isključivo primijenjena na ovu formu iz razloga što su se u formi nalazila pitanja koja nisu analizirala zainteresiranost ili neku drugu funkciju, već parametre izvan. To su uglavnom hobiji i ciljevi individualca koji je u ispitivanju. Dodatno, metodologija koja se koristila u svrhu ispitivanja je najbliža metodi analize i sinteze u kojoj se podatci analiziraju te se od njih sintezom tvore misaone skupine. Pod tim mislim na stvaranje informacije od podataka koje je doprinijela ova anketa. To se radi na način da se analiziraju podatci i pogledaju najizraženiji postotci ankete te se iz toga vodi zaključak o interesu, izboru hobija ili cilju individualca kojeg se ispituje.

3.3. Anketni upitnik

Anketni upitnik se sastoji od jedanaest pitanja koja ispituju mišljenja, ciljeve, fizičke karakteristike i hobije individualca koji su pristupili istraživanju. Neki su se upiti u anketi razlikovali za muškarce i žene pa su zato bili označeni s (za žene) i (za muškarce) oznakama prije samog pitanja. U sljedećim će se paragrafima proći sva pitanja iz ankete.

Prvo pitanje na slici Slika 3.1 Slika prvog pitanja ankete se odnosilo na spol. Ispitanik bira izjašnjava li se kao muško, žensko ili drugo.



Koji je vaš spol? *

Muško

Žensko

Drugo

Slika 3.1 Slika prvog pitanja ankete

Drugo pitanje na slici Slika 3.2 Slika drugog pitanja ankete se odnosilo na to koliko ispitanik vježba. Ponudeni odgovori su:

- „Ne vježbam uopće“
- „Svakih mjesec dana“
- „Svakih tjedan dana“
- „Svaki dan s odmorima“
- „Svaki dan bez odmora“

Koliko redovito vježbate? *

Ne vježbam uopće

Svakih mjesec dana

Svakih tjedan dana

Svaki dan s danima odmorima

Svaki dan bez dana odmora

Slika 3.2 Slika drugog pitanja ankete

Treće pitanje na slici Slika 3.3 Slika trećeg pitanja ankete se odnosi na to da li ispitanik vježba sam ili u grupi. Ponuđeni odgovori su:

- Sam
- U grupi
- Ne vježbam

Vježbate li sami ili u grupi?

Sam

U grupi

Ne vježbam

Slika 3.3 Slika trećeg pitanja ankete

Četvrto pitanje na slici Slika 3.4 Slika četvrtog pitanja ankete se odnosi na to kako se ispitanik osjeća po motiviranosti. Također se mora napomenuti da je ovo pitanje samo za one koji ne vježbaju ili vježbaju sami. Odgovori koji su ponuđeni su:

- Izrazito sam demotiviran
- Malo sam demotiviran
- Niti jedno niti drugo
- Malo sam motiviran
- Izrazito sam motiviran

(Ako vježbate sami ili ne vježbate) Kako se osjećate po motiviranosti?

Izrazito sam demotiviran

Lagano sam demotiviran

Niti jedno niti drugo

Lagano sam motiviran

Izrazito sam motiviran

Slika 3.4 Slika četvrtog pitanja ankete

Peto pitanje na slici Slika 3.5 Slika petog pitanja ankete se odnosi motivaciju ispitanik u slučaju da nađe društvo za vježbu. Mora se napomenuti da ovo pitanje na neki način navodi na odgovor koji je pozitivan pošto su ljudi koji su ispitani neinformirani o istraživanjima efekata [1] *Quality of life* bodova (skraćeno QOL). Ovo pitanje je i za one koji vježbaju sami ili ne vježbaju uopće. Ponuđeni odgovori su:

- Ne bi me motiviralo
- Malo bi me motiviralo
- Izrazito bi me motiviralo
- Ništa od navedenog

(Ako vježbate sami ili ne vježbate) Bi li vas mogućnost pronalaženja društva za vježbu motiviralo da vježbate više?

- Ne bi me motiviralo
- Malo bi me motiviralo
- Izrazito bi me motiviralo
- Ništa od navedenog

Slika 3.5 Slika petog pitanja ankete

Šesto pitanje na slici Slika 3.6 Slika šestog pitanja ankete se odnosi na primarne aktivnosti anketiranog ispitanika. Ovo pitanje ima više mogućih kombinacija odgovora; ispitanik može odabrati skup odgovora umjesto samo jednog. Odgovori koji su ponuđeni su:

- Dizanje utega
- Aerobik (bilo koja vrsta)
- Vanjske aktivnosti (vožnja bicikla, trčanje, i sl.)
- Joga
- Borilačka aktivnost
- Plivanje
- Natjecateljski sportovi (košarka, nogomet, hokej na ledu, itd.)

Koje su vaše primarne aktivnost? *

- Dizanje utega
- Aerobik (bilo koja vrsta)
- Vanjske aktivnosti (vožnja bicikla, trčanje...)
- Joga
- Borilačka aktivnost
- Plivanje
- Natjecateljski sportovi (Košarka, Nogomet, Hokej na ledu...)
- Nemam primarnu aktivnost

Slika 3.6 Slika šestog pitanja ankete

Sedmo pitanje na slici Slika 3.7 Slika sedmog pitanja ankete se odnosi na zainteresiranost ispitanika za vježbanje u grupi ako pritom vježbaju sami. Ponuđeni odgovori su:

- Izrazito sam nezainteresiran
- Lagano sam nezainteresiran
- Niti jedno niti drugo
- Lagano sam zainteresiran
- Izrazito sam zainteresiran

(Ako vježbate sami) Koliko ste zainteresirani za izvođenje vaših aktivnosti u grupi?

- Izrazito sam nezainteresiran
- Lagano sam nezainteresiran
- Niti jedno niti drugo
- Lagano sam zainteresiran
- Izrazito sam zainteresiran

Slika 3.7 Slika sedmog pitanja ankete

Osmo pitanje na slici Slika 3.8 Slika osmog pitanja ankete se odnosi na specifičan cilj koji ispitanik ima bez obzira na to da li vježba ili ne vježba. Ponuđeni odgovori su:

- Izgubiti na težini
- Dobiti na težini (*bulk* ili mišićna masa)
- Oblikovati tijelo estetički
- Pospješiti elastičnost
- Pospješiti izdržljivost
- Samoobrana
- Nemam specifični cilj

Koji je vaš trenutni cilj? *

Izgubiti na težini

Dobiti na težini (bulk ili mišićna masa)

Oblikovati tijelo estetički

Pospješiti vašu elastičnost

Pospješiti izdržljivost

Samoobrana

Nemam specifičan cilj

Slika 3.8 Slika osmog pitanja ankete

Deveto pitanje na slici Slika 3.9 Slika devetog pitanja ankete se odnosi na tip tijela ako je ispitanik muško. U ovom pitanju je u privitku dodana i internetska stranica koja objašnjava svaki tip tijela i obrazuje ispitanika na temu odabira tipa tijela. Ponuđeni odgovori su:

- Ektomorf (eng. *Ectomorph*)
- Mezomorf (eng. *Mezomorph*)
- Endomorf (eng. *Endomorph*)

(Za muškarce) Koji ste tip tijela po stranici u privitku?
<https://www.medicalnewstoday.com/articles/body-types>

- Ektomorf
- Mezomorf
- Endomorf

Slika 3.9 Slika devetog pitanja ankete

Deseto pitanje na slici Slika 3.9 Slika devetog pitanja ankete se odnosi na tip tijela ukoliko je ispitanik ženskog spola. U privitku je također poslana ista internetska stranica. Ponuđeni odgovori su:

- Trokut (eng. *Triangle*)
- Invertirani trokut (eng. *Inverted triangle*)
- Pravokutnik (eng. *Rectangle*)
- Ovalni (eng. *Oval*)
- Pješčani sat (eng. *Hour glass*)

(Za žene) Koji ste tip tijela po stranici u privitku?
<https://www.medicalnewstoday.com/articles/body-types>

- Trokut (Triangle)
- Invertirani trokut (Inverted Triangle)
- Pravokutnik (Rectangle)
- Ovalni (Oval)
- Pješčani sat (Hour Glass)

Slika 3.10 Slika desetog pitanja ankete

Jedanaesto pitanje na slici Slika 3.11 Slika jedanaestog pitanja ankete je osobno pitanje o interesu ispitanika za samu aplikaciju za spajanje pojedinaca u svrhu vježbe u grupi. Ponuđeni odgovori su:

- Izrazito sam nezainteresiran
- Lagano sam nezainteresiran

- Niti jedno niti drugo
- Lagano sam zainteresiran
- Izrazito sam zainteresiran

Koliko ste zainteresirani za aplikaciju koja bi grupirala pojedince u svrhu grupnog vježbanja? *

Izrazito sam nezainteresiran

Lagano sam nezainteresiran

Niti jedno niti drugo

Lagano sam zainteresiran

Izrazito sam zainteresiran

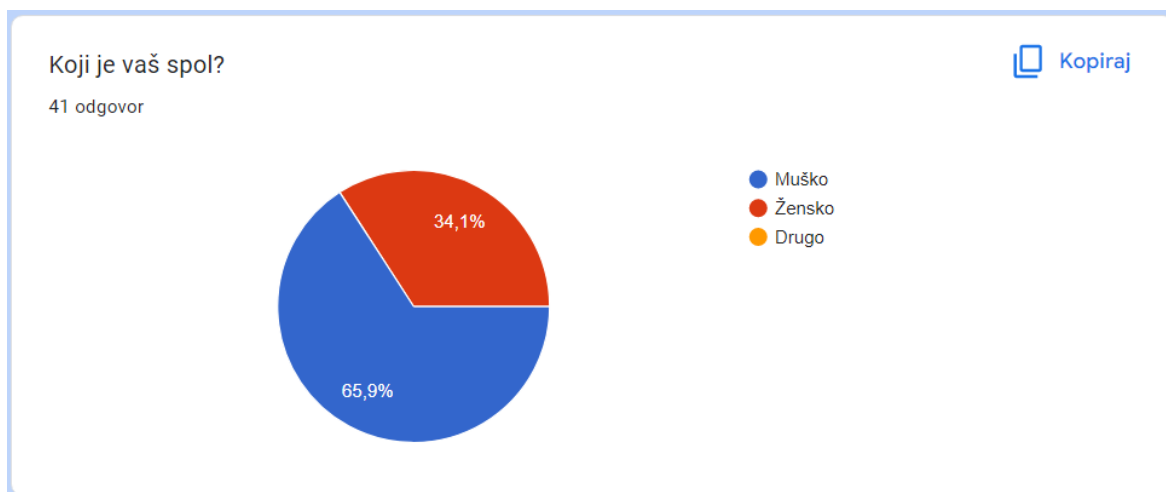
Slika 3.11 Slika jedanaestog pitanja ankete

U sljedećem će se potpoglavlju analizirati rezultati te pomoću analize i sinteze tih podataka dobiti realistična slika onoga što ljudi trebaju i žele.

3.4. Analiza rezultata

Nakon što je izrađena anketa puštena u javnost, četrdeset i jedan ispitanik je popunio anketu do kraja. Tu se mora napomenuti tolerancija za grešku koja je uzrokovana preskakanjem pitanja u nekim slučajevima. Na primjer, preskočilo se pitanje tipa tijela za muškarce i žene i na taj način se ukupan odgovor u tom segmentu ima nekonzistentan broj odgovora. Zato anketa ima toleranciju na grešku za sad nepoznatu.

Ako se analizira prvo pitanje na slici Slika 3.12 Analiza prvog pitanja ankete koje je bilo „Koji je vaš spol?“, dobit će se rezultat seta podataka gdje je 65,9% ispitanika muško, a 34,1% ispitanika žensko. Niti jedan ispitanik nije odgovorio „Drugo“ što znači da nema pripadnika nebinarne grupe individualaca. Ovi nam podatci ne govore ništa pošto vježba nije ograničena po spolu, ali će biti važni za kontinuitet analize ostalih anketnih pitanja.

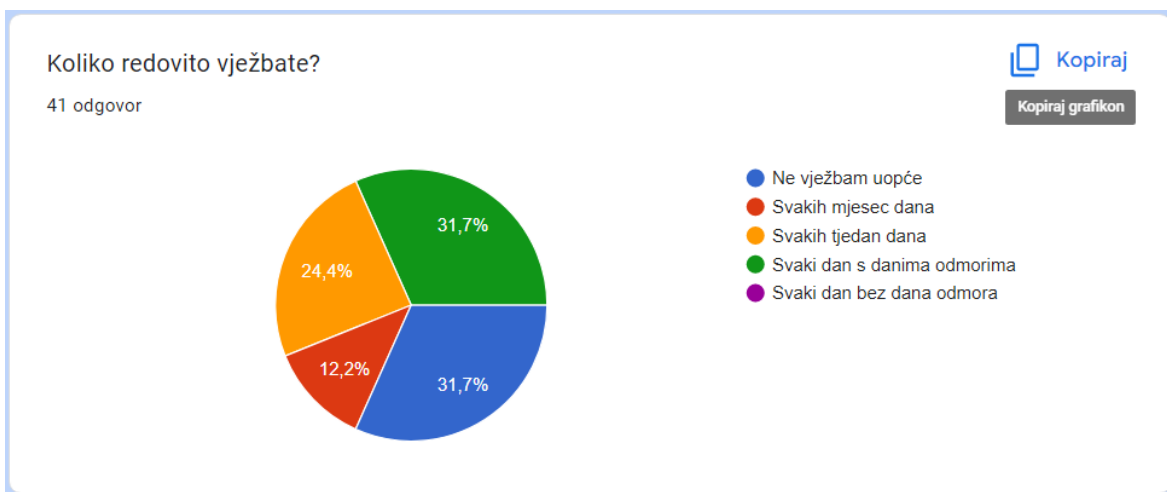


Slika 3.12 Analiza prvog pitanja ankete

Kada se analizira drugo pitanje ankete na slici Slika 3.13 Analiza drugog pitanja ankete koje je primijenjeno slikom Slika 3.1 Slika prvog pitanja ankete dobivaju se podatci koji govore o redovitosti vježbanja ispitanika. Podatci pokazuju da od 41 ispitanika 31,7% vježba svaki dan s danima odmora. Isto tako 31,7% ispitanika ne vježba uopće. 24,4% ispitanika vježba svakih tjedana dana, dok 12,2% ispitanika vježba svakih mjesec dana. Ako se pretpostavi da redoviti vježbači vježbaju pet dana u tjednu s danima odmora u subotu i nedjelju te ako se pretpostavi da neredoviti vježbači, što podrazumijeva one koji vježbaju jednom tjedno i jednom mjesečno, vježbaju točno jedan dan u tjednu ili mjesecu, tada se može izračunati postotak prosjeka dana koji svi ispitanici provedu u vježbi. Pretpostavlja se, također, da svi provode istu količinu vremena u vježbi. Nakon svih ovih pretpostavki se može izračunati prosjek dana u godini provedenih na vježbanje za sve ispitanike po ovoj formuli:

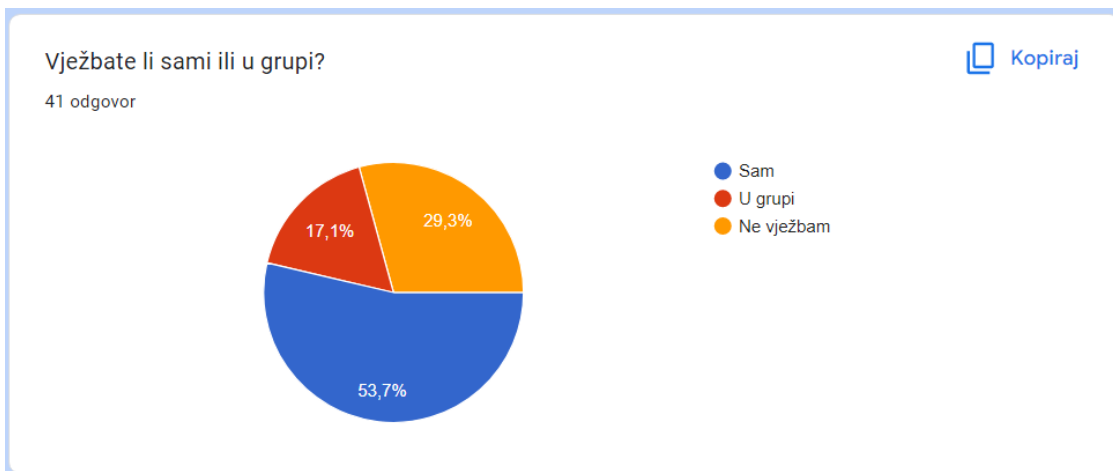
$$a = \frac{x_1 \cdots x_n}{n}$$

U ovoj formuli varijabla x predstavlja dane u godini za svakog ispitanika ovisno o broju pod varijablama x . Varijabla n predstavlja broj ispitanika. Kako bi izračunali prosjek dana u godini vježbanih od svih ispitanika treba se zbrojiti sve dane na koje su ispitanici vježbali i podijeliti ih s brojem ispitanika. Kada se formula primjeni dobije se rezultat od 96.59 dana provedenih u vježbi, što je samo 26.46% dana kada se vježbalo. Kada bi gledali realističnu sliku i uzeli u obzir da većina redovitih ispitanika vježba tri dana u tjednu umjesto pet dana u tjednu, statistika je još gora. No taj skup parametara koji se odnose na vježbanje se može promijeniti s daljnjom promocijom aplikacija za vježbu bilo koje vrste, a ne samo one koja je prezentirana u ovom završnom radu.



Slika 3.13 Analiza drugog pitanja ankete

Kada se analizira treće pitanje ankete na slici Slika 3.14 Analiza trećeg pitanja ankete primijenjeno slikom 3.3. Slika 3.3 Slika trećeg pitanja ankete dobivaju se podatci o tome vježbaju li ispitanici sami ili u grupi. I rezultat nije iznenađujuć. 53,7% ispitanika vježba sama, dok samo 17,1% ispitanika vježba u grupi. Ovo pokazuje da je vježbanje još uvijek individualan sport što je velika šteta iz razloga što vježba u grupi nudi veliki broj načina da se poboljša kvaliteta života.



Slika 3.14 Analiza trećeg pitanja ankete

Kada se analizira četvrto pitanje ankete na slici Slika 3.15 Analiza četvrtog pitanja ankete primijenjeno slikom Slika 3.4 Slika četvrtog pitanja ankete dobivaju se podatci o osjećajima motiviranosti ispitanika koji vježbaju sami ili ne vježbaju. Podatci su sljedeći: 16,2% ispitanika se osjeća izrazito motivirano, 32,4% ispitanika se osjeća lagano motivirano, 24,3% je neutralno, 8,1% je lagano demotivirano i 18,9% je izrazito demotivirano. Ovo pokazuje mali postotak osoba koje trebaju vježbu u grupi za motivaciju gledajući na to da je većina

ispitanika koji vježbaju sami ili ne vježbaju dovoljno motivirana da vježba dalje ili počne vježbati, no u petom pitanju će se pokazati da ispitanicima ipak treba grupno vježbanje zbog raznih benefita koji se doznaju po istraživanjima.



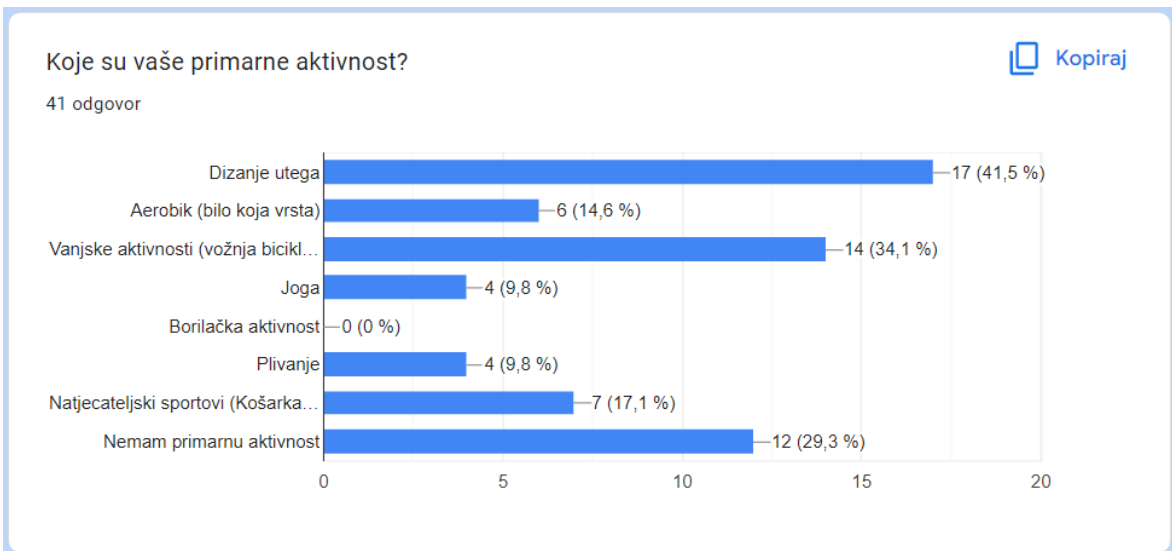
Slika 3.15 Analiza četvrtog pitanja ankete

[1] Kada se analizira peto pitanje na slici Slika 3.16 Analiza petog pitanja ankete primijenjeno slikom Slika 3.5 Slika petog pitanja ankete dobivaju se rezultati mišljenja ispitanika koji ne vježbaju ili vježbaju sami o tome da li bi mogućnost za pronalazak društva bio dobar motivator da vježbate. Podatci su sljedeći: 27% ispitanika misli da bi ih pronalazak društva izrazito motivirao, 64,9% ih misli da bi ih pronalazak društva za vježbu malo motivirao, 5,4% ih misli da ih ne bi motiviralo, dok je 2,7% (1 ispitanik) neutralan. Ovi podatci su ključni za potrebnost aplikacije koja je prezentirana u ovom završnom radu iz razloga što pokazuje da su ispitanici mišljenja da bi im vježbanje u društvu pomoglo u motivaciji da vježbaju više. Postoji i dokaz za takvu hipotezu. [2] U istraživanju provedenom od Thomasa Clausena i Elizabeth Muller, grupna vježba, iako ne pomaže u socijalnim odnosima statistički, samo vrijeme je imalo utjecaj na QOL psihološkog zdravlja dok su vrijeme i vježba imale veliki utjecaj na QOL fizičkog zdravlja. Vježba je također snizila anksioznost sa 74% na 46% i depresiju sa 78% na 36%. Iako nije bilo statističkog dokaza da grupno vježbanje pomaže u socijalnim odnosima, klinička zapažanja su indicirala na poboljšanje u socijalnom krugu i motivaciju za ponovnim dolaženjem na trening kao i obavljanje dnevnih zadaća.



Slika 3.16 Analiza petog pitanja ankete

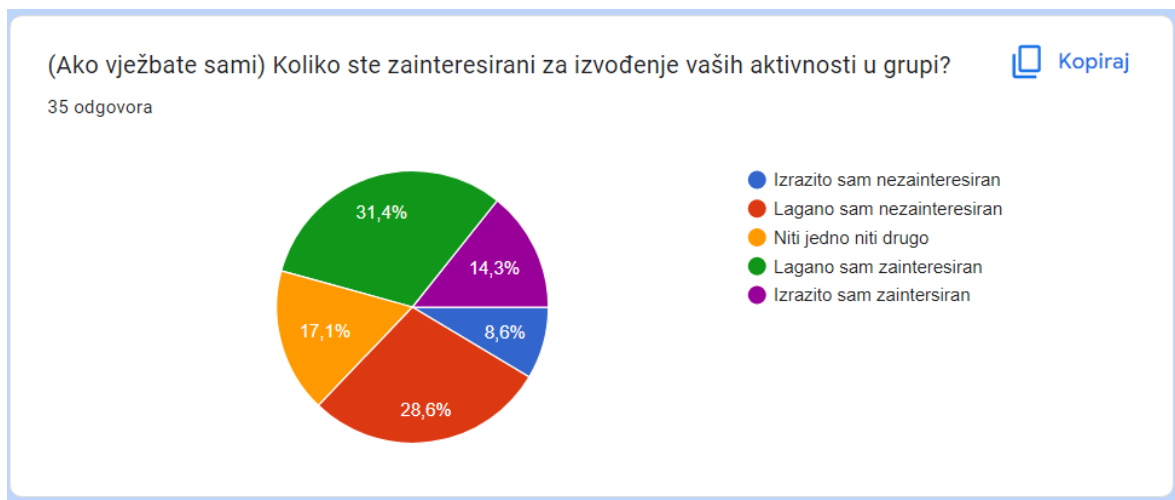
Kada se analizira šesto pitanje na slici Slika 3.17 Analiza šestog pitanja ankete koje je primijenjeno slikom Slika 3.6 Slika šestog pitanja ankete dobivaju se podatci o primarnim aktivnostima ispitanika. Podatci su sljedeći: 41,5% odgovora je bilo dizanje utega, 14,6% odgovora je bilo aerobik, 34,1% odgovora je bilo vanjske aktivnosti, 9,8% odgovora je bilo joga, 9,8% odgovora je bilo plivanje, 17,1% odgovora je bilo natjecateljski sportovi i 29,3% odgovora je bilo da ispitanici nemaju primarnu aktivnost. Analiza ovog pitanja pokazuje samo interese ispitanika i predodređuje u kojem smjeru bi se aplikacija koja se prezentira u ovom završnom radu trebala razvijati.



Slika 3.17 Analiza šestog pitanja ankete

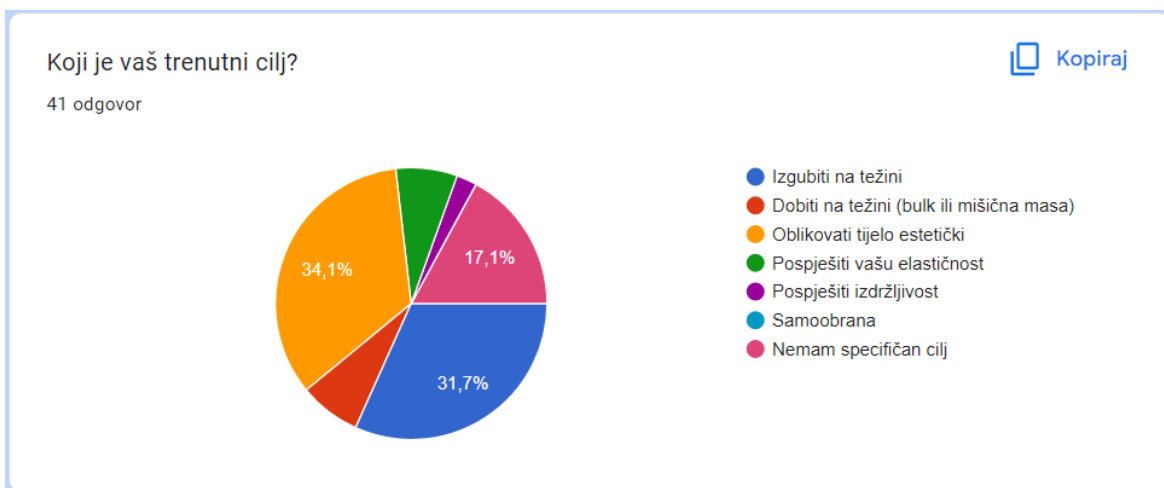
Kada se analizira sedmo pitanje na slici Slika 3.18 Analiza sedmog pitanja ankete koje je primijenjeno slikom Slika 3.7 Slika sedmog pitanja ankete dobiju se podatci interesa

ispitanika koji ne vježbaju sami o izvođenju aktivnosti, navedenih u prošlom pitanju, u grupi. Podatci su sljedeći: 14,3% ispitanika je izrazito zainteresirano, 31,4% ih je lagano zainteresirano, 17,1% ih je neutralno, 28,6% ih je lagano nezainteresirano i 8,6% ih je izrazito nezainteresirano. Ovi podatci pokazuju lagani interes za izvođenje aktivnosti navedenih u prethodnom pitanju u grupi. Ako se pobroje svi zainteresirani i nezainteresirani vidjet će se da je omjer između izrazito zainteresiranih i izrazito nezainteresiranih jednak 5:3, a omjer između malo zainteresiranih i malo nezainteresiranih je jednak 11:10. To dokazuje malu superiornost onih koji su zainteresirani za izvođenje aktivnosti u grupi.



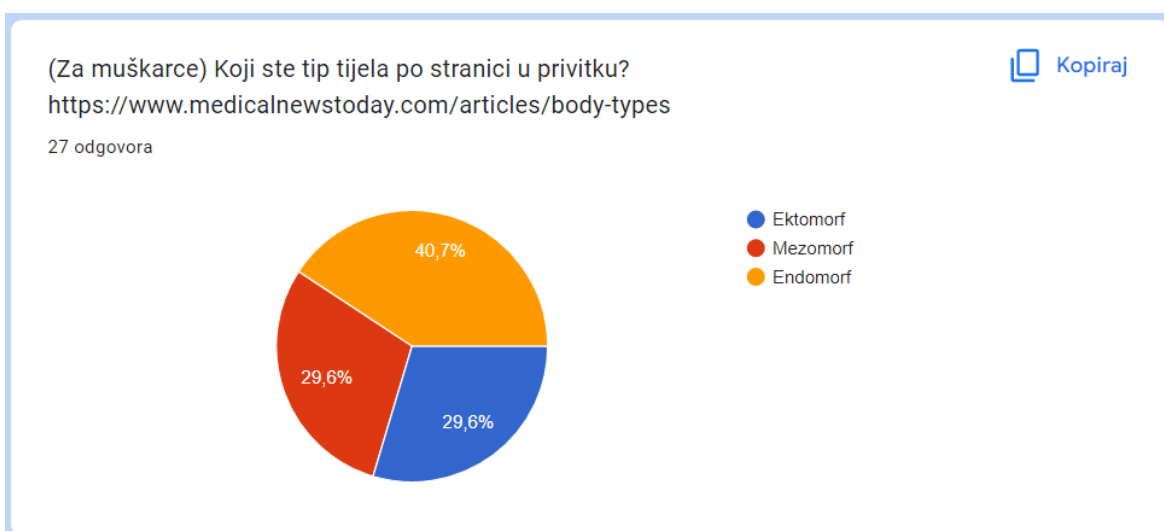
Slika 3.18 Analiza sedmog pitanja ankete

Kada se analizira osmo pitanje ankete na slici Slika 3.8 Slika osmog pitanja ankete primijenjeno slikom Slika 3.8 Slika osmog pitanja ankete dobivaju se podatci o trenutnom cilju ispitanika. Podatci su sljedeći: 31,7% ispitanika ima cilj izgubiti na težini, 7,3% ih ima cilj dobiti na težini, 34,1% ih ima cilj oblikovati tijelo estetički, 7,3% ih ima cilj pospješiti elastičnost, 2,4% ih ima cilj pospješiti izdržljivost i 17,1% ih nema specifičan cilj. Ovi podatci samo određuju smjer kamo bi se aplikacija koja je prezentirana u ovom završnom radu trebala razvijati. Ako bi se poslušala isključivo ova skupina ispitanika, bilo bi najbolje razviti aplikaciju koja je fokusirana na otporni trening i *cardio* vježbe.



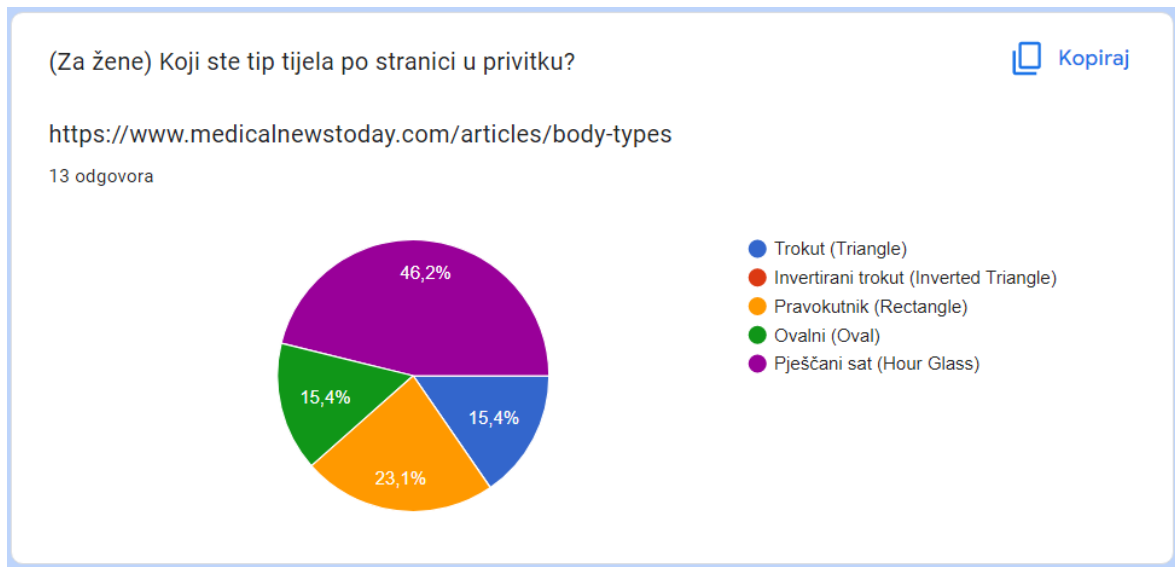
Slika 3.19 Analiza osmog pitanja ankete

Kada se analizira deveto pitanje ankete na slici Slika 3.20 Analiza devetog pitanja ankete primijenjeno slikom Slika 3.9 Slika devetog pitanja ankete dobivaju se podatci o tipu tijela ispitanika ako je ispitanik muško. Podatci su sljedeći: 29,6% ispitanika su ektomorfi što znači da imaju brz metabolizam te teško dobivaju na težini ili mišićnoj masi. 29,6% ispitanika su mezomorfi što znači da imaju srednji metabolizam te srednje dobivaju na težini ili mišićnoj masi. Oni zapravo imaju najbolje od oba svijeta, što znači da mogu lako manipulirati svojom težinom. Također 29,6% ispitanika su endomorfi. To znači da lako dobivaju na mišićnoj masi, ali im je teško to iskazati zbog njihovog sporog metabolizma i velike figure. Ovi nam podatci govore kakve tipove tijela muškarci imaju što pokazuju koliko će biti teško postići ciljeve u prethodnom pitanju.



Slika 3.20 Analiza devetog pitanja ankete

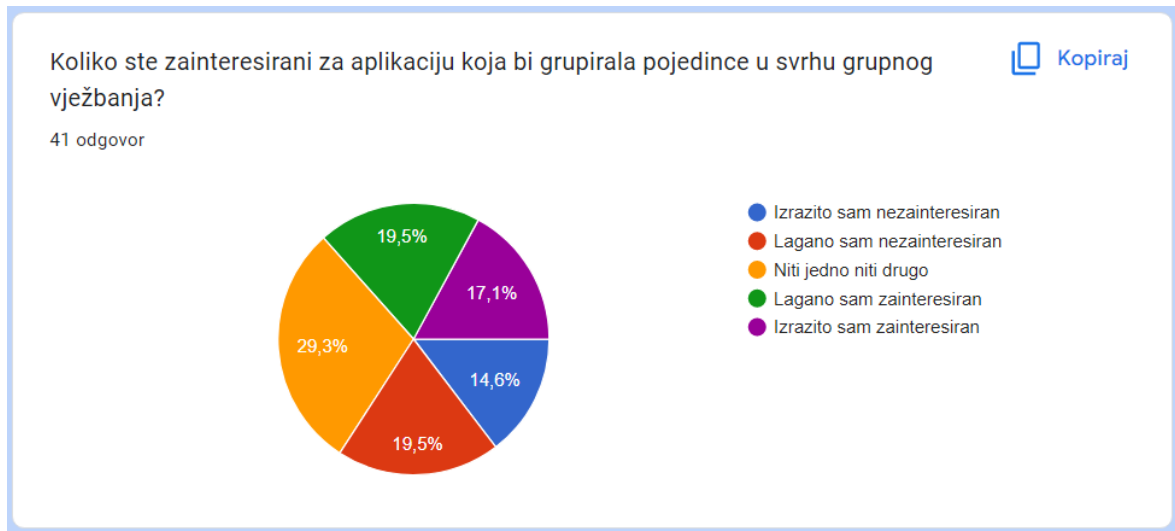
Kada se analizira jedanaesto pitanje ankete primijenjeno slikom Slika 3.11 Slika jedanaestog pitanja ankete dobivaju se podatci o tipu tijela ispitanika ako je ispitanik žensko. Podatci su sljedeći: 46,2% ispitanika su oblika pješčanog sata, 15,4% ispitanika su ovalnog oblika, 23,1% ispitanika su pravokutnog oblika i 15,4% ispitanika su trokutnog oblika. Iako i žene isto imaju tipove tijela navedene u pitanju za muškarce, zbog njihove građe se navodi više oblika tijela kako bi razaznali više tipova tijela ne vezanih samo za metabolizam.



Slika 3.21 Analiza desetog pitanja

Kada se analizira jedanaesto pitanje ankete na slici Slika 3.22 Analiza jedanaestog pitanja ankete primijenjeno slikom Slika 3.11 Slika jedanaestog pitanja ankete dobivaju se podatci o zainteresiranosti ispitanika za aplikaciju prezentiranu u ovom završnom radu. Podatci su sljedeći: 14,6% ispitanika je izrazito nezainteresirano, 19,5% ih je lagano nezainteresirano, 29,3% ih je neutralno, 19,5% ih je lagano zainteresirano i 17,1% ih je izrazito zainteresirano. Ovi podatci su jako miješani što pokazuje da je i interes dosta neutralan što se tiče ukupne ocjene podataka. Ako se napravi omjer između izrazito zainteresiranih i izrazito nezainteresiranih, dobit će se omjer 7:6, dok je broj malo zainteresiranih i malo

nezainteresiranih jednak. To daje malu prednost interesu za aplikaciju koja se prezentira u ovom završnom radu, iako je broj neutralnih najveći u ovom ovalno grafu.



Slika 3.22 Analiza jedanaestog pitanja ankete

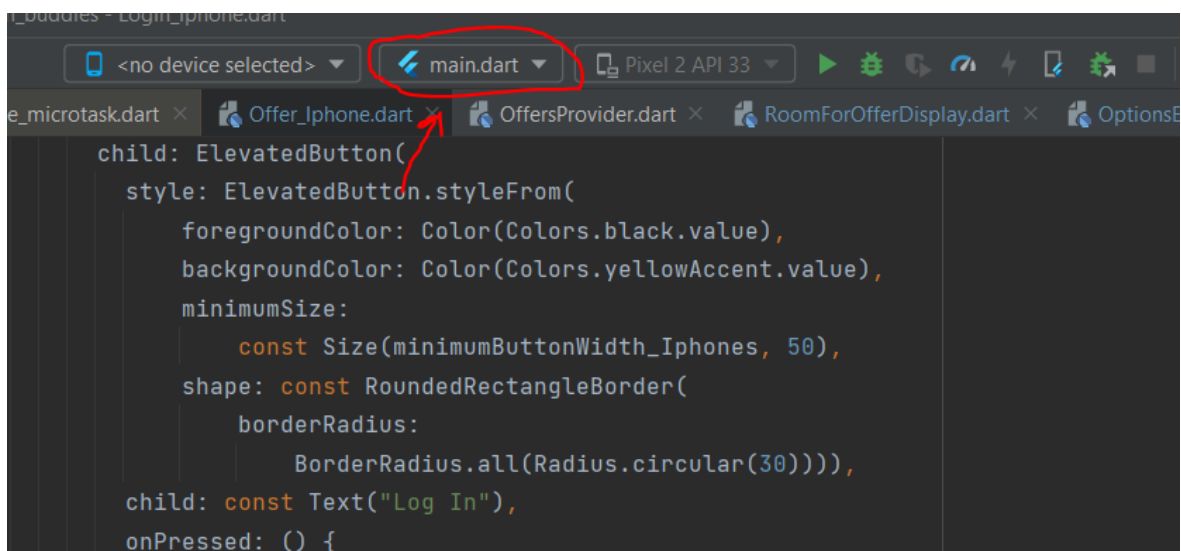
4. Arhitektura implementiranog rješenja

Arhitektura aplikacije koja se prezentira u ovom završnom radu je raznolika i obostrana. Proteže se kroz grafičko korisničko sučelje do zadnjeg djela iza sučelja koji sadrži bazu podataka i sučelje za prijenos podataka u bazu i iz nje. U sljedećim koracima će se proći kroz slojeve arhitekture implementiranog rješenja. Potpoglavlja su ujedno i slojevi arhitekture, a nazivaju se: Sloj prezentacijske logike, Sloj poslovne logike, Sloj pristupa bazi podataka, Sigurnosni aspekti baze podataka.

4.1. Sloj prezentacijske logike

U sloju prezentacijske logike će se prezentirati grafičko korisničko sučelje. U ovom slučaju je to sučelje unutar programskog jezika *Flutter*. Flutter je SDK (engl. System Development Kit) koji koristi dart programski jezik Iako je dio poslovne logike također unutar tog programskog jezika, njega ćemo proći u analizi poslovnog sloja aplikacije.

Struktura prezentacijske logike počinje od funkcije `main` koja se pokreće pomoću android studio IDE-a (kratica za engleski naziv *Integrated Development Environment*) odabiranjem datoteke koja je glavni pokretač na gornjoj traci iznad mjesta za kodiranje.



Slika 4.1 Slika dugmeta za određivanje glavne pokretačke funkcije

Ako se pogleda sama `main` datoteka na slici Slika 4.1 Slika dugmeta za određivanje glavne pokretačke funkcije vidjet će se da se sastoji od pokretačke funkcije zvane `runApp`, `MultiProvider` objekta s svim *providerima* i objektom djetetom koji predstavlja ono što

se pokreće kao početni prikaz aplikacije. U tom objektu djetetu je obavezno razred (engl. *Class*) kojeg nadograđuje `StatelessWidget` apstraktni razred. U ovom slučaju je u objektu djetetu razred koji se zove `MyApp`. Ako je dano da je taj razred prezentiran kao objekt djetete, taj objekt mora vraćati `MaterialApp` objekt kao rezultat njegovog izvršavanja. To se može vidjeti u sljedećem kodu:

```
void main() {
  runApp(
    MultiProvider(
      providers: [
        ChangeNotifierProvider(create: (_) =>
          MessageProvider()),
        ChangeNotifierProvider(create: (_) =>
          MarkersProvider()),
        ChangeNotifierProvider(create: (_) =>
          RegisterProvider()),
        ChangeNotifierProvider(create: (_) =>
          OptionsEditProvider()),
        ChangeNotifierProvider(create: (_) =>
          LoginProvider()),
        ChangeNotifierProvider(create: (_) =>
          HomeProvider()),
        ChangeNotifierProvider(create: (_) =>
          OfferProvider()),
        ChangeNotifierProvider(create: (_) => RoomProvider())
      ],
      child: const MyApp()
    )
  );
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      initialRoute: '/',
      onGenerateRoute: (settings) =>
        RoutePusher.routeGenerator(settings)
    );
  }
}
```



```
}  
}
```

Kod 4.1 Kod glavne datoteke iz koje se pokreće *frontend*

U kodu se može primijetiti da `MaterialApp` sadrži dva parametra. Jedan pod imenom `initialRoute` koji prima niz znakova koji predstavljaju ime rute i drugi koji predstavlja funkciju koja se izvršava pri slanju imenovane rute zvanu `onChangeRoute`. Ova funkcija u svoje parametre prima parametar `settings` čiji je tip `RouteSettings`. Funkcija je zatim `lambda` (`Lambda` je funkcija koje se implementira poput parametra u novu funkciju ili delegat. Ako je dano da `lambda` nema parametara onda ju karakterizira otvorena i zatvorena zagrada te znakovi `=>` ili `->` ovisno o programskom sučelju) produljena u jedinstvenu vanjsku funkciju u klasi `RoutePusher` zvanu `routeGenerator`. Ta funkcija odrađuje navigaciju po rutama svaki put kad je imenovana ruta poslana preko `Navigator` objekta. U sljedećem primjeru će se pokazati sadržaj `routeGenerator` funkcije kako bi se поближе objasnila navigacija kroz rute u prezentacijskom sloju.

```

static Route<dynamic> routeGenerator(RouteSettings settings){
  switch (settings.name){
    case '/':
      return MaterialPageRoute(builder: (_) =>
        LayoutPusher(iphoneLogIn: LogInIphoneWidget(),
          desktopLogIn: LogInDesktopWidget()));
    case '/HomeAndOffers':
      return MaterialPageRoute(builder: (_) =>
        BottomNavigatorWidget());
    case '/Room':
      return MaterialPageRoute(builder: (_) =>
        LayoutPusher(iphoneLogIn: RoomIphoneWidget(),
          desktopLogIn: RoomIphoneWidget()));
    case '/Chat':
      return MaterialPageRoute(builder: (_) =>
        LayoutPusher(iphoneLogIn: ChatDesktopWidget(),
          desktopLogIn: ChatDesktopWidget()));
    case '/Map':
      return MaterialPageRoute(builder: (_) =>
        LayoutPusher(iphoneLogIn: MapDesktopWidget(),
          desktopLogIn: MapDesktopWidget()));
    case '/Options':
      return MaterialPageRoute(builder: (_) =>
        LayoutPusher(iphoneLogIn: OptionsIphoneWidget(),
          desktopLogIn: OptionsDesktopWidget()));
    case '/Register':
      return MaterialPageRoute(builder: (_) =>
        LayoutPusher(iphoneLogIn: RegisterIphoneWidget(),
          desktopLogIn: RegisterIphoneWidget()));
    case '/RegisterSecond':
      return MaterialPageRoute(builder: (_) =>
        LayoutPusher(iphoneLogIn: RegisterSecondIphoneWidget(),
          desktopLogIn: RegisterSecondIphoneWidget()));
    case '/OptionsEdit':
      return MaterialPageRoute(builder: (_) =>
        LayoutPusher(iphoneLogIn: OptionsEditIphoneWidget(),
          desktopLogIn: OptionsEditIphoneWidget()));
    case '/BfpResult':
      return MaterialPageRoute(builder: (_) =>
        LayoutPusher(iphoneLogIn: BfpResultIphoneWidget(),

```

```

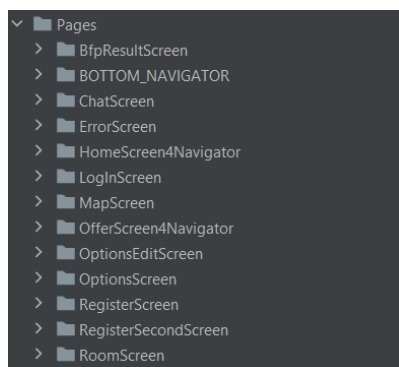
        desktopLogIn: BfpResultIphoneWidget());
    }
    return MaterialPageRoute(builder: (_) => ErrorPage());
}

```

Kod 4.2 Kod iz funkcije routeGenerator

Kroz funkciju se vidi jednostavna `switch-case` naredba koja provjerava ime u `settings` parametru i na osnovi imena proslijeđenog preko Navigator objekta odabire `LayoutPusher` objekte koji primaju dva objekta nadograđena od razreda `StatefullWidget`. `LayoutPusher` na osnovi rezolucije ekrana vraća ispravan *layout* za mobitel ili desktop. Na taj način se kroz ruter šalje kontroler koji prikazuje *widgete* na responzivan način.

Svi objekti koji su isključivo zaduženi za prikaz *widgeta* su proslijeđeni u `LayoutPusher` i proizlaze iz razreda koji se nalaze u datoteci *pages* na slici Slika 4.2 Lista *screenova* u projektu.



Slika 4.2 Lista *screenova* u projektu

To je prikaz čitavog prezentacijskog sloja. Ako bi se svaki zaslon zasebno analizirao, bilo bi previše sadržaja koji je relativno beskoristan ako je dano da će se u opisu funkcionalnosti svi zaslone objasniti.

4.2. Sloj poslovne logike

U sloju poslovne logike će se prezentirati podatci koji će se proslijediti na prezentacijski sloj i logičke odluke koje se provode pri prosljeđivanju tih podataka. U sloju će se proći po dijelu *Flutter* tehnologije i dijelu *C#* tehnologije kako bi se u potpunosti ustanovilo kakav je sustav iza prezentacijskog sloja. Sloj poslovne logike se može podijeliti na četiri djela: dobavljački modul, model, DTO, HTTP *request* modul.

Prvi dio sloja poslovne logike je dobavljački modul koji direktno dobavlja model ili DTO (engleska skraćenica, *Data Transfer Object*). Dobavljači se doslovno zovu *provideri* i njihova je zadaća da se ponašaju kao neka vrsta statičkih razreda koji se ne moraju deklarirati kako bi se izvukli podatci iz njih, već se do njih može doći preko konteksta stanja u kojem se nalazimo. Oni su dobar izbor za dobavljanje informacija jer se do svih dobavljača može doći s bilo kojeg modula s bilo kojeg zaslona. Jedini uvjet koji svaki *provider* treba imati jest da je nadograđen razredom `ChangeNotifier` i da je deklariran u `MultiProvider` objektu u `runApp` funkciji. U nastavku će se ispisati primjer dobavljača.

```
class HomeProvider extends ChangeNotifier{
  UserDTO user = UserDTO.empty();

  set setUser(UserDTO value) {
    user = value;
    notifyListeners();
  }

  Future reload() async {
    HttpRequests().getUserDTO(user.idUser).then((value) {
      user = value;
      notifyListeners();
    });
  }

  Future<int> setUserById(int id) async {
    user = await HttpRequests().getUserDTO(id);
    return 1;
  }
}
```

Kod 4.3 Primjer *provider*a iz aplikacije

Ovaj se dobavljač nakon njegovog kreiranja treba deklarirati u `runApp` funkciji kako je prikazano na slici Slika 4.3 Prikaz deklariranih *provider*a.

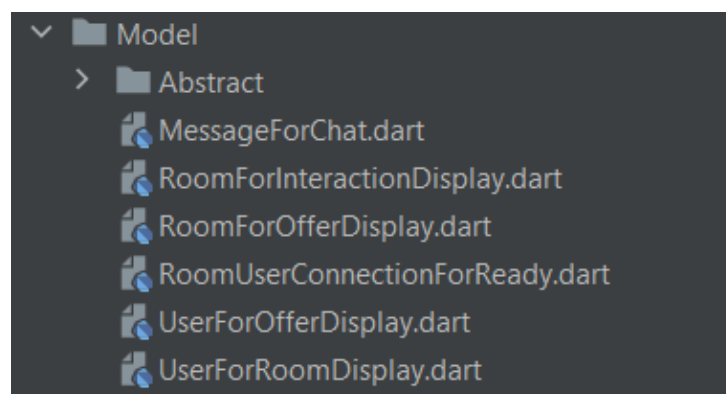
```

runApp(
  MultiProvider(
    providers: [
      ChangeNotifierProvider(create: (_) => MessageProvider()),
      ChangeNotifierProvider(create: (_) => MarkersProvider()),
      ChangeNotifierProvider(create: (_) => RegisterProvider()),
      ChangeNotifierProvider(create: (_) => OptionsEditProvider()),
      ChangeNotifierProvider(create: (_) => LoginProvider()),
      ChangeNotifierProvider(create: (_) => HomeProvider()),
      ChangeNotifierProvider(create: (_) => OfferProvider()),
      ChangeNotifierProvider(create: (_) => RoomProvider())
    ],
    child: const MyApp()
  ) // MultiProvider
);

```

Slika 4.3 Prikaz deklariranih *providera*

Drugi dio poslovne logike je model. Model je skup razreda koji, nakon konvertiranja iz DTO-a koji je dohvaćen preko `HttpRequest`, postaje glavno odredište za vraćanje podataka na prezentacijski sloj. Model sam po sebi nema puno funkcija što se tiče pretvaranja samog sebe u druge oblike, no zato ima funkcije za vraćanje *widgeta* koji su sukladni modelu kako bi ga bilo lakše ispisati na zaslon. Model je prikazan na slici Slika 4.4 Popis modela u aplikaciji.



Slika 4.4 Popis modela u aplikaciji

Uzet će se za primjer `UserForOfferDisplay`. U ovom razredu je normalan konstruktor koji je prisutan u većini razreda modela, no također postoji i funkciju `build_a_full_place` koja gradi mjesto za prikaz u ponudi soba.

```

class UserForOfferDisplay extends UserForDisplay{

    UserForOfferDisplay(super.id, super.image, super.name,
super.surname, super.roomUserConnectionId, super.ready);

    @override
    Widget build_a_full_place(BuildContext context) {
    return Container(
        color: (ready == true) ? Colors.yellowAccent :
            Colors.white,

        child: Column(
            children: [
                ClipOval(child: Container(child:
                    Image.memory(base64Decode(image), width: 50))),
                Container(child: Text(name, style: const
                    TextStyle(fontWeight: FontWeight.w700))),
                Container(child: Text(surname, style: const
                    TextStyle(fontWeight: FontWeight.w700)))
            ],
        ),
    );
}
}

```

Kod 4.4 Prikaz razreda UserForOfferDisplay iz aplikacije

Taj se model preko *provider*a vraća na prezentacijski sloj. *Provider* uzima model i jednostavno ga javno deklarira u sebi. Na taj način svi zaslone mogu pristupiti tom djelu modela i koristiti ga po volji.

Treći dio sloja polovne logike je *Data transfer object* (kratica DTO). DTO je skup razreda koji služi isključivo za prijenos podataka s *backenda* na *frontend*. Kao i svi drugi razredi imaju parametre i konstruktor, ali ih također treba nadograditi s pretvaračima `fromJson` i `toJson` pri čemu `fromJson` prima `Map` objekt kao parametar. U nastavku će se prikazati kod jednog DTO razreda kako bi se razjasnilo o čemu se točno radi.

```

class RoomUserConnectionDTO{
    int? idRoomUserConnection = 0;
    int? roomId = 0;
    int? userId = 0;
    bool? userReady;

    RoomUserConnectionDTO(this.idRoomUserConnection,
this.roomId, this.userId, this.userReady);

    RoomUserConnectionDTO.empty();

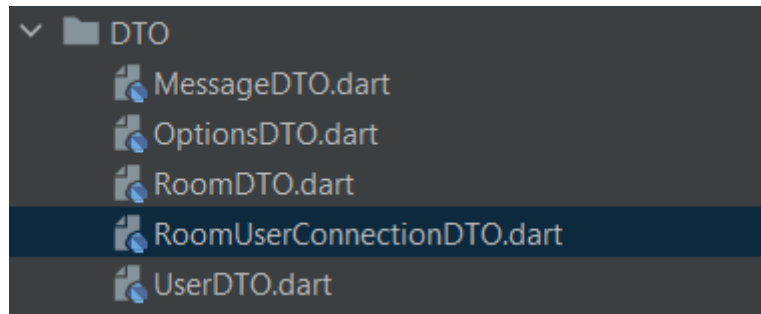
    factory RoomUserConnectionDTO.fromJson(Map<String, dynamic>
json){
        return
RoomUserConnectionDTO(json['idRoomUserConnection'],
        json['roomId'],
        json['userId'],
        json['userReady']);
    }

    Map<String, dynamic> toJson(){
        return{
            'idRoomUserConnection' : idRoomUserConnection,
            'roomId' : roomId,
            'userId' : userId,
            'userReady' : userReady
        };
    }
}

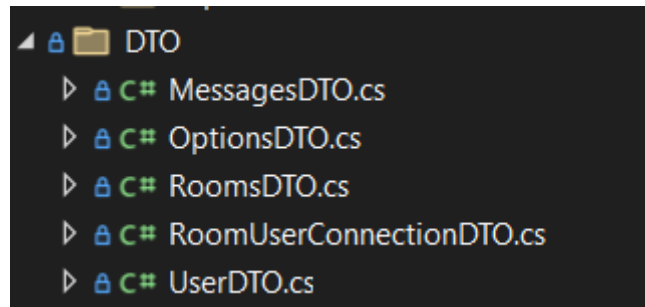
```

Kod 4.5 Primjer DTO razreda u aplikaciji

DTO razred se preko pretvarača (engl. *Converter*) i mapera (engl. *Mapper*) pretvara u model koji se preko *providera* vraća na prezentacijski sloj. Ono što je važno napomenuti je to da je popis DTO razreda u *flutteru* paralelan s popisom DTO razreda u C# tehnologiji. U nastavku će se pokazati slika *flutter* DTO razreda i slika DTO razreda C# tehnologije.



Slika 4.5 Slika DTO razreda u *flutteru*



Slika 4.6 Slika DTO razreda u *C#* tehnologiji

Kao što se može vidjeti na slikama Slika 4.5 Slika DTO razreda u *flutteru* i Slika 4.6 Slika DTO razreda u *C#* tehnologiji, DTO razredi su potpuno isti na *frontendu* i *backendu*. To je napravljeno namjerno jer DTO u svom imenu govori da je namijenjen samo za prijenos podataka i ništa više. DTO-i s *C#* tehnologije se pomoću *JSON* tehnologije prenose pomoću HTTP (kratica za *Hyper Text Transfer Protocol*) zahtjeva na *frontend* gdje se taj JSON kod pretvara, koristeći `fromJson` funkciju unutar DTO razreda na *frontendu*, u instancu tog istog DTO razreda.

Četvrti dio sloja poslovne logike je HTTP *request* modul. Ovaj modul se sastoji isključivo od asinkronih funkcija koje vraćaju `Future` što zapravo znači da vraćaju obećanje da će se nešto ili `void` vratiti u budućnosti. Sve funkcije u modulu su radi jednostavnosti postavljene u jednom razredu koji se zove `HttpRequests` i na taj način imamo jedno univerzalno mjesto od kud se vuku sve naredbe za dohvaćanje podataka s *backendu*. Na *backendu* se preko kontrolera upravlja sadržaj koji će se izložiti na određenom linku. Iz tog razloga je za svaki kontroler rezerviran jedan link. Primjerice, za `UserController` je rezervirana poveznica „`http://localhost:nnnn/api/Users`“ (put poveznice je jednak onome što je postavljeno u opcijama kontrolera). Na tu poveznicu se može slati razne HTTP zahtjeve. Među glavnima su GET, POST, PUT i DELETE pri čemu GET zahtjev ima dva tipa

poveznice. Jednu koja se primjenjuje na sve elemente u bazi podataka i jednu koja se primjenjuje na individualan element u bazi podataka.

U nastavku će se navesti kod jedne funkcije za dohvaćanje podatak pomoću HTTP zahtjeva s *frontenda*:

```
Future<UserDTO> getUserDTO(int id) async {
    try{
        UserDTO _userDto = UserDTO.empty();

        http.Response response = await http.get(Uri.http(authority,
        "${unencodedPath}Users/$id"));
        _userDto = UserDTO.fromJson(jsonDecode(response.body));

        return Future<UserDTO>(() => _userDto);
    }catch(e) {
        print(e);
        throw Exception(e);
    }
}
```

Kod 4.6 Prikaz *frontend* koda za dohvaćanje jednog elementa iz tablice `Users`

Mora se napomenuti da je `authority` varijabla u prethodnom primjeru jednaka nizu znakova „10.0.2.2:5009“, a `unencodedPath` varijabla je jednaka nizu znakova „api“. Te varijable preko funkcije `http` zajedno tvore poveznicu „http://10.0.2.2:5009/api“, a nastavak niza znakova „Users/\$id“, pri čemu je `$id` umetanje varijable `id` u niz znakova, tvori punu poveznicu „http://10.0.2.2:5009/api/Users/\$id“. Na tu se lokaciju šalje HTTP zahtjev za dohvaćanje jednog elementa baze podataka u tablici `Users` bazirano na varijabli `id` koja se šalje na kraju poveznice. U nastavku će se prikazati mjesto na *backendu* koje se pokrene kada HTTP zahtjev dođe do poveznice na koju se šalje.

```

// GET api/<UsersController>/5
    [HttpGet("{id}")]
    public ActionResult<UserDTO> Get(int id)
    {
        UserDTO userDTO = _repoFactory.GetUserRepository().GetOne(id)

        if (userDTO.idUser != default)
        {
            return userDTO;
        }

        return StatusCode(404);
    }

```

Kod 4.7 Prikaz *backend* funkcije za dohvaćanje jednog elementa iz tablice *Users*

Kao što se može vidjeti, kada HTTP zahtjev doprije do poveznice na koju se taj zahtjev šalje (u ovom slučaju na „http://10.0.2.2:5009/api/Users/{id}“), sustav pokrene funkciju unutar kontrolera koja vraća `ActionResult<UserDTO>`. To znači da se može vratiti ili objekt `UserDTO` kao JSON ili određene funkcije kao što je `StatusCode` funkcija. U ovom slučaju funkcija kontrolera `Get` ima ulogu dohvaćanja `UserDTO`a iz tablice `Users` u bazi podataka. Ako dohvaćeni `UserDTO` nije prazan tada se vraća taj isti `UserDTO` u JSON tekstu. Inače se šalje `StatusCode` tipa 404 što normirano znači da nešto nije nađeno. Na taj način se HTTP zahtjevi obrađuju i izvršavaju na servisu.

Ovo zaključuje sloj poslovne logike.

4.3. Sloj pristupa bazi podataka

Sloj pristupa bazi podataka je zadnji sloj aplikacije i predstavlja sloj koji čini razliku između obične *desktop* ili *web* aplikacije i *Enterprise* aplikacije koja ima mjesto za pohranu podataka. U ovom će se sloju opisati pristup bazi podataka iz *backenda* i struktura tablica unutar same baze podataka. Sloj pristupa bazi podataka se dijeli na dva dijela: Na modul pristupa i bazu podataka.

Prvi dio sloja pristupa bazi podataka je modul pristupa. Radi se o takozvanim `Repository` razredima koji su zaduženi da preko adaptera za `MSSQL` (kratica za *Microsoft Structured*

Query Language) server napune DTO s podacima iz baze podataka. Kako bi se to postiglo treba se napraviti niz *repository* razreda koji zrcale kontrolere koji dopremaju podatke prema *frontendu*. Tako na primjer za `UserController` se sukladno treba napraviti `UserRepository` koji sadrži funkcije koje su sukladne s funkcijama u kontroleru. Uzet će se za primjer jedna funkcija iz `UserRepository` razreda koja se primjenjuje u `UserController` razredu.

```
public UserDTO GetOne(int id)
{
    UserDTO user = new UserDTO();
    adapter.openConnection();

    SqlDataReader data =
adapter.getResultByScript("SELECT * from Users LEFT JOIN
Options on Options.UserID = Users.IDUser LEFT JOIN
RoomUserConnection on RoomUserConnection.UserID =
Users.IDUser LEFT JOIN Messages on Users.IDUser =
Messages.UserID where Users.IDUser = @id;",
        new SqlParameter { ParameterName = "@id",
Value = id }
    );

    RoomUserConnectionDTO roomUser = new RoomUserConnectionDTO();
    MessagesDTO message = new MessagesDTO();

    while (data.Read())
    {
        user.idUser = (int)data["IDUser"];
        user.name = (string)data["Name"];
        user.surname = (string)data["Surname"];
        user.email = (string)data["Email"];
        user.userImage = (string)data["UserImage"];

        if(data["IDOptions"] != DBNull.Value) {
            user.options = new OptionsDTO
            {
                idOptions = (int)data["IDOptions"],
                gender = (int)data["Gender"],
                bodyType = (int)data["BodyType"],
                weightInKg = (double)data["WeightInKg"],
```

```

weightInLbs = (double)data["WeightInLbs"],
heightInCms = (double)data["HeightInCms"],
heightInInches = (double)data["HeightInInches"],
bodyFatPercentage = (double)data["BodyFatPercentage"],
userId = (int)data[14],
    };
}

if (data["IDRoomUserConnection"] != DBNull.Value)
{
    if(!user.roomUserConnections.Any(a =>
        a.idRoomUserConnection ==
            (int)data["IDRoomUserConnection"]))
    {
        roomUser = new RoomUserConnectionDTO
        {
            idRoomUserConnection =
                (int)data["IDRoomUserConnection
                    "],
            roomId = (int)data[16],
            userId = (int)data[17],
            userReady =
                ((int)data["UserReady"] == 0) ?
                    false : true
        };

        user.roomUserConnections.Add(
            roomUser);
    }
}

if (data["IDMessage"] != DBNull.Value)
{
    if (!user.messages.Any(a => a.idMessage ==
        (int)data["IDMessage"]))
    {
        message = new MessagesDTO
        {
            idMessage = (int)data["IDMessage"],
            message = (string)data["Message"],
            roomId = (int)data[21],
            userId = (int)data[22],

```

```

        };
        user.messages.Add(message);
    }
}
}
adapter.closeConnection();
return user;
}

```

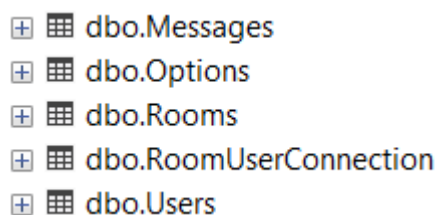
Kod 4.8 Prikaz funkcije za popunjavanje jedne instance `UserDTO` razreda

Funkcija se može svesti na par koraka funkcije:

- Deklaracija praznog DTO objekta koji se na kraju vraća
- Korištenje adaptera kako bi se izvršila SQL skripta te vratio `SqlDataReader` ako je to potrebno.
- Provlačenje `SqlDataReader` objekta kroz `while` funkciju i popunjavanje instance DTO objekata ovisno o poziciji u `SqlDataReader` objektu
- Vratiti DTO objekt koji je popunjen

Na taj način se direktno pristupa bazi podataka preko `SqlConnection` objekta koji je naveden u adapteru.

Drugi dio je sama baza podataka. Baza podataka se sastoji od pet tablica: `Users`, `RoomUserConnection`, `Rooms`, `Options` i `Messages`. Svaka ima svoju ulogu i svrhu ovisno o imenu tablice.



Slika 4.7 Prikaz svih tablica unutar baze podataka.

- A) Tablica `Messages` je tablica za pohranjivanje poruka koje se pišu unutar zatvorene sobe. U njih se pohranjuju poruke prilikom komunikacije s ostalim članovima u sobi kada se ona zatvori i postane soba za razgovor
- B) Tablica `Options` je tablica za pohranjivanje statistika o korisniku. U nju se pohranjuju statistike korisnika pri registraciji. Pod statistike se podrazumijeva visina, težina, postotak tjelesne masnoće, tip tijela i spol.

- C) Tablica `Rooms` pohranjuje sve sobe i podatci o sobama, kao na primjer: ime, opis sobe, da li je soba zatvorena ili otvorena, ukupan broj mjesta...
- D) Tablica `RoomUserConnection` je poveznica između sobe i korisnika pošto se radi o M:M (*many to many*) odnosom između ta dva elementa u aplikaciji.
- E) Tablica `Users` pohranjuje korisnike koji su se registrirali na stranici za registraciju. U njoj se nalaze osobni podatci korisnika isključujući opcije koji su se naveli prije.

Sve se navedene tablice mogu vidjeti na slici Slika 4.7 Prikaz svih tablica unutar baze podatka.

4.4. Sigurnosni aspekti aplikacije

Jedini sigurnosni aspekt aplikacije je *Hashirani password*. Password se osigurava pomoću PBKDF2 funkcije (kratica za *Password Based Key Derivation Function 2*). Ta funkcija je namijenjena specifično za *hashiranje* passworda pomoću HMAC kriptografskog ključa i SHA512 *hashing* algoritma. Sljedeći kod će dodatno predstaviti ovu verziju *hashing* funkcije.

```

set setPassword(String value){
    try{
        String _salt = "dominarlyin";
        PBKDF2KeyDerivator pbkdf2 =
            PBKDF2KeyDerivator(HMac(SHA512Digest(), 512));

        Uint8List bytes = Uint8List(86);

        pbkdf2.init(Pbkdf2Parameters(Uint8List.fromList(_
            salt.codeUnits), 100, 32));
        pbkdf2.deriveKey(Uint8List.fromList(
            value.codeUnits), 0, bytes, 0);

        String result = base64Encode(bytes);

        password = result;
    }catch(e){
        throw Exception(e);
    }
}

```

Kod 4.9 Prikaz funkcije koja *hashira* password i pridružuje ga varijabli *password*

Za kreaciju funkcije iznad jednostavno se mogu predložiti par koraka:

- Kreirati samu set funkciju i nazvati je `setPassword`
- Deklarirati `_salt` varijablu koja može biti bilo što, no služi za kao važna varijabla za enkripciju *passworda*.
- Deklarirati `PBKDF2KeyDerivator` objekt zajedno s `_mac` parametrom unutar tog istog objekta.
- Unutar `_mac` objekta deklarirati objekt `SHA512Digest` (ili `SHA256Digest`, ili `SHA128Digests`) kao prvi parametar i broj bajtova kao drugi parametar.
- Napraviti `Uint8List` kao mimiku za `byte` i postaviti duljinu na duljinu bajtova u `_mac` objektu ili manje
- Pokrenuti `init` funkciju i zadati parametre
- Pokrenuti `deriveKey` i unijeti zadane parametre
- Kao rezultat postaviti rezultat `deriveKey` funkcije te ga base64 kodirati
- Pridružiti rezultat kodiranja varijabli `password`

Ovo je jedina sigurnosna mjera u aplikaciji.

5. Korištene tehnologije

Od korištenih tehnologija koja će se navesti u ovom poglavlju su Flutter i C#.

[3] Integrirano u ASP.NET tehnologiju, u kojoj se u ovom implementiranom rješenju koristi ASP.NET Core Web API projektni *template*, je C# jezik. U njemu se pisao *backend* za implementirano rješenje. Evo kako knjiga „*Beginning ASP.NET 4.5 in C#*“ navodi .NET framework jezike i opisuje sam ASP.NET: „*The .NET languages: These include Visual Basic, C#, F#, and C++ although third party developers have created hundreds more.*“, „*ASP.NET: This is the engine that hosts web applications you create with .NET and supports almost any feature fro the .NET Framework class library. ASP.NET also includes a set of web specific services, such as scure authentication and data storage.*“.

6. Opis implementiranog rješenja

U ovom poglavlju će se opisivati implementirano rješenje. Pod time se misli opisivanje korištenja prezentacijskog sloja i grafičkog korisničkog sučelja kako bi se pokrenule logičke operacije na poslovnom dijelu aplikacije. Kroz sljedeća potpoglavlja će se proći kroz sve zaslone u aplikaciji i ukratko objasniti kako se koriste.

6.1. Registracija i Prijava

Registracija i prijava su dva osnovna modula u bilo kojoj aplikaciji za dodavanje i autorizaciju korisnika. Zaslone za prijavu korisnika je ujedno i prvi zaslon u implementiranom rješenju. Taj zaslon sadrži tekst naslova, dva okvira za unos teksta: jedan za adresu elektroničke pošte i jedan za korisničku zaporke. Ispod su tri gumba. Prvi je za potvrdu unosa korisničkog imena i zaporke, drugi je za prelazak na registracijski zaslon i treći je za izlazak iz aplikacije. Slika Slika 6.1 Prikaz zaslona za prijavu će prikazati točan izgled zaslona:

The image shows a login interface with the following elements:

- Title:** Log In
- Form Fields:**
 - Email:** A text input field with the placeholder text "Email".
 - Password:** A text input field with the placeholder text "Password".
- Buttons:**
 - Log In:** A yellow rounded button.
 - Register:** A yellow rounded button.
 - Exit:** A black rounded button.

Slika 6.1 Prikaz zaslona za prijavu

Kako bi se pokrenuo sustav za autorizaciju korisnika na osnovi adrese elektroničke pošte i zaporke, treba se odabrati dugme za potvrdu unosa s tekстом „*Log In*“. Nakon odabira te opcije, sustav će provjeriti da li su informacije točne. Ako se ne podudaraju s nekim od korisnika na bazi podataka ili ako se nisu popunili svi okviri, sustav će obavijestiti o grešci. Ako su se popunili svi okviri ispravno sustav prosljeđuje na glavnu stranicu.

Kako bi se pristupilo zaslonu za registraciju, treba se odabrati gumb s tekстом „*Register*“. Kada se odabere taj gumb, sustav ide do zaslona za registraciju. On se sastoji od pet okvira za unos teksta: prvi za unos imena, drugi za unos prezimena, treći za unos adrese elektroničke pošte, četvrti za unos zaporke i peti za ponovo unošenje zaporke kao provjeru. Ispod okvira su dva gumba: prvi je za nastavak na zaslon za upravljanje opcijama i drugi za povratak na stranicu za prijavu. Slika Slika 6.2 Prikaz zaslona za registraciju će pokazati поближе kako izgleda zaslon za registraciju:

Register

Name:

Surname:

Email:

Password:

Repeat password:

Save and proceed

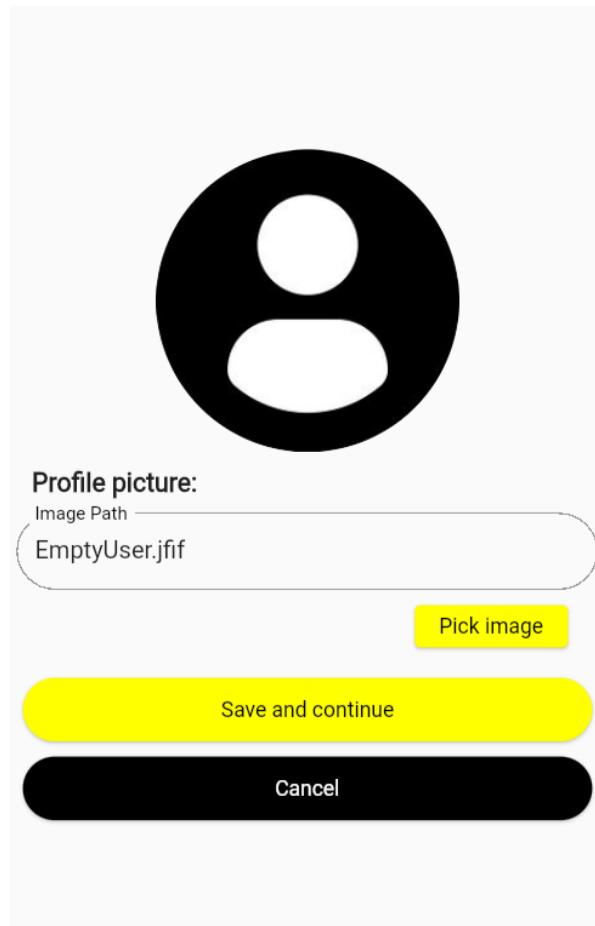
Back

Slika 6.2 Prikaz zaslona za registraciju

Kada bi se unijeli svi podatci ispravno i ako se pretpostavi da korisnik koji je unesen već ne postoji u bazi podataka, sustav bi išao do zaslona za postavljanje slike. Inače bi obavijestio o grešci.

Zaslon za postavljanje slike se sastoji od okvira za sliku na vrhu, okvira za tekst koji prikazuje ime odabrane slike s gumbom na desnoj strani koji služi za odabir slike i gumbom za navigaciju na zaslon za upravljanje postavkama koji će ići tamo ako je slika odabrana.

Slika Slika 6.3 Prikaz zaslona za odabir slike prikazuje zaslon:



Slika 6.3 Prikaz zaslona za odabir slike

6.2. Upravljanje postavkama

Ako su se unijeli svi podatci ispravno na zaslonu za odabir slike, on će ići do zaslona za upravljanje postavkama. Zaslون za upravljanje postavkama je kompliciranije građe. Sastoji se od dva djela: Djela za ažuriranje podataka i djela za izračun postotka tjelesne masnoće. Dio za ažuriranje podataka se sastoji od reda okvira za unos visine i reda okvira za unos težine pri čemu je jedna vrijednost američka, a jedna europska. Ispod ta dva reda se nalaze dva niza za odabir: prvi je za odabir spola, a drugi za odabir tipa tijela koji se generira na osnovi odabranog spola.

Update Options

Weight:

kg
 lbs

Height:

cm
 in

Gender:

Male ▼

Body Type:

None ▼

Slika 6.4 Prikaz dijela za ažuriranje postavki korisnika na zaslonu za uređivanje postavki

Kada se upisuje težina u jedna od okvira za tekst u zadanom redu, u drugom okviru se automatski generira i zaokružuje druga vrijednost. Isto se događa kada se upisuje u red za visinu. Pod naslovom „*Gender*“ je niz za odabir spola. Nakon što se odabere spol, ispod pod naslovom „*Body Type*“ se bira tip tijela.

Drugi dio upravljanja postavkama je izračunavanje postotka tjelesne masnoće. Taj se dio također modificira ovisno o spolu odabranom u prvom djelu na način da se dodaje unos za obujam kuka ako je odabrani spol ženski.

Calculate body fat percentage

Neck girth:

cm
 in

Waist girth:

cm
 in

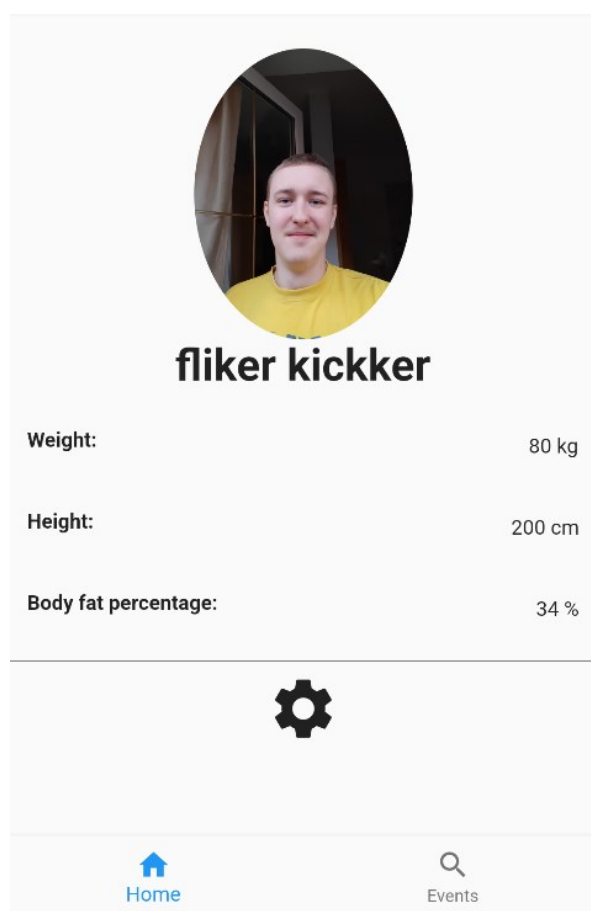
Please select a valid body type

Slika 6.5 Prikaz dijela za izračunavanje postotka tjelesne masnoće u zaslonu za uređivanje postavki

Sastoji se od dva ili tri reda okvira za unos pri čemu je u jednom redu jedan okvir za unos europske vrijednosti, a drugi je za američke vrijednosti. Ako se nije odabrao tip tijela, sustav zamjenjuje gumb za nastavak sa crvenim tekstom „*Please select a valid body type*“. Nakon što se odabere validan tip tijela, pojavljuje se gumb za nastavak sa tekstom „*Calculate body fat precentage*“. Taj nas gumb vodi na zaslon koji samo objavljuje rezultat izračunatog postotka tjelesne masnoće. Nakon toga se može završiti registracija odabirom gumba „*Finish registration*“. Taj nas gumb vodi natrag na stranicu za prijavu korisnika. Tamo se može korisnik prijaviti koristeći podatke koji su bili zapisali u registracijskom dijelu implementiranog rješenja.

6.3. Korištenje glavne stranice i stranice za pretraživanje

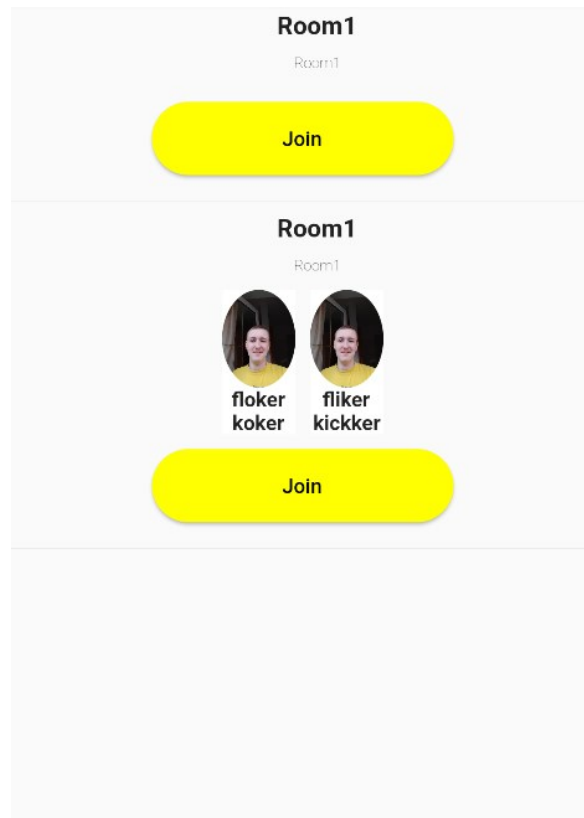
Kada se ispravno unesu podatci na zaslonu za prijavu, gumb za nastavak nas vodi do glavnog zaslona. On se sastoji od osnovnih informacija korisnika koji se prijavio kao što je slika, njegovo ime i prezime, njegova težina, njegova visina i njegov postotak tjelesne masnoće.



Slika 6.6 Prikaz glavnog zaslona u implementiranom rješenju

Uz sve navedeno postoji i gumb koji ima opciju navigacije na zaslon sa svim opcijama korisnika. Na dnu se može uočiti kartica za navigaciju između glavnog zaslona i zaslona za pretraživanje soba. Sve je prikazano na slici Slika 6.6 Prikaz glavnog zaslona u implementiranom rješenju

Zaslon za pretraživanje soba se sastoji od niza soba s gumbima za pridruživanje. Svaka soba ima svoj naslov, svoj opis i prikaz korisnika koji trenutno u sobi.



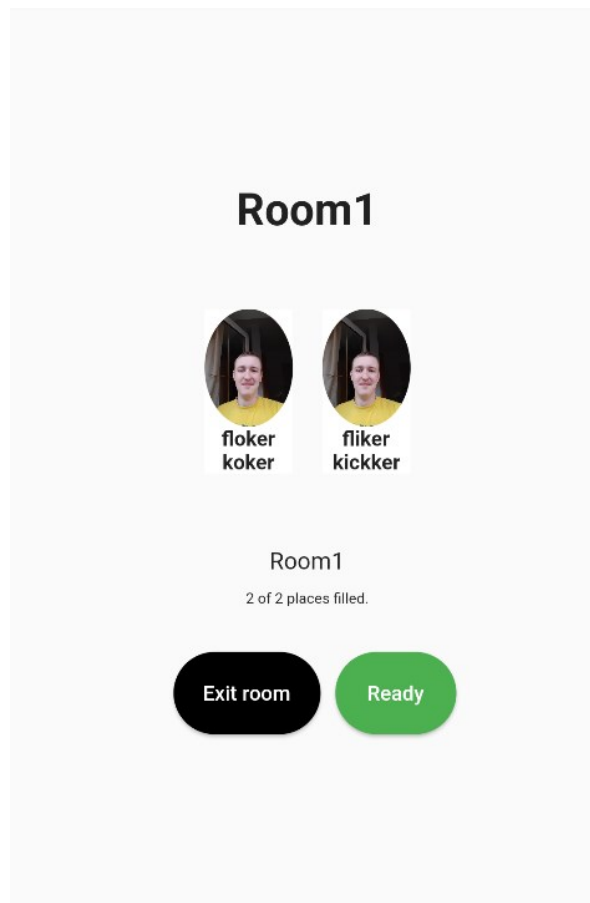
Slika 6.7 Prikaz zaslona za pretraživanje soba

Kako bi se pridružilo u neku od soba treba se odabrati opcija za pridruživanje pritiskom na gumb s tekстом „Join“. Nakon toga sustav odabire hoće li pustiti u sobu na osnovu popunjenosti same sobe. Ako je soba puna, tada sustav otvara *dialog box* objekt koji obavještava da je soba puna. Inače ide u dotičnu sobu. Sve je prikazano na slici Slika 6.7 Prikaz zaslona za pretraživanje soba

6.4. Korištenje sobe

Kada se izabere soba u koju se korisnik pridružuje i u nju pridruži, implementirano rješenje će ići do sobe pod određenom *id* vrijednosti. U toj sobi će se prikazati svi sudionici, uključujući i nedavno pridruženog korisnika, naslov i opis sobe. Ispod svih grafičkih

elemenata sobe su jedan ili dva gumba, ovisno o tome je li soba puna ili nije: prvi je za izlazak iz sobe i razdvajanje praćenja sobe, drugi (ako je prikazan) je za označavanje spremnosti korisnika.

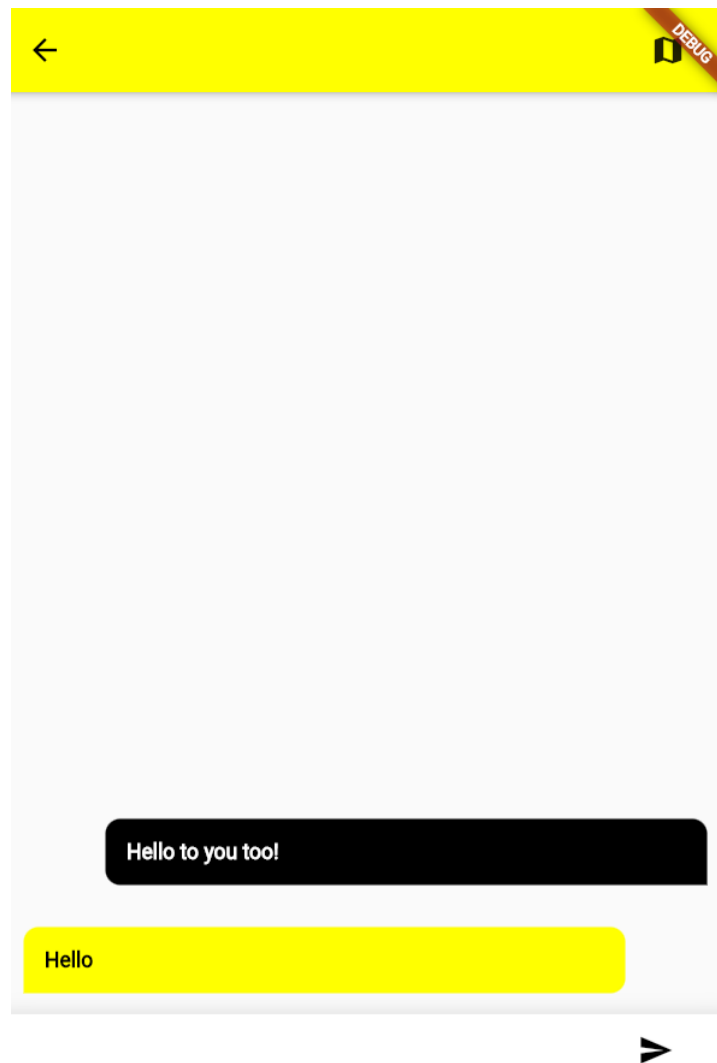


Slika 6.8 Prikaz zaslona sobe u aplikaciji

Kada se odabere opcija gumba s tekстом „Ready“, korisniku koji je odabrao opciju se mijenja boja pozadine u svjetlo žutu kako bi ostalima označio svoju spremnost. Kada su svi sudionici u sobi spremni, soba se zaključava u sobu za razgovor. Sve je prikazano na slici Slika 6.8 Prikaz zaslona sobe u aplikaciji.

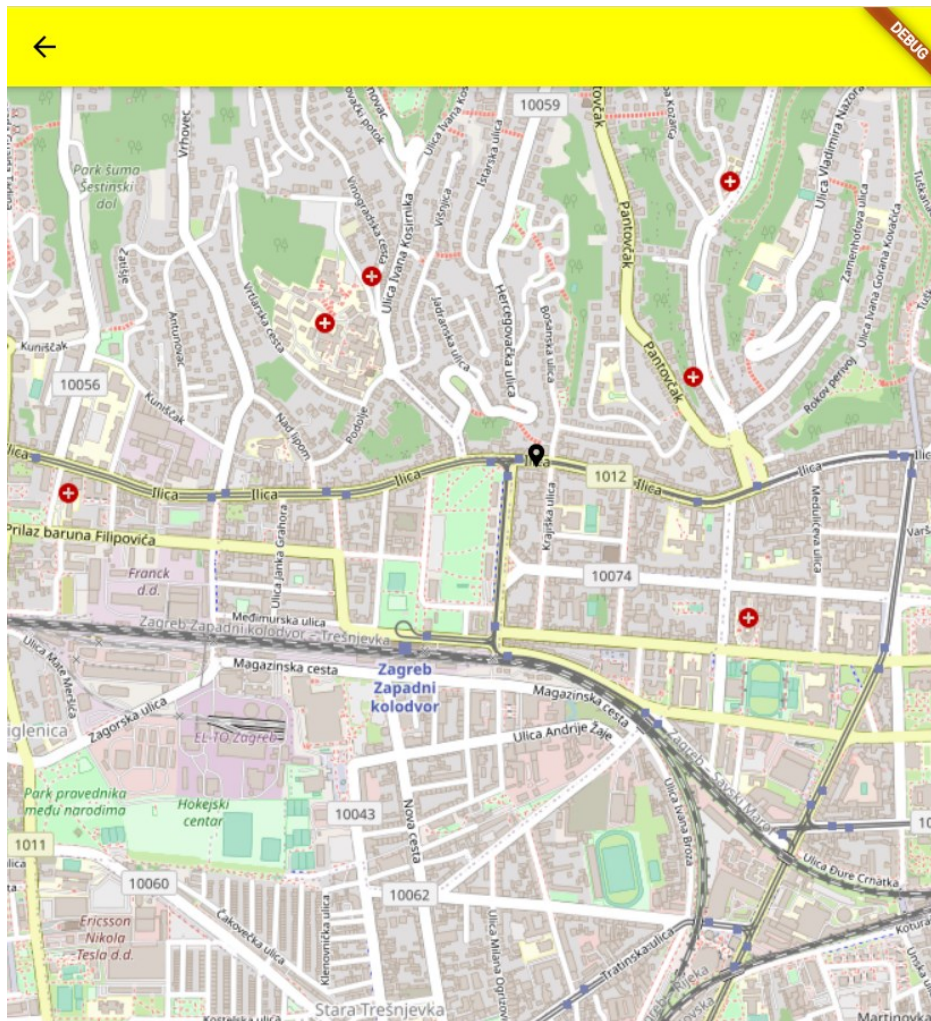
6.5. Stranica za razgovor

Kada se soba zaključa, ona postaje soba za razgovor. Sastoji se od niza poruka koje se nalaze unutar sobe te okvira za unos teksta u donjem dijelu zaslona.



Slika 6.9 Prikaz zaslona za razgovor (soba koja je zatvorena)

Pokraj okvira za unos teksta je gumb koji se koristi za slanje poruke napisane u okviru. U gornjem desnom kutu se nalazi gumb koji ide do drugog dijela sobe za razgovor. U drugom dijelu sobe za razgovor se nalazi karta na kojoj se može označiti proizvoljna lokacija gdje se nalazi mjesto za sastanak. Sve je prikazano na slici Slika 6.9 Prikaz zaslona za razgovor (soba koja je zatvorena).



Slika 6.10 Prikaz zaslona za određivanje proizvoljne pozicije za sastanak

Kako bi se odabralo mjesto za sastanak samo treba škljocnuti mišem na karti i sustav će zabilježiti lokaciju na karti i poslati *Google Maps* link na zaslone s porukama. Sve je prikazano na slici Slika 6.10 Prikaz zaslona za određivanje proizvoljne pozicije za sastanak.

Zaključak

Aplikacija za povezivanje individualaca u svrhu vježbe je potrebna aplikacija u današnjim vremenima. Gledajući na istraživanja provedena u ovom završnom radu i izvan njega, vježba u grupi drastično pomaže u unapređenju kvalitete života. Kroz ovaj završni rad se prošlo po analizi ankete provedene od autora koja je napravljena u svrhu istraživanja mišljenja uže populacije o vježbi i njihovim navikama. Tu se zaključilo da je aplikacija potrebna za svrhe osobnog razvoja u smislu vježbanja iako interes nije bio ekstreman. Zatim se prošlo po slojevima arhitekture aplikacije gdje se pokazalo da su svi slojevi zadovoljeni i čvrsti. Od sloja prezentacijske logike do baze podataka svi su slojevi doprinijeli objedinjenom skupu funkcionalnosti koje jest implementirano rješenje prezentirano u ovom završnom radu. Definirali su se tipovi tehnologije koji su korišteni kako bi učinili aplikaciju ono što je. Nakon toga se opisalo samo implementirano rješenje pri čemu se opisivao prezentacijski sloj i kako se koristi. Ono što je moglo bolje jest sama responzivnost sustava i implementacija *pushera*. *Pusher* nije implementiran u završnoj verziji rada zbog nepoznatih greški u sustavu i komunikaciji između *C# backend* i *Flutter Frontend*.

„Pod punom odgovornošću pismeno potvrđujem da je ovo moj autorski rad čiji niti jedan dio nije nastao kopiranjem ili plagiranjem tuđeg sadržaja. Prilikom izrade rada koristio sam tuđe materijale navedene u popisu literature, ali nisam kopirao niti jedan njihov dio, osim citata za koje sam naveo autora i izvor, te ih jasno označio znakovima navodnika. U slučaju da se u bilo kojem trenutku dokaže suprotno, spreman sam snositi sve posljedice uključivo i poništenje javne isprave stečene dijelom i na temelju ovoga rada“.

U Zagrebu, 28.3.2023.

Popis kratica

QOL	<i>Quality of life</i>	Kvaliteta života
HTTP	<i>Hypertext transfer protocol</i>	Hipertekstualni transferni protokol
PBKDF2	<i>Password based key derivation function 2</i>	Funkcija za derivaciju ključa bazirana na lozinki
MSSQL	<i>Microsoft Structured Query Language</i>	Mikrosoftov strukturirani upitni jezik
ASP.NET	<i>Active Server Pages Network Enabled Technologies</i>	Aktivne serverske stranice mrežno osposobljenih tehnologija

Popis slika

Slika 3.1 Slika prvog pitanja ankete	6
Slika 3.2 Slika drugog pitanja ankete	7
Slika 3.3 Slika trećeg pitanja ankete.....	7
Slika 3.4 Slika četvrtog pitanja ankete	8
Slika 3.5 Slika petog pitanja ankete.....	9
Slika 3.6 Slika šestog pitanja ankete	10
Slika 3.7 Slika sedmog pitanja ankete	10
Slika 3.8 Slika osmog pitanja ankete.....	11
Slika 3.9 Slika devetog pitanja ankete.....	12
Slika 3.10 Slika desetog pitanja ankete	12
Slika 3.11 Slika jedanaestog pitanja ankete.....	13
Slika 3.12 Analiza prvog pitanja ankete.....	14
Slika 3.13 Analiza drugog pitanja ankete.....	15
Slika 3.14 Analiza trećeg pitanja ankete	15
Slika 3.15 Analiza četvrtog pitanja ankete	16
Slika 3.16 Analiza petog pitanja ankete	17
Slika 3.17 Analiza šestog pitanja ankete	17
Slika 3.18 Analiza sedmog pitanja ankete.....	18
Slika 3.19 Analiza osmog pitanja ankete.....	19
Slika 3.20 Analiza devetog pitanja ankete.....	19
Slika 3.21 Analiza desetog pitanja	20
Slika 3.22 Analiza jedanaestog pitanja ankete	21
Slika 4.1 Slika dugmeta za određivanje glavne pokretačke funkcije	22
Slika 4.2 Lista <i>screenova</i> u projektu	26

Slika 4.3 Prikaz deklariranih <i>providera</i>	28
Slika 4.4 Popis modela u aplikaciji	28
Slika 4.5 Slika DTO razreda u <i>flutteru</i>	31
Slika 4.6 Slika DTO razreda u <i>C#</i> tehnologiji.....	31
Slika 4.7 Prikaz svih tablica unutar baze podataka.	36
Slika 6.1 Prikaz zaslona za prijavu.....	40
Slika 6.2 Prikaz zaslona za registraciju	41
Slika 6.3 Prikaz zaslona za odabir slike	42
Slika 6.4 Prikaz dijela za ažuriranje postavki korisnika na zaslonu za uređivanje postavki	43
Slika 6.5 Prikaz dijela za izračunavanje postotka tjelesne masnoće u zaslonu za uređivanje postavki.....	43
Slika 6.6 Prikaz glavnog zaslona u implementiranom rješenju.....	44
Slika 6.7 Prikaz zaslona za pretraživanje soba	45
Slika 6.8 Prikaz zaslona sobe u aplikaciji.....	46
Slika 6.9 Prikaz zaslona za razgovor (soba koja je zatvorena).....	47
Slika 6.10 Prikaz zaslona za određivanje proizvoljne pozicije za sastanak	48

Popis kodova

Kod 4.1 Kod glavne datoteke iz koje se pokreće <i>frontend</i>	24
Kod 4.2 Kod iz funkcije <code>routeGenerator</code>	26
Kod 4.3 Primjer <i>provider</i> a iz aplikacije	27
Kod 4.4 Prikaz razreda <code>UserForOfferDisplay</code> iz aplikacije	29
Kod 4.5 Primjer DTO razreda u aplikaciji	30
Kod 4.6 Prikaz <i>frontend</i> koda za dohvaćanje jednog elementa iz tablice <code>Users</code>	32
Kod 4.7 Prikaz <i>backend</i> funkcije za dohvaćanje jednog elementa iz tablice <code>Users</code>	33
Kod 4.8 Prikaz funkcije za popunjavanje jedne instance <code>UserDTO</code> razreda.....	36
Kod 4.9 Prikaz funkcije koja <i>hashira</i> password i pridružuje ga varijabli <i>password</i>	37

Literatura

- [1] T. J. S. e. a. Gregory G. Kolden*, »A Pilot Study Of Group Exercise Training (Get) For Women With Primary Breast Cancer: Feasibility And Health Benefits,« *Psychology: Journal of the Psychological, Social and Behavioral Dimensions of Cancer*, 2002.
- [2] T. C. Ashley Elizabeth Muller, »Group exercise to improve quality of life among substance users,« *Scandinavian journal of public health*, Oslo, 2015.
- [3] M. MacDonald, *Beginning ASP.NET 4.5 in C#*, Berkley: Apress, 2012.