

Mobilna aplikacija za teretane

Madžar, Maksim

Undergraduate thesis / Završni rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Electrical Engineering and Computing / Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:168:450765>

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

Download date / Datum preuzimanja: **2025-03-14**



Repository / Repozitorij:

[FER Repository - University of Zagreb Faculty of Electrical Engineering and Computing repository](#)



SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1367

MOBILNA APLIKACIJA ZA TERETANE

Maksim Madžar

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1367

MOBILNA APLIKACIJA ZA TERETANE

Maksim Madžar

Zagreb, lipanj 2024.

ZAVRŠNI ZADATAK br. 1367

Pristupnik: **Maksim Madžar (0036541695)**
Studij: Elektrotehnika i informacijska tehnologija i Računarstvo
Modul: Računarstvo
Mentor: prof. dr. sc. Igor Mekterović

Zadatak: **Mobilna aplikacija za teretane**

Opis zadatka:

Napraviti mobilnu aplikaciju za vlasnike teretane, trenere i članove teretane. Članovima teretane treba omogućiti da imaju uvid u sadržaje koje teretana pruža (npr. vođeni treninzi), mogućnost prijave na sadržaje, uvid u popunjenost grupa te uvid u broj ljudi u teretani u stvarnom vremenu. Članovi se prilikom dolaska u teretanu evidentiraju putem QR koda koji generira aplikacija, pa je u svakom trenutku dostupna informacija tko je u teretani. Članovima omogućiti recenziranje sadržaja i prijavu kvarova i prijedloga za poboljšanje. Kroz aplikaciju voditi i status članstva i omogućiti automatska obnavljanja članstva. Ostvariti prototip sustava preporuka gdje bi članovi dobivali pogodnosti kad teretana dobije novog člana na temelju njihove preporuke. Dodatno, razmisliti o uvođenju elemenata gejmfikacije kako bi potaklo korištenje pasivnih termina i sadržaja teretane. Trenerima omogućiti upravljanje kalendarom, definiranje i uređivanje termina vođenih treninga te omogućiti da članovi primaju push notifikacije o tome. Vlasnicima teretane omogućiti skup izvještaja o radu teretane s obzirom na prikupljene podatke (izvještaj o članovima, iskorištenosti termina, trenerima itd.). Donijeti ocjenu ostvarenog pristupa te smjernice za budući razvoj.

Rok za predaju rada: 14. lipnja 2024.

Sadržaj

1. Uvod.....	3
2. Opis problema i zahtjevi.....	5
2.1. Opis problema.....	5
2.2. Korisnički zahtjevi.....	5
2.3. Funkcionalni zahtjevi.....	6
2.4. Ne-funkcionalni zahtjevi.....	7
3. Tehnologije.....	8
3.1. Kotlin.....	8
3.2. Spring Framework.....	10
3.3. Android.....	12
4. Opis podatkovnog modela rješenja.....	14
5. Opis rješenja.....	18
5.1. Arhitektura sustava.....	18
5.2. Funkcionalnosti aplikacije.....	19
6. Zaključak.....	38
7. Literatura.....	40

1. Uvod

U suvremenom, ubrzanom načinu života, održavanje zdravlja postaje sve bitnije, a teretane i fitness centri ključna su mjesta za one koji žele ostati u dobroj fizičkoj formi. Broj ljudi koji prepoznaju prednosti kontinuiranog vježbanja neprestano raste, zbog čega su teretane sve popularnije. Upravljanje teretanom nije lak zadatak, naročito s rastućim očekivanjima i brojem korisnika. Vlasnici teretana i njihovi treneri suočavaju se s izazovima kako bi ostali konkurentni na natjecateljskom tržištu. Ključ uspjeha leži u stalnom uvođenju inovacija, uključujući primjenu naprednih tehnologija koje unapređuju korisničko iskustvo. Današnji posjetitelji očekuju da teretane pružaju raznovrsne programe i da koriste moderne tehnologije, poput aplikacija za rezervaciju treninga i praćenje napretka. Stoga postoji potreba za softverom koji će rješavati probleme vlasnika teretana, trenera i korisnika, čineći usluge dostupnijima i efikasnijima. Ovaj završni rad bavi se razvojem sveobuhvatnog programskog rješenja koje adresira ključne probleme s kojima se suočavaju vlasnici, treneri i korisnici teretana. Cilj je stvoriti aplikaciju koja će omogućiti jednostavno upravljanje teretanom, optimizirati rad trenera te poboljšati iskustvo korisnika. Uvođenjem naprednih tehnologija kao što su mobilne aplikacije za rezervaciju treninga, praćenje aktivnosti i napretka korisnika, te upravljanje članarinama i plaćanjima, značajno će unaprijediti operativnu učinkovitost i korisničko zadovoljstvo.

Razvoj jednog takvog programskog rješenja namijenjenog mobilnim uređajima temelji se na korištenju najnovijih tehnologija i alata. Mobilna aplikacija će zato biti razvijena koristeći Kotlin, moderan programski jezik dizajniran za razvoj Android aplikacija. Kotlin nudi brojne prednosti, uključujući sigurnost od *null* vrijednosti, inferenciju tipova i korutine za jednostavno asinkrono programiranje. *Backend* sustav bit će razvijen koristeći Spring Framework, koji omogućuje inverziju kontrole i injekciju ovisnosti, što olakšava upravljanje kompleksnim aplikacijama. Spring Boot će se koristiti za ubrzanje postavljanja i konfiguracije aplikacije, dok će Spring Data JPA omogućiti jednostavan pristup bazama podataka. Osim toga, aplikacija će biti integrirana s Auth0 za sigurno upravljanje autentifikacijom i autorizacijom korisnika. Stripe će biti korišten za upravljanje plaćanjima i članarinama, osiguravajući sigurno i pouzdano procesiranje financijskih transakcija. Firebase Cloud Messaging (FCM) omogućit će slanje *push* notifikacija korisnicima, čime će se poboljšati komunikacija i angažman korisnika.

Ovaj rad strukturiran je na način da detaljno obuhvati sve aspekte razvoja programskog rješenja za upravljanje teretanom. U prvom dijelu rada opisani su problemi i zahtjevi korisnika, uključujući funkcionalne i ne-funkcionalne zahtjeve aplikacije. Sljedeći dio rada fokusira se na tehnologije korištene za razvoj aplikacije, uključujući Kotlin, Spring Framework i Android SDK. Treći dio rada obuhvaća opis podatkovnog modela, dok četvrti dio detaljno opisuje arhitekturu sustava i funkcionalnosti aplikacije. Na kraju rada nalazi se zaključak koji sažima postignuća rada i predlaže smjernice za buduća istraživanja i razvoj.

2. Opis problema i zahtjevi

2.1. Opis problema

Upravljanje teretanom postaje sve složenije kako raste broj korisnika i njihove potrebe. Vlasnici teretana moraju se nositi s administrativnim zadacima kao što su praćenje članstava, upravljanje rasporedima treninga, te osiguravanje visokog standarda usluga. Također, u današnjem tehnološki naprednom društvu, korisnici očekuju jednostavan i brz pristup informacijama putem mobilnih aplikacija. Na primjer, žele biti u mogućnosti pregledavati rasporede treninga, rezervirati termine, pratiti svoj napredak i davati povratne informacije o uslugama. Nedostatak takvih funkcionalnosti može rezultirati nezadovoljstvom korisnika i gubitkom konkurentnosti teretane na tržištu.

2.2. Korisnički zahtjevi

- **Vlasnici** - Vlasnici teretana zasigurno imaju važnu ulogu u uspješnom poslovanju svojih objekata. Kako bi si olakšali posao te optimizirali odluke, potrebni su im svakakvi podaci o njihovim teretanama, a za prikupljanje istih treba im alat koji im to omogućuje. Također, sustav za praćenje broja ljudi u teretani u stvarnom vremenu ključan je sprječavanje prenatrpanosti i održavanje visokog standarda korisničke usluge. Isto tako, potrebni su im i sveobuhvatni izvještaji o aktivnostima teretane i angažmanu članova.
- **Treneri** – Treneri su ključni za pružanje usluga u teretani. Najčešće, njihov glavni posao je vođenje grupnih treninga, što predstavlja jednu od najbitnijih aktivnosti u teretanama s obzirom da veliki udio ljudi zapravo i koristi usluge teretana upravo zbog grupnih treninga. Kako bi osigurali da treninzi nisu prenatrpani, te kako bi članovi na vrijeme bili obavješteni, trenerima je potrebno omogućiti sustav za vođenje treninga kroz koji mogu objavljivati nadolazeće treninge, otkazivati i mijenjati iste, pregledavati treninge drugih trenera u teretani kroz kalendarski posloženo grafičko sučelje, ali i od članova dobivati povratne informacije o vođenim treninzima.
- **Članovi** - Članovi, kao krajnji korisnici, ključni su dio procesa održavanja i razvoja teretana, s obzirom da i treneri i vlasnici djeluju kako bi članovi bili što zadovoljniji.

Kako bi se njihovo zadovoljstvo maksimiziralo, potrebno je omogućiti im da pregledavaju informacije o popunjenosti teretane i dostupnosti i popunjenosti treninga u stvarnom vremenu, kako bi mogli planirati svoje dolaske i rezervirati termine treninga u kojima žele sudjelovati. Također je bitno omogućiti im davanje povratne informacija o uslugama teretane.

2.3. Funkcionalni zahtjevi

Aplikacija mora omogućiti sljedeće funkcionalnosti za članove teretane, trenere i vlasnike teretane:

Za članove:

- **Pregled sadržaja:** Omogućiti uvid u sadržaje koje teretana pruža, s mogućnošću prijave na sadržaje i uvidom u popunjenost grupa.
- **Prijava i evidencija:** Omogućiti evidenciju dolazaka putem QR koda koji generira aplikacije, čime se osigurava trenutni pregled prisutnih korisnika.
- **Uvid u broj ljudi u teretani:** Osigurati članovima uvid u broj ljudi u teretani u stvarnom vremenu.
- **Recenzije:** Implementirati sustav za recenziranje sadržaja i trenera.
- **Automatsko obnavljanje članstva:** Omogućiti automatsko obnavljanje članstva kroz aplikaciju
- **Sustav preporuka:** Razviti sustav preporuka gdje bi članovi dobivali pogodnosti kada teretana dobije novog člana na temelju njihove preporuke
- **Igrifikacija:** Uvesti elemente igifikacije kroz razne vrste izazova kako bi se potaklo korištenje pasivnih termina i povećali dolasci u teretanu.

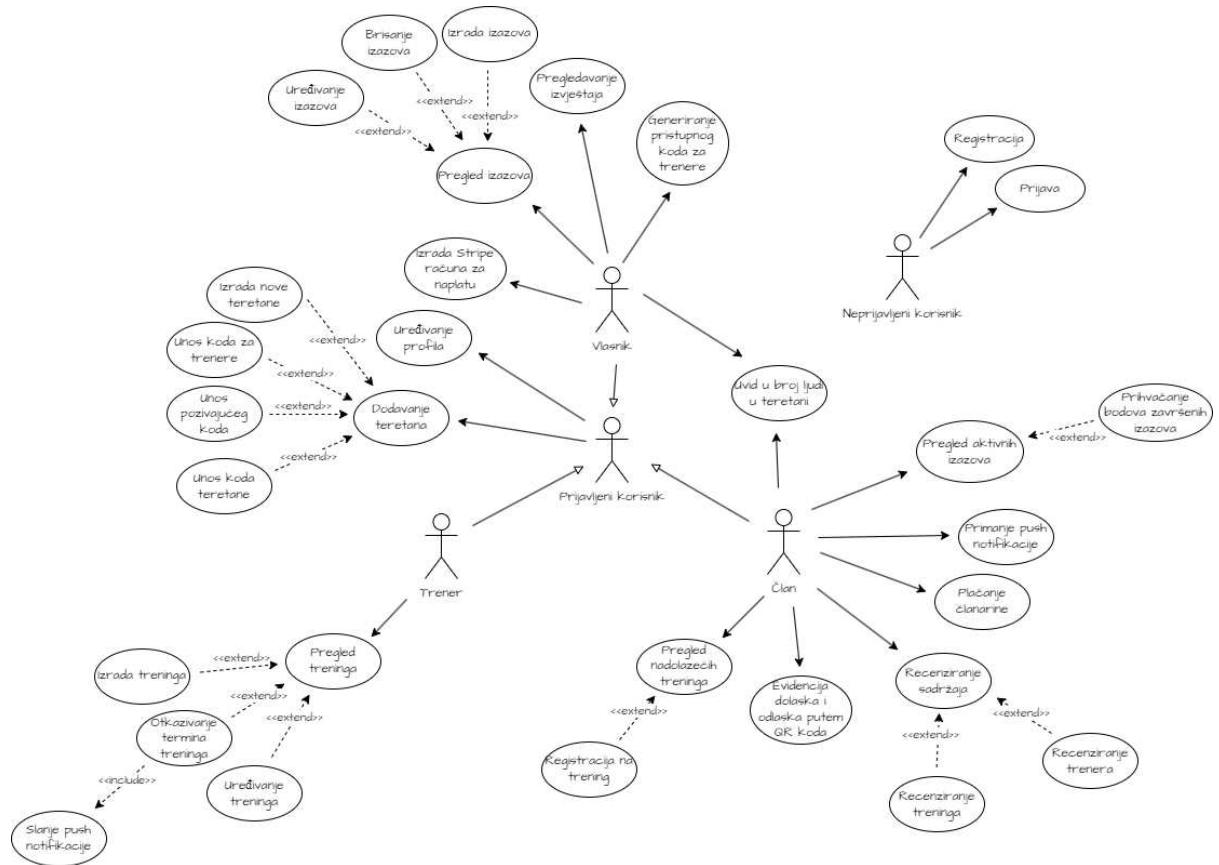
Za trenere:

- **Upravljanje kalendarom:** Omogućiti upravljanje kalendarom s mogućnošću definiranja, uređivanja i otkazivanja termina vođenih treninga.
- **Push notifikacije:** Omogućiti da članovi primaju push notifikacije o promjenama termina vođenih treninga.

Za vlasnike:

- **Izveštaji:** Pružiti skup izvještaja o radu teretane temeljenih na prikupljenim podacima, uključujući izvještaje o članovima, iskorištenosti termina i učinkovitosti trenera.

Dijagram obrasca uporabe može se vidjeti na slici 2.1.:



Slika 2.1: Dijagram obrasca uporabe

2.4. Ne-funkcionalni zahtjevi

Aplikacija mora biti jednostavna za korištenje kako bi ju i manje tehnički spretni korisnici znali koristiti. Sustav mora pružati pouzdane informacije u stvarnom vremenu. Također, potrebno je osigurati visoku razinu sigurnosti kako bi se zaštitili osobni podaci članova.

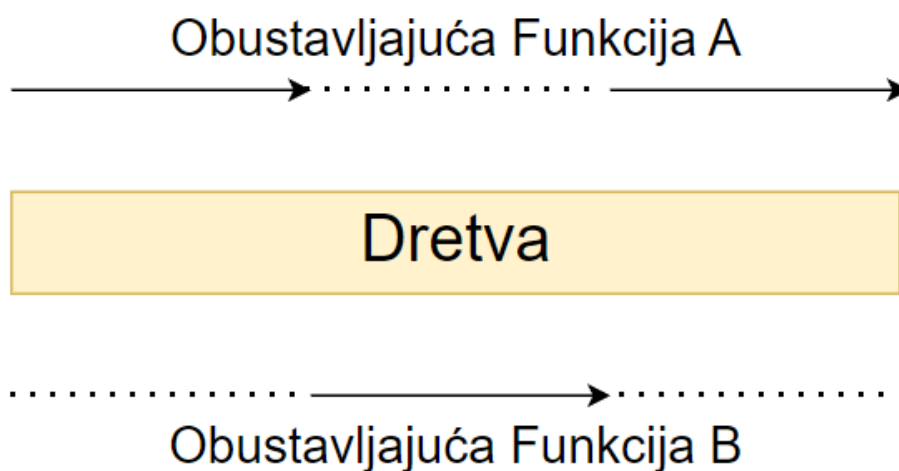
3. Tehnologije

3.1. Kotlin

Kotlin je moderan, statički tipiziran programski jezik dizajniran da bude u potpunosti interoperabilan s Javom s kojom dijeli jako puno sličnosti. Kao i Java, Kotlin se može prevesti u *bytecode* koji JVM razumije, što ga čini prenosivim, odnosno može se obavljati na različitim platformama (engl. *cross-platform*). Jedna od glavnih ideja iza razvoja Kotlina jest bila da pospješi nedostatak konciznosti koji programskom jeziku Java nedostaje. Upravo taj nedostatak konciznosti je problem koji uzrokuje mnoge prepreke u produktivnosti programera. Od mnogo novih tehničkih značajki Kotlina naspram Jave, potrebno je izdvojiti nekolicinu.

- 1. Sigurnost nedostajućih vrijednosti (eng. *Null-safety*)** – *Null-safety* jedna je od najvažnijih značajki Kotlina. Radi se o ugrađenoj sigurnosti od *null* vrijednosti kako bi se spriječile *Null pointer* iznimke koje su u Javi čest izvor neočekivanih grešaka. Kotlin rješava taj problem razlikujući *nullable* od *non-nullable* podataka za vrijeme kompilacije, odnosno, zahtjeva od programera da eksplicitno označe koje varijable mogu sadržavati *null* vrijednosti, a također uvodi i dodatne operatore koji olakšavaju rad s *nullable* tipovima podataka. Ovo uvelike smanjuje vjerojatnost pojave iznimki uzrokovanih *null* vrijednostima.
- 2. Inferencija tipova (eng. *type inference*)** – Inferencija tipova jest mogućnost prevodioca (engl. compiler) da automatski tijekom kompilacije može zaključiti tip varijable na temelju dodijeljenih vrijednosti. To pridonosi konciznosti programskog jezika Kotlin tako što programeri više nisu obavezni eksplicitno definirati tip varijable pri njihovoj deklaraciji.
- 3. Ekstenzijske funkcije (eng. *extension functions*)** – Jedna od značajki koja itekako pridonosi konciznosti koda jesu upravo ekstenzijske funkcije. Ekstenzijske funkcije su funkcije koje program može pisati poput proširenja funkcionalnosti postojećih klasa bez izmjene njihovog izvornog koda, što naravno omogućuje čišći i modularniji kod dodavanjem novih funkcionalnosti na vrlo jednostavan način. Pomoću njih možemo, na primjer, proširiti funkcionalnost *String* klase svojim kodom, pa tako te metode onda možemo pozivati na svakoj instance *String* klase.

4. Korutine (eng. *coroutines*)- Korutine su Kotlinov alat za upravljanje asinkronim programiranjem koji omogućava jednostavno i učinkovito izvođenje pozadinskih zadataka bez blokiranja glavne niti. Bitno je spomenuti da je korutina instanca obustavljajućih (eng. *suspending*) funkcija. Gotovo svaka funkcija se može eksplicitno označiti kao obustavljajuća, a jedina stvar koja obustavljajuću funkciju čini drugačijom od obične, jest to da one imaju jedan ili više obustavljajućih točaka (eng. *suspension points*), odnosno naredbe u tijelu funkcije koje mogu pauzirati izvođenje kako bi se ono nastavilo u kasnijem trenutku. Koncept korutine vrlo je sličan već dobro poznatom konceptu dretve (eng. *thread*) u tradicionalnom paralelnom programiranju. Bitna razlika jest da se, za razliku od dretvi, kontekstno prebacivanje ne događa na razini operativnog sustava. Umjesto toga, korutina radi kontekstno prebacivanje unutar same aplikacije kada dođe do obustavljajuće točke unutar *obustavljajuće* funkcije, a to je moguće upravo zato što korutine nisu vezane za neku specifičnu dretvu, već nakon zaustavljanja svoje izvođenje mogu nastaviti na bilo kojoj drugoj slobodnoj dretvi. To znači da, umjesto da se oslanja na operativni sustav za prebacivanje između različitih zadataka, korutina kontrolira ovaj proces sama, što omogućava lakše i učinkovitije upravljanje resursima. Kada se kontekstno prebacivanje dešava u interakciji s operativnim sustavom, svaki put kada OS prebacuje između dretvi, potrebno je spremi stanje trenutne dretve i učitati stanje nove dretve, a to uključuje niz drugih premještanja unutar memorije i registara koji troše vrijeme i procesorske cikluse. S druge strane, korutine koriste lagano (eng. *lightweight*) prebacivanje konteksta koje je dio same aplikacije, a kojeg izvodi *Kotlin Runtime*. Kada korutina dosegne *suspension point*, *Kotlin runtime* jednostavno pauzira izvršavanje te korutine, spremi njeno trenutno stanje i nastavi izvršavanje druge korutine ili nastavak glavne niti. Konceptualni prikaz navedenog se može vidjeti na slici Slika 3.1



Slika 3.1: Prikaz rada obustavljajućih funkcija

3.2. Spring Framework

Spring Framework je vrlo raznolik radni okvir (eng. *framework*) za izgradnju aplikacija na tehnologijama baziranim na JVM-u . Glavni principi Springa su Inverzija kontrole (eng. *Inversion of Control*) te injekcija ovisnosti (eng. *Dependency injection*).

- **Inverzija kontrole** – Inverzija kontrole je oblikovni princip u kojem je kontrola nad životnim ciklusom objekata (eng. *object lifecycle*) ili nad djelom koda, prenesena radnom okviru.
- **Injekcija ovisnosti** – Injekcija ovisnosti je oblikovni obrazac kojim se implementira inverzija kontrole, a uključuje propuštanje (injekciju) ovisnosti klase iz njene okoline, odnosno radnog okvira, umjesto da klasa svoje ovisnosti sama kreira.

Inverzija kontrole u Springu postignuta je pomoću komponente naziva *IoC Container* (*Inversion of Control Container*). Ta je komponenta zaslužna za upravljanje konfiguracijom i životnim ciklusom objekata aplikacije. Objekti kojima taj kontejner upravlja nazivaju se *Beans*. Tijekom pokretanja programa, Spring će skenirati naš kod, te će pomoću anotacija koje su namijenjene za označavanje *Beanova*, shvatiti koje sve objekte mi delegiramo Spring kontejneru. Kontejner će te *Beanove* instancirati i inicijalizirati te koristiti oblikovni obrazac jedinstveni objekt (eng. *Singleton*) kako bi u kontejneru imao samo jednu instancu svake klase čiji smo životni ciklus delegirali. Injekcija ovisnosti u Springu je omogućena pomoću nekolicine anotacija, a najbitnija je *@Autowired* koja predstavlja znak Springu da bi na ovo mjesto trebao obaviti proces naziva samoožičenje (eng. *Autowiring*). Samoožičenje je slanje

jedinstvenih instanci razredu kojemu su te instance potrebne, od strane Spring kontejnera. Osim baznog Spring radnog okvira postoji mnoštvo projekata unutar Spring ekosustava, a potrebno je spomenuti neke najvažnije.

- **Spring Boot** – Spring boot je ekstenzija Spring radnog okvira koja pojednostavljuje te ubrzava postavljanje, konfiguraciju i razvoj aplikacija koje koriste Spring radni okvir. Automatska konfiguracija izuzetno olakšava i pojednostavljuje konfiguraciju aplikacije na temelju ovisnosti koje su prisutne u projektu tako što eliminira potrebu za ručnom konfiguracijom i smanjuje količinu konfiguracijskih datoteka. Također, omogućava nam korištenje takozvanih Spring Boot Starter paketa, koji su zapravo samo prikladno zapakirane i konfigurirane često zajedno korištene biblioteke.
- **Spring Data** – Spring Data je jedan od projekata u Spring ekosustavu, a čija je glavna uloga da pruža jednostavan i konzistentan način pristupa bazama podataka (eng. *databases*). Važan dio ovog projekta jest i Spring Data JPA koji koristeći JPA (*Java Persistence API*) automatski upravlja povezivanjem i konfiguracijom ORM (*Object-Relational Mapping*) sloja, omogućavajući rad s bazama podataka. Također, Spring Data JPA koristi koncept repozitorija (eng. *repository*) koji predstavljaju apstrakciju za rad nad podacima baze podataka, ali i omogućavaju automatsko generiranje osnovnih CRUD (*Create-Read-Update-Delete*) operacija na temelju definicija entiteta (eng. *entity*) nad kojima obavljaju operacije.
- **Spring Security** – Spring Security visoko je prilagodljiv okvir za različite sigurnosne aspekte Spring aplikacija. Za autentifikaciju pruža mnoštvo različitih implementaciji uključujući osnovnu autentifikaciju (eng. *basic authentication*), JWT autentifikaciju (*Json Web Token*), LDAP autentifikaciju i druge. Osim toga, Spring Security omogućava i autorizaciju, odnosno definiranje pravila pristupa resursima unutar aplikacije, a definiraju se na način da korisnici mogu imati različite uloge (eng. *roles*) te privilegije (eng. *privileges*) na temelju kojih je pristup ograničen. Osim autentifikacije i autorizacije, Spring Security još pruža i niz drugih pogodnosti kao što je zaštita od uobičajenih sigurnosnih prijetnji (na primjer CSRF i XSS) ili jednostavna integracija s različitim sigurnosnim protokolima kao što je OAuth2.

Spring Framework, sa svojim širokim spektrom funkcionalnosti, ustaljenosti u industriji i vrlo aktivnom zajednicom, predstavlja važan alat za razvoj modernih, sigurnih i kvalitetnih aplikacija baziranih na JVM tehnologijama.

3.3. Android

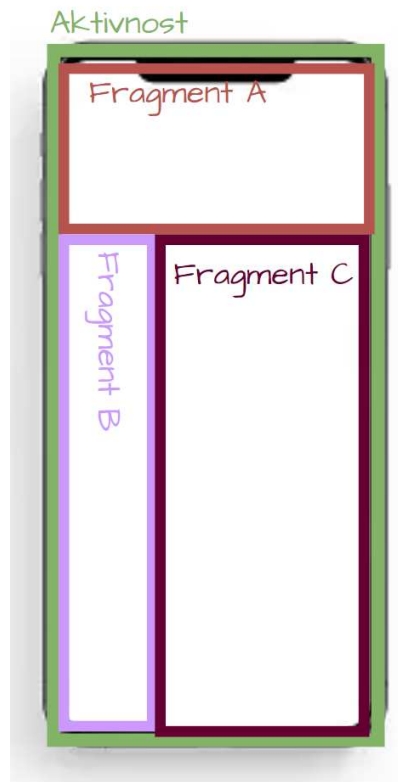
Android je jedan od najrasprostranjenijih i najpopularnijih operativnih sistema u različitim uređajima poput mobitela, hladnjaka, satova pa čak i automobila. Prema podacima iz 2024. godine, čak 70.69% mobilnih uređaja koristi Android kao operativni sistem (Backlinko, 2024.). A s obzirom da je i nekolicina drugih operativnih sistema napravljena na bazi Androida, neupitno je da je Android vrlo bitna tehnologija koju je potrebno razumjeti. S programerske strane, Google, proizvođač Androida, nam nudi Android komplet za razvoj softvera koji nam omogućava efikasan i brz razvoj aplikacija za Android OS. Android je također vrlo siguran OS. Naime, svaka Android aplikacija živi u svom sigurnosnom pješčaniku (eng. *security sandbox*) u kojem ih čuva nekoliko sigurnosnih značajki. Jedna od tih jest da je Android OS zapravo više-korisnički Linux sustav u kojoj svaka aplikacija ima ulogu korisnika u Linux sustavu, te sistem svakoj aplikaciji pridiželi korisnički identifikator uz kojeg sistem veže dozvole (eng. *permissions*) nad datotekama aplikacije, čime osigurava da datotekama aplikacija može pristupiti samo ta aplikacija čije su datoteke. Osim toga, svaki proces ima svoju virtualnu mašinu što omogućava da se kod jedne aplikacije izvršava u izolaciji od koda druge aplikacije.

Korisničko sučelje Androida sagrađeno je kroz dvije ključne komponente koje čine temelj za razvoj bilo koje Android aplikacije, a to su aktivnost (eng. *activity*) i fragment (eng. *fragment*)

Aktivnost je osnovna komponenta svake Android aplikacije koja predstavlja ekran s kojim korisnik može komunicirati. Svaka aplikacije ima barem jednu aktivnost, a vrlo često ih imaju i više. Nadalje, svaka aktivnost igra važnu ulogu u interakciji aplikacije i Android sistema tako što, među ostalim, prati s kojim komponentama korisnik trenutno komunicira, kako bi sistem znao koje procese mora držati živima. Također, svaka aktivnost, kao i mnogo drugih komponenti unutar Android sistema, ima životni ciklus u kojemu prolazi kroz faze kreiranja, pokretanja, pauziranja i uništavanja, koje je svakom programeru bitno razumjeti kako bi znali efikasno upravljati resursima aplikacije.

Fragmenti su još jedna vrlo važna komponenta svake Android aplikacije a predstavljaju ponovno upotrebljive dijelovi svake aktivnosti te omogućuju dinamičnije korisničko sučelje. Fragmenti su usko vezani za aktivnosti na način da s njima usklađuju

vlastiti životni ciklus, toliko da ne mogu postojati u izolaciji. Unutar jedne aktivnosti može postojati mnoštvo fragmenata, no fragmenti se kao ponovno upotrebljive jedinice mogu koristiti i u drugim aktivnostima te nisu vezani samo za jednu, a uz to da svoje stanje mogu pratiti i zadržati kroz više aktivnosti, smatra ih se idealnim alatom za stvaranje složenih i responzivnih korisničkih sučelja.



Slika : Prikaz aktivnosti i fragmenta

Objašnjenje tablica:

- **users** – Pohranjuje osnovne informacije o korisnicima sustava, uključujući njihovo ime, prezime, email adresu, link na kojem se nalazi njihova profilna slika te token (*fcm_token*) koji je jedinstveni identifikator korisnika na *FCM (Firebase Cloud Messaging)* servisu zaduženom za slanje *push* notifikacija.
- **gym_users** – Predstavlja korisnika teretane. Korisnik teretane može biti vlasnik, trener ili član.
- **roles** – Pohranjuje ime role koja se koristi pri autorizaciji zahtjeva pristupa resursima.
- **gym_user_roles** – Predstavlja rezultat *many-to-many* veze između *gym_users* i *roles* tablica.
- **gym_members** – Predstavlja člana teretane, a za njega pohranjujemo podatak o tome kada se prvi put pridružio teretani, njegov privatni kod kojim se njegovi prijatelji mogu pridružiti istoj teretani gdje je i taj član, te identifikator koji se koristi pri naplati članstva i koji je potreban servisu *Stripe*.
- **gym_trainers** – Predstavlja trenera u teretani.
- **gyms** – Pohranjuje podatke o teretani uključujući ime, sliku, kod kojim se članovi mogu pridružiti teretani, identifikator koji predstavlja račun teretane na *Stripe* servisu, podatak o cijeni članarine te identifikator cijene članarine koje je potreban za *Stripe* servis.
- **access_codes** – predstavlja jedinstveni kod koji vlasnik teretane može generirati kroz aplikacijsko sučelje, a pomoću kojeg se u teretanu korisnik može pridružiti u ulozi trenera.
- **gym_visits** – Pohranjuje podatke o dolascima pojedinog člana u teretanu, konkretno, datum, vrijeme te trajanje posjeta teretani.
- **gym_classes** – Pohranjuje podatke o generičkom terminu vođenih treninga. Svaki trening vođen je od strane jednog trenera i može imati određeni broj participanata. Podaci koje pohranjuje su ime, opis, trajanje u minutama, maksimalni broj participanata, datum i vrijeme, zastavica koja govori jeli generički trening izbrisan te zastavicu koja predstavlja jeli generički trening ponavljajući. Ako je ponavljajući, znači da će za taj generički *gym_class* biti vezano više instanci iz *gym_class_instance* tablice, a ako nije, tada će postojati samo jedan povezani *gym_class_instance*.

- ***recurring_patterns*** – Predstavlja uzorak ponavljanja ponavljajućih treninga, a pohranjuje podatak o tome koliko termina treninga istog generičkog treninga će se ponoviti.
- ***days_of_week*** – Predstavlja dan u tjednu u brojčanom obliku (1 - 7).
- ***recurring_patterns_days_of_week*** – Tablica potrebna za uspostavljanje *many-to-many* veze između *recurring_patterns* i *days_of_week* tablice.
- ***gym_class_instances*** – Predstavlja određenu instancu treninga na koju se participanti mogu prijaviti, odnosno predstavlja jedan termin treninga, a pohranjuje datum i vrijeme kada se termin odvija. Sve ostale informacije o terminu dohvatljive su preko referenciranog generičkog treninga (*gym_class*).
- ***gym_member_classes*** – Tablica nužna kako bi se ostvarila *many_to_many* veza između termina treninga i člana teretane.
- ***gym_class_modified_instances*** – Predstavlja one instance ponavljajućih treninga koje su mijenjanje, a pohranjuje sve podatke koje je moguće mijenjati, dakle datum i vrijeme, opis, trajanje, identifikator trenera, maksimalni broj participanata te zastavicu *is_canceled* koja predstavlja jeli upravo taj termin treninga otkazan.
- ***gym_class_reviews*** – Predstavlja recenzije termina treninga od strane članova koji su bili participanti tih termina. Pohranjuje datum i vrijeme ostavljanja recenzije, ocjenu (broj od 1 do 5) te samu recenziju.
- ***gym_trainer_reviews*** – Konceptualno isto kao spomenuti *gym_class_reviews*, no recenzija se odnosi na trenera koji je vodio termin. Također pohranjuje datum i vrijeme ostavljanja recenzije, ocjenu te samu recenziju.
- ***challenges*** – Predstavlja izazove koje postavlja vlasnik teretane, a kojima članovi mogu skupljati bodove. Pohranjeni su opis, ime, datum isteka, tip izazova (vremenski, frekvencijski ili pozivajući), broj bodova koji se dobije rješavanjem izazova te zastavicu *is_deleted* koja predstavlja jeli taj izazov izbrisan od strane vlasnika teretane.
- ***challenge_criterias*** – Predstavlja generički kriterij koji mora biti ispunjen da bi se izazov smatrao riješenim, a pohranjuje tip kriterija koji može biti frekvencijski ili vremenski. Ukoliko uz izazov nije vezan niti jedan kriterij, znači da se radi o pozivajućem izazovu, odnosno o izazovu koji je riješen kada se novi član učlani u teretanu koristeći *invite_code* već postojećeg člana te teretane.

- ***time_based_criterias*** – Pohranjuje podatke o vremenskom tipu kriterija, a podaci su početno vrijeme u danu od kad izazov vrijedi te krajnje vrijeme u danu do kad izazov vrijedi. Vremenski se izazov može smatrati riješenim ako član unutar vremenskog perioda kriterija posjeti teretanu na određeni dan. Svaki dan, izazov se može iskoristiti samo jednom.
- ***frequency_based_criteria*** – Predstavlja frekvencijski tip kriterija, a pohranjuje podatak o broju dolazaka potrebnih unutar jednog mjeseca da bi se izazov smatrao riješenim.
- ***member_challenge*** – Predstavlja riješeni izazov od strane člana teretane, a pohranjuje datum i vrijeme kada je izazov riješen, te zastavicu *is_claimed* koja daje informaciju o tome jeli član prihvatio bodove kroz aplikaciju.

5. Opis rješenja

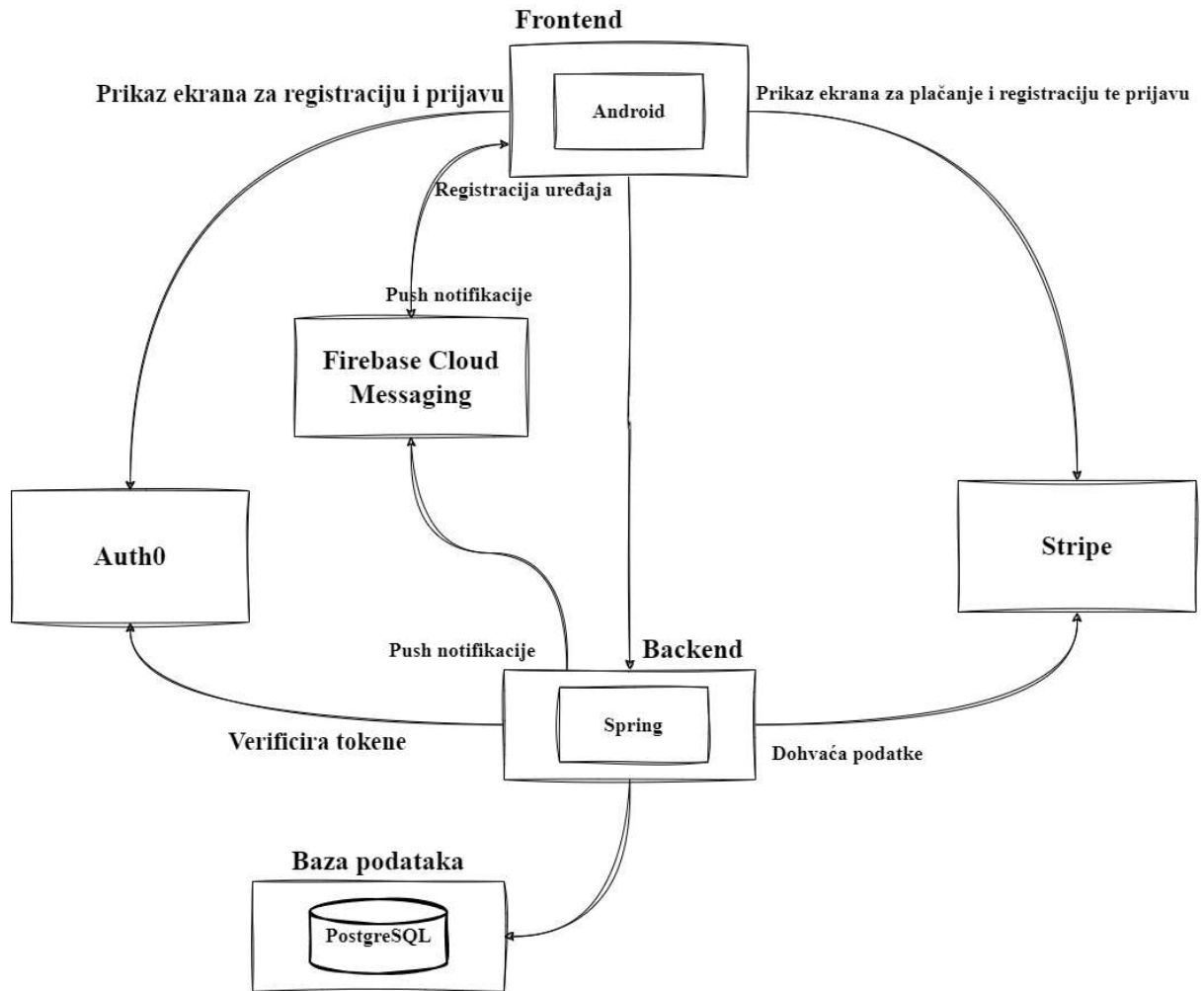
5.1. Arhitektura sustava

Razvijena aplikacija sveobuhvatno je programsko rješenje koje je dizajnirano da bi zadovoljilo potrebe vlasnika, trenera te članova teretana, a čija je glavna svrha omogućiti jednostavno upravljanje teretanom od strane vlasnika, upravljanje treninzima od strane trenera te pregled aktivnosti i drugih mogućnosti koje aplikacija nudi od strane članova.

Tehnologije korištene za razvoj aplikacije uključuju Android SDK (eng. *Android Software Development Kit*) koji nudi programerima skup alata i biblioteka koje omogućuju brz i efikasan razvoj Android aplikacija, Kotlin kao koncizan i efikasan jezik za razvoj na Androidu te Jetpack Compose za izradu grafičkog sučelja. Jetpack Compose moderan je alat za izradu Android grafičkog sučelja, a glavni razlog njegovog nastajanja jest kako bi se s imperativnog pristupa definiranja sučelja korištenjem XML datoteka prešlo na brži, i čišći deklarativni pristup pojednostavljujući i ubrzavajući razvoj korisničkih sučelja. *Backend* aplikacija razvijena je koristeći Spring Framework te Kotlin, dok je za bazu podataka korišten PostgreSQL, a osim navedenih osnovnih tehnologija, korišteni su i neki vanjski servisi:

- **Auth0** - Korišten za autentifikaciju i autorizaciju korisnika. Pri prijavi (eng. *sign in*) ili registraciji (eng. *sign up*), mobilna aplikacija koristi Auth0 kako bi osigurala sigurnu autentifikaciju kroz web preglednik. Auth0 vraća JWT (*JSON Web Token*) koji se pohranjuje na mobilnom uređaju i koristi se pri svakom zahtjevu prema *backendu* za validaciju i autorizaciju.
- **Stripe** - Integriran za upravljanje plaćanjima. Koristi se za postavljanje načina plaćanja, procesiranje uplata i upravljanje pretplatama unutar aplikacije. Stripe osigurava sigurno i pouzdano rukovanje svim financijskim transakcijama.
- **Firebase Cloud Messaging (FCM)**: Koristi se za slanje *push* notifikacija korisnicima. Prilikom registracije, *frontend* aplikacija registrira uređaj na FCM servisu i dobiva jedinstveni FCM token koji se koristi za slanje notifikacija putem *backend* sustava.

Arhitektura sustava prikazana je na slici Slika 5.1.

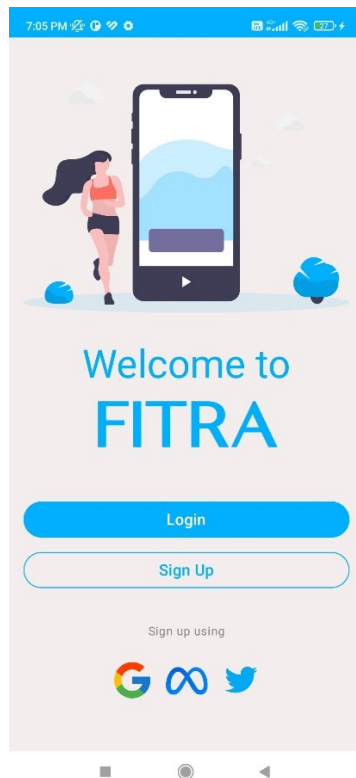


Slika 5.1: Dijagram arhitekture sustava

Slika prikazuje sve velike komponente sustava, a one međusobno komuniciraju na sljedeći način. Mobilna aplikacija komunicira sa serverom za dohvat i ažuriranje podataka. Također komunicira s Auth0 za autentifikaciju i autorizaciju korisnika, te sa Stripe servisom za upravljanje plaćanjima. Također, mobilna aplikacija registrira uređaj na FCM-u za primanje *push* notifikacija. Backend upravlja poslovnom logikom, validacijom tokena i upravljanjem podacima, komunicira s Auth0 za validaciju JWT tokena, sa Stripeom za procesiranje uplata, te s FCM-om za slanje *push* notifikacija korisnicima. Također, *backend* komunicira s PostgreSQL bazom podataka za pohranu i dohvat podataka.

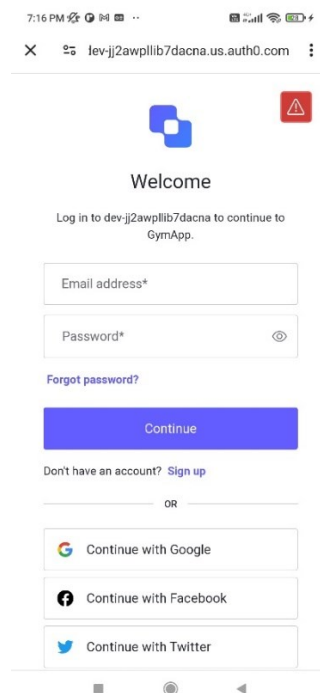
5.2. Funkcionalnosti aplikacije

Ekran koji korisnik aplikacije vidi pri prvom ulasku u aplikaciju prikazan je slikom Slika 5.2



Slika 5.2: Prikaz početnog ekrana

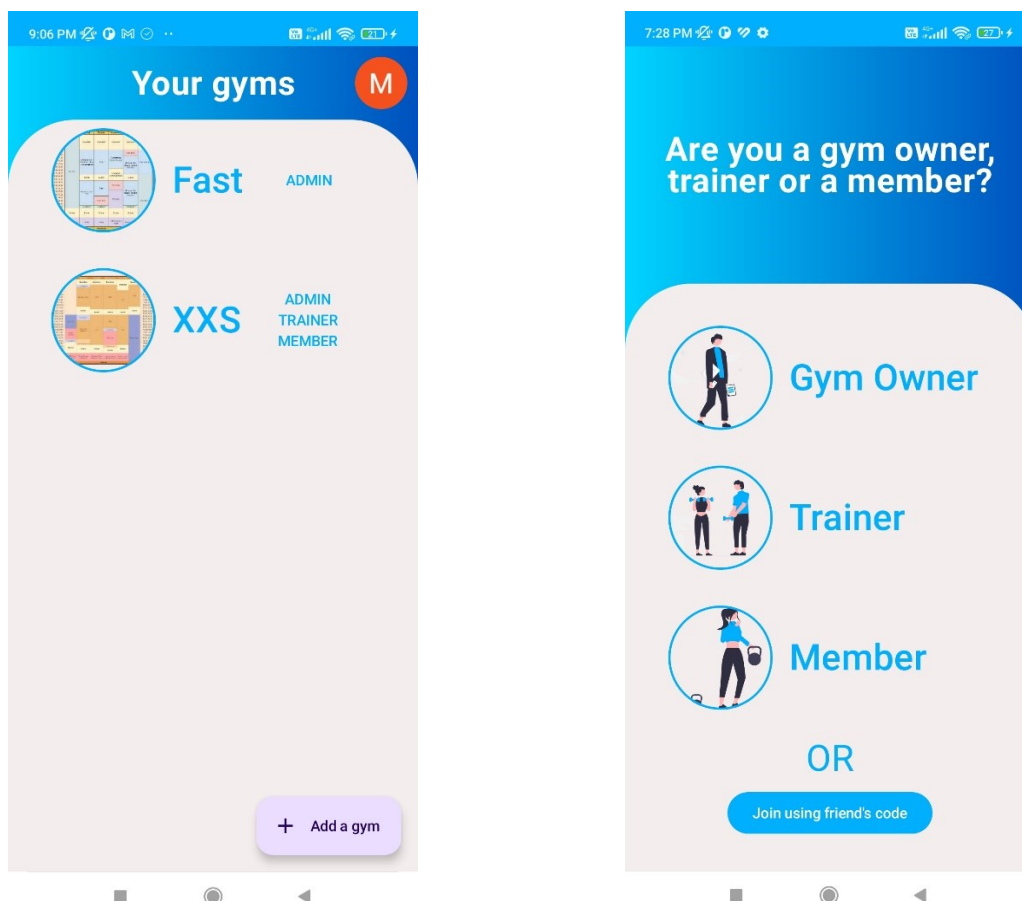
Pritiskom na neki od gumbova, biva preusmjeren na registraciju ili prijavu u već postojeći račun putem Auth0 servisa kroz web preglednika. Spomenuto se može vidjeti na slici Slika 5.3



Slika 5.3: Prikaz prijave putem Auth0 servisa

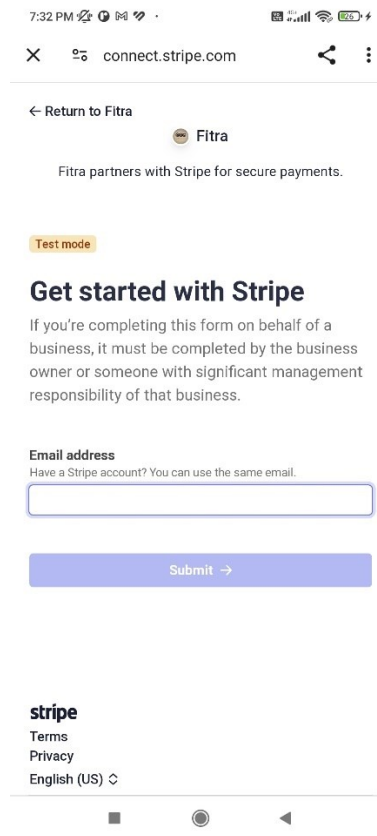
Rezultat registracije ili prijave putem Auth0 servisa jest *Credentials* objekt, iz kojega možemo izvući nekoliko zanimljivih podataka, a jedan od njih je *idToken*, koji je zapravo JWT token koji u sebi ima nekolicinu podataka o prijavljenom korisniku, a to su identifikator korisnika, email, ime i prezime te link na profilnu sliku prijavljenog korisnika. Osim toga, iz *Credentials* objekta dobije se još jedan bitan token a to je *accessToken* koji je potreban kako bi se prijavljeni korisnik uspješno autorizirao za dohvat resursa na *backendu*. Kako se korisnik ne bi trebao prijavljivati iznova svaki put kada ulazi u mobilnu aplikaciju, *accessToken* sprema se lokalno na korisnikovom uređaju za daljnju upotrebu dokle god je *accessToken* aktivan, a aktivan je 24h, nakon čega se od Auth0 servisa zatraži novi token. Prikupljene informacije, kao i *accessToken* šalju se na *backend* kako bi se kreirao profil na aplikacijskoj strani. *accessToken* se na dalje treba slati u svakom zahtjevu kako bi se autoriziralo korisnika aplikacije.

Nakon što je korisnik prijavljen, nalazi se na ekranu na kojem se nalazi popis svih teretana kojih je član, kao i mogućnosti dodavanja novih teretana, bilo u ulozi, vlasnika, trenera ili običnog člana teretane. U ulozi vlasnika, teretanu dodajemo tako da upisujemo ime teretane, cijenu članarine te dodajemo sliku, u ulozi trenera upisujemo kod koji vlasnik teretane može jednokratno generirati koristeći svoju aplikaciju, a u ulozi člana upisujemo jedinstveni kod teretane koji je namijenjen da bude javno vidljiv kako bi ga članovi mogli upisati u aplikaciju. Također, moguće je i pristupiti teretani upisujući jedinstveni kod već postojećeg člana teretane, čime taj član rješava izazov i dobiva bodove. Navedeno možemo vidjeti na slici Slika 5.4



Slika 5.4: Prikaz popisa teretana i unosa nove teretane

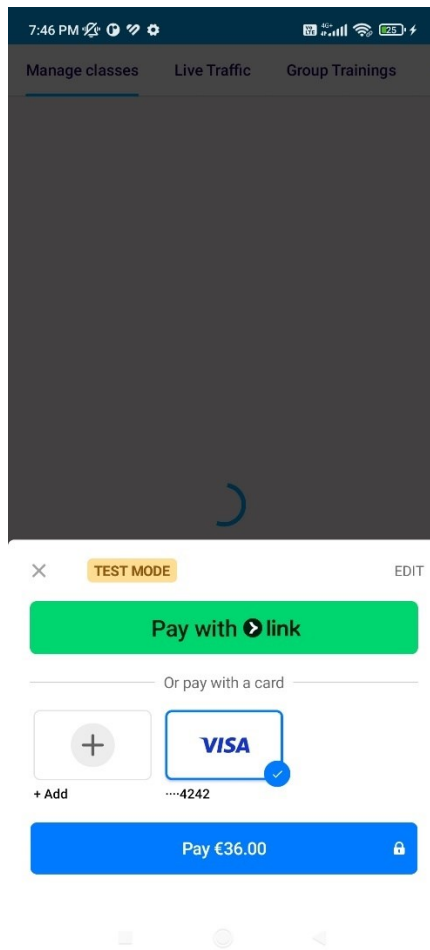
Nakon dodavanje teretane, ovisno o privilegijama koje korisnik ima, odnosno jeli vlasnik, trener ili član, korisnik vidi traku koja se može horizontalno pomicati na kojoj mu se prikazuju one opcije koje mu njegova privilegija dozvoljava. Također, početni ekrani za konkretnu teretanu su drugačiji s obzirom koju ulogu korisnik ima. No prije nego li može išta vidjeti i napraviti, ukoliko je korisnik vlasnik teretane, potrebno je registrirati se na Stripe servis koji mu omogućava primanje članarina na njegovu bankovnu karticu, a početni ekran procesa registracije na Stripe prikazan je slikom Slika 5.5.



Slika 5.5: Prikaz registracije na Stripe

Nakon registracije ili prijave na već postojeći Stripe račun, korisnik je vraćen na stranicu na kojoj je cijeli proces i započeo, odnosno na početnu stranicu teretane. Vlasnik teretane tada može posjećivati sve stranice namijenjene vlasnicima.

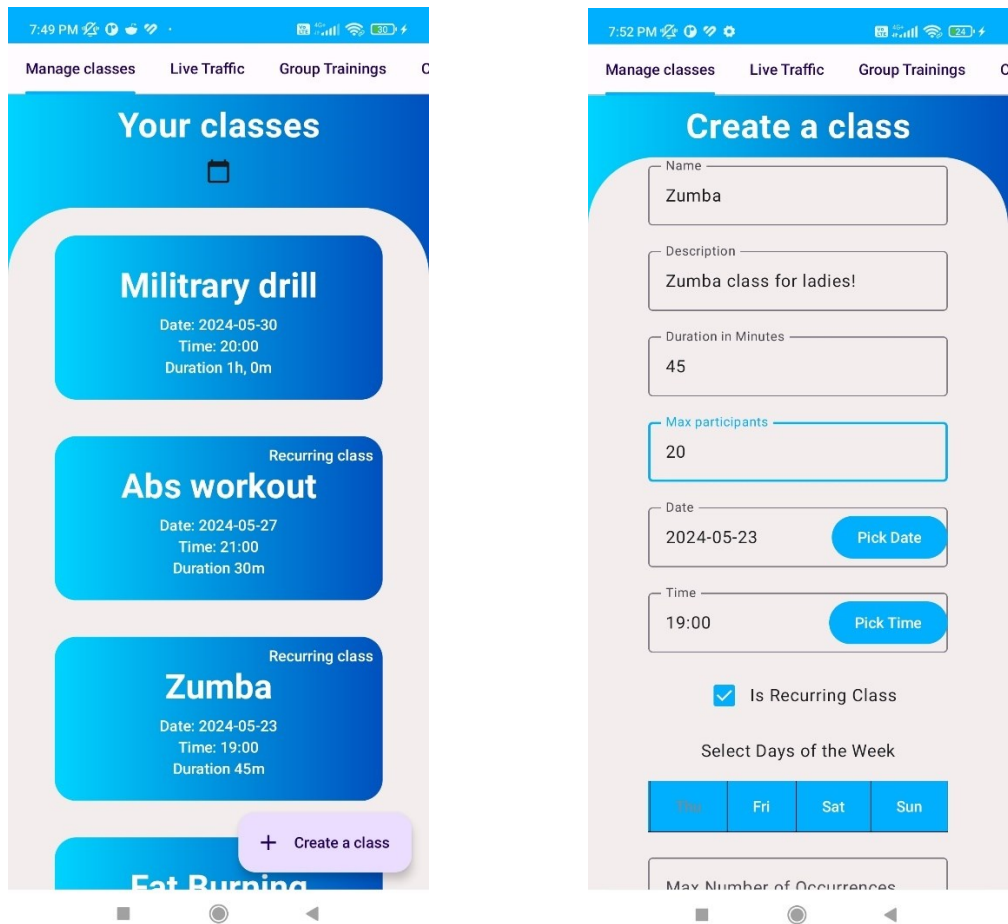
Osim vlasnika, kreirati i prijaviti se na Stripe dužan je i član teretane, kako bi mogao plaćati članarinu. Taj dio procesa ne odvija se kroz web pretražitelja, kao što je to do sada bio slučaj, već je Stripe integriran u mobilnu aplikaciju, a to se može vidjeti na slici Slika 5.6



Slika 5.6 Prikaz plaćanja članarine

Nakon plaćanja članarine član je 30 dana u mogućnosti vidjeti sve ekrane koji su mu namijenjeni, nakon isteka od 30 dana, ponuditi će mu se da ponovno plati članarinu, a njegovi kartični podaci biti će spremljeni i sigurno pohranjeni na Stripe servisu.

Spomenuti Stripe koristi se samo u slučajevima da je trenutno prijavljen korisnik vlasnik ili član teretane, no u slučaju trenera to nije tako. Kada se trener prvi put prijavi u teretanu koristeći *access_token* koji mu je vlasnik jednokratno generirao, odmah može krenuti pregledavati ekrane dostupne trenerima. Funkcionalnost koja je dostupna trenerima jest objavljivanje i pregledavanje planiranih termina treninga. Ekran dostupan trenerima koji pokazuje sve njihove treninge, odnosno treninge na kojima su oni treneri, kao i ekran koji prikazuje postupak izrade novog treninga može se vidjeti na slici Slika 5.7. Za svaki trening se na ekranu može vidjeti njegovo ime, datum i vrijeme te trajanje sljedećeg po redu termina treninga. Dakle u slučaju ponavljajućih treninga, pokazuju se informacije o onom terminu koji je u najbližoj budućnosti, a treninzi koji nisu ponavljajući i prošli su, ili ponavljajući ali su se svi njihovi termini već dogodili, ne prikazuju se.



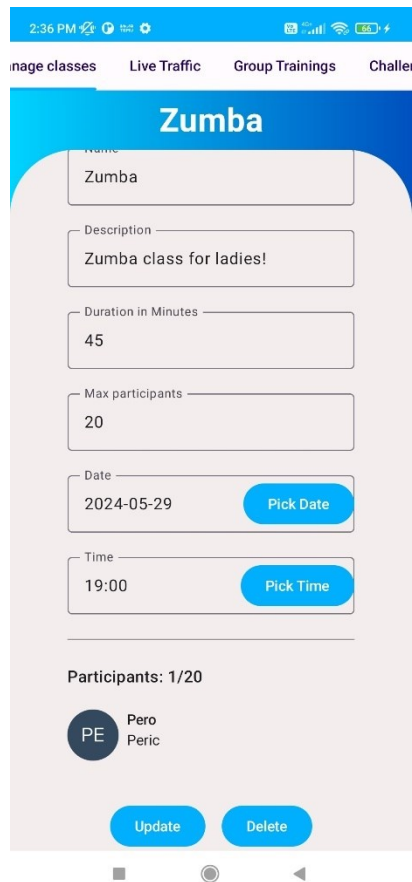
Slika 5.7: Prikaz trenerovih treninga te izrade novog treninga

Osim toga, radi lakšeg snalaženja i pregledavanja termina, moguće je otvoriti kalendar koji prikazuje sve termine pritiskom ikone kalendara na vrhu ekrana sa slike Slika 5.7, a pritiskom nekog treninga na istoj slici, otvara se prozor kroz koji možemo mijenjati pojedine podatke o treningu, izbrisati odnosno otkazati trening, te otvoriti kalendar, no samo sa upravo otvorenim treningom u kalendaru. Ukoliko je taj trening ponavljajući, pokazuju se svi predviđeni termini tog treninga, a ako nije, prikazuje se samo jedan. Navedeno se može vidjeti na slici Slika 5.8.



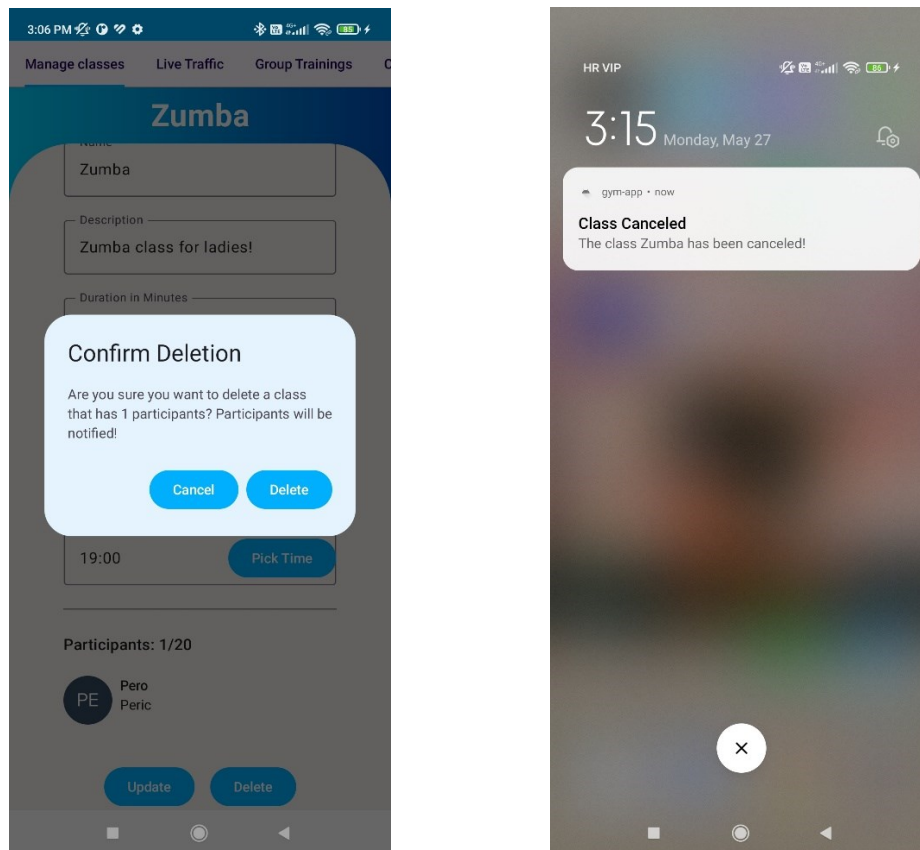
Slika 5.8: Prikaz kalendara

Pritiskom na neki od vizualnih elemenata određenog termina u pogledu kalendara, otvara se novi prozor koji prikazuje dodatne informacije o odabranom terminu treninga kao i mogućnost ažuriranja podataka te pregleda prijavljenih particijanata na taj termin. Također, kroz ovaj ekran izabrani se termin treninga može i otkazati. Navedeno se može vidjeti na slici Slika 5.9



Slika 5.9: Prikaz detalja o terminu treninga

Kada trener odluči otkazati termin treninga, pritiskom na gumb *Delete* pokreće se proces provjere prijavljenih članova. Ako su na termin već prijavljeni članovi, pojavljuje se *iskočni* prozor s porukom: "Želite li otkazati trening koji ima X prijavljenih korisnika? Ako da, korisnici će biti obavješteni." Ova obavijest daje treneru mogućnost da potvrdi ili odustane od otkazivanja termina. Ako trener potvrdi otkazivanje, termin se uklanja iz rasporeda, a svi prijavljeni korisnici bivaju obaviješteni putem *push* notifikacije poslane pomoću FCM servisa. Ova funkcionalnost osigurava da su članovi uvijek pravovremeno informirani o promjenama u rasporedu, što doprinosi boljem korisničkom iskustvu i pomaže u održavanju transparentnosti komunikacije između teretane i njenih članova. Na slici Slika 5.10 prikazan je *iskočni* prozor koji se pojavljuje prilikom pokušaja otkazivanja termina s prijavljenim članovima, te izgled obavijesti na korisnikovom mobilnom uređaju.



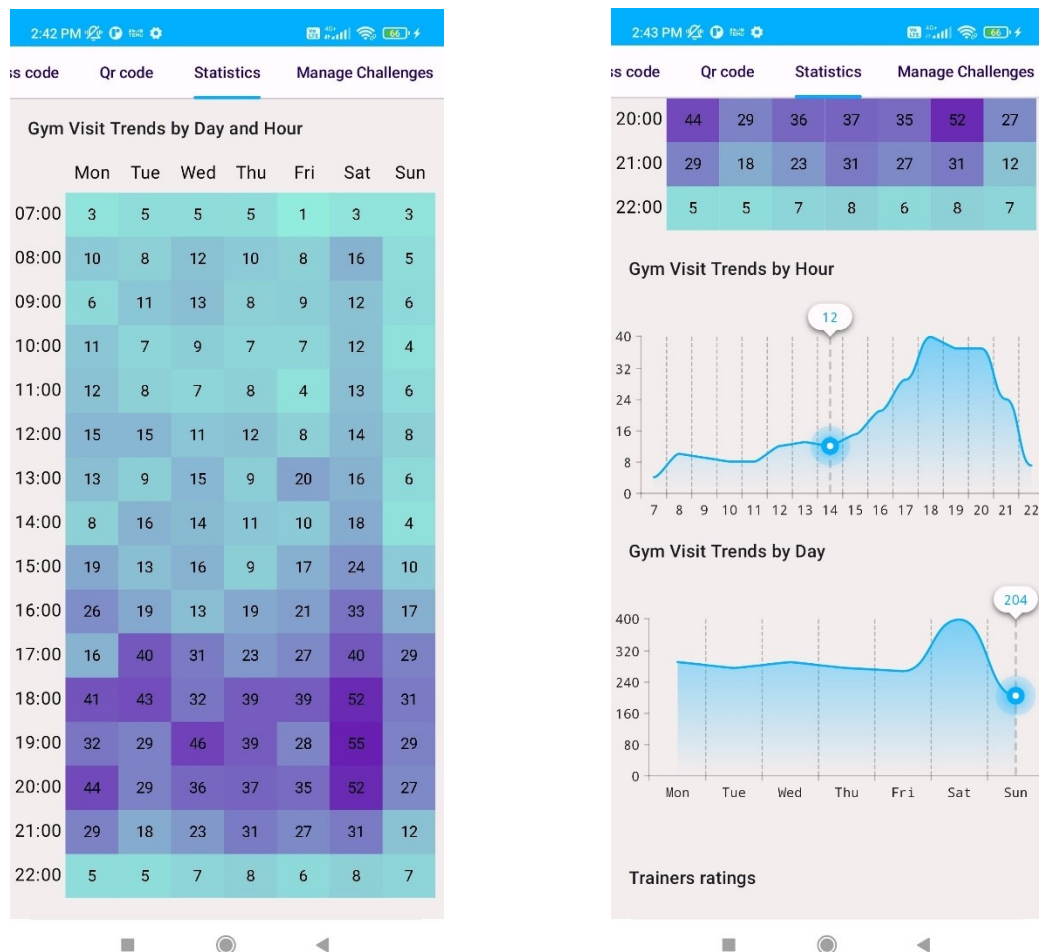
Slika 5.10: Prikaz iskočnog prozora te pristigle notifikacije

S druge strane, korisnici koji su vlasnici imaju drugačiju ulogu u vođenju teretane, što znači da imaju pristup različitim funkcionalnostima i ekranima u aplikaciji. Jedna od ključnih funkcionalnosti koja je potrebna za optimalno upravljanje teretanom i bolji uvid u njeno funkcioniranje je prikupljanje i prikazivanje statističkih podataka. Vlasnicima je bitno imati pristup informacijama o iskorištenosti različitih termina kako bi mogli donositi informirane odluke.

Primjerice, u teretanama je uobičajeno da većina članova dolazi na treninge u večernjim satima. Međutim, za učinkovito upravljanje teretanom, vlasnicima su potrebni precizni podaci o posjetama tijekom cijelog dana. Ovi podaci omogućuju vlasnicima da prepoznaju obrasce ponašanja članova i optimiziraju termine treninga kako bi bolje raspodijelili posjete. Na taj način mogu prilagoditi resurse, poput broja trenera ili rasporeda aktivnosti, kako bi izbjegli gužve u određeno vrijeme i poboljšali korisničko iskustvo.

Također je važno analizirati raspodjelu dolazaka članova po danima u tjednu. Ovi podaci pomažu vlasnicima razumjeti kada je teretana najposjećenija te mogu planirati posebne događaje ili promocije u manje zauzetim danima. Sve ove statistike i analize

prikazane su na detaljnom grafičkom prikazu, kao što je prikazano na slici Slika 5.11, što omogućava vlasnicima da vizualno interpretiraju podatke i donose strateške odluke temeljem njih.



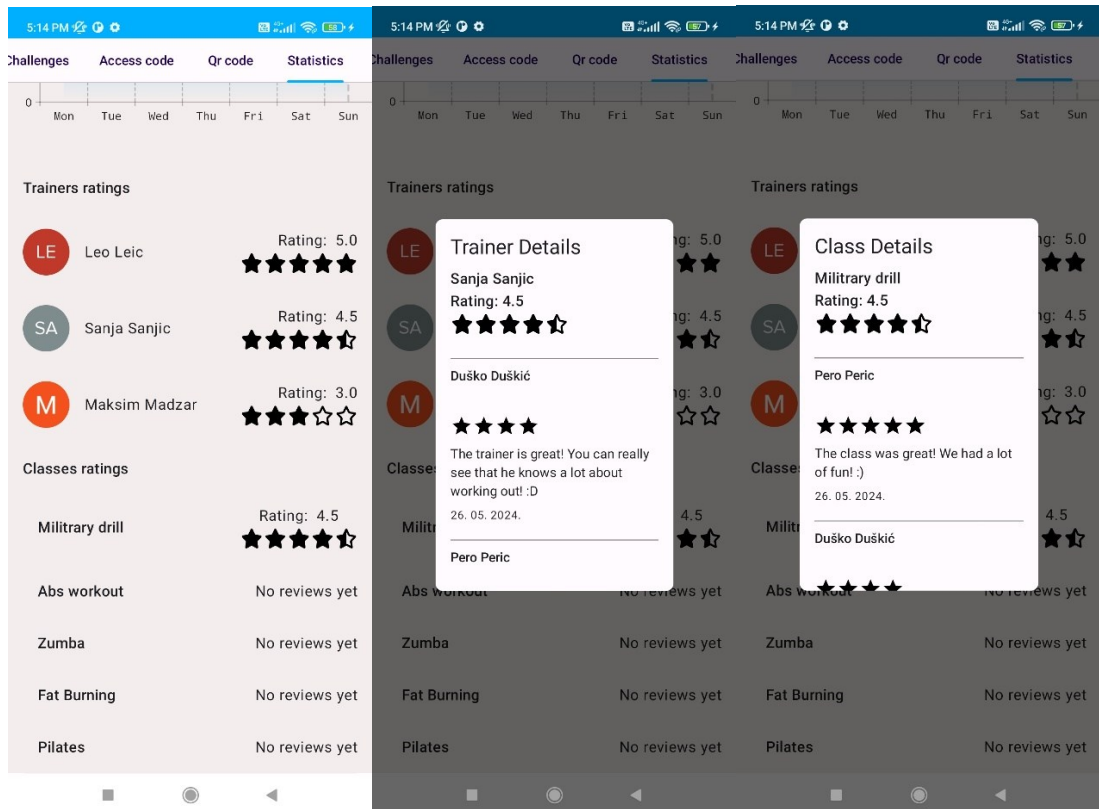
Slika 5.11: Prikaz statistike dolazaka u teretanu

Kombiniranjem ovih statističkih podataka s ostalim funkcionalnostima aplikacije, vlasnici teretana mogu značajno poboljšati operativnu učinkovitost i pružiti bolju uslugu svojim članovima, što dugoročno vodi do većeg zadovoljstva korisnika i uspješnijeg poslovanja.

Osim praćenja statistike vezane uz posjećenost, jednako je važno znati i kakve performanse imaju zaposlenici, odnosno treneri, te što članovi misle o njihovom radu. Zato je u aplikaciju uveden sustav recenziranja treninga i trenera. Nakon svakog odrađenog treninga, korisnicima se omogućuje da ocijene i daju povratne informacije o treningu i treneru. Takav sustav recenziranja pruža vlasnicima teretane dragocjene informacije o kvaliteti rada trenera i zadovoljstvu članova. Podaci dobiveni putem recenzija mogu se koristiti za prepoznavanje

najboljih trenera, identificiranje područja za poboljšanje i donošenje odluka o eventualnim edukacijama ili promjenama u programu. Također, recenzije pomažu u motivaciji trenera da kontinuirano unapređuju svoje vještine i pružaju kvalitetne usluge članovima.

Prikaz recenzija dostupan je u aplikaciji, a vizualni prikaz tih recenzija može se vidjeti na slici Slika 5.12. Korištenjem ovih podataka, vlasnici mogu dodatno optimizirati operacije teretane, poboljšati uslugu i osigurati visoku razinu zadovoljstva članova.



Slika 5.12: Prikaz recenzija trenera i treninga

Nakon uvođenja sustava recenziranja trenera i treninga, važno je napomenuti i funkcionalnost izazova koju aplikacija pruža kao alat za dodatnu motivaciju članova teretane. Ovi izazovi osmišljeni su kako bi potaknuli korisnike da češće dolaze u teretanu, pozivaju nove članove i pridonose smanjenju gužvi u najfrekventnijim terminima. Članovi skupljaju bodove ispunjavanjem izazova, a te bodove mogu iskoristiti za različite pogodnosti u teretani, poput besplatnih ili sniženih proteinskih napitaka, 1 na 1 treninga sa trenerima i slično.

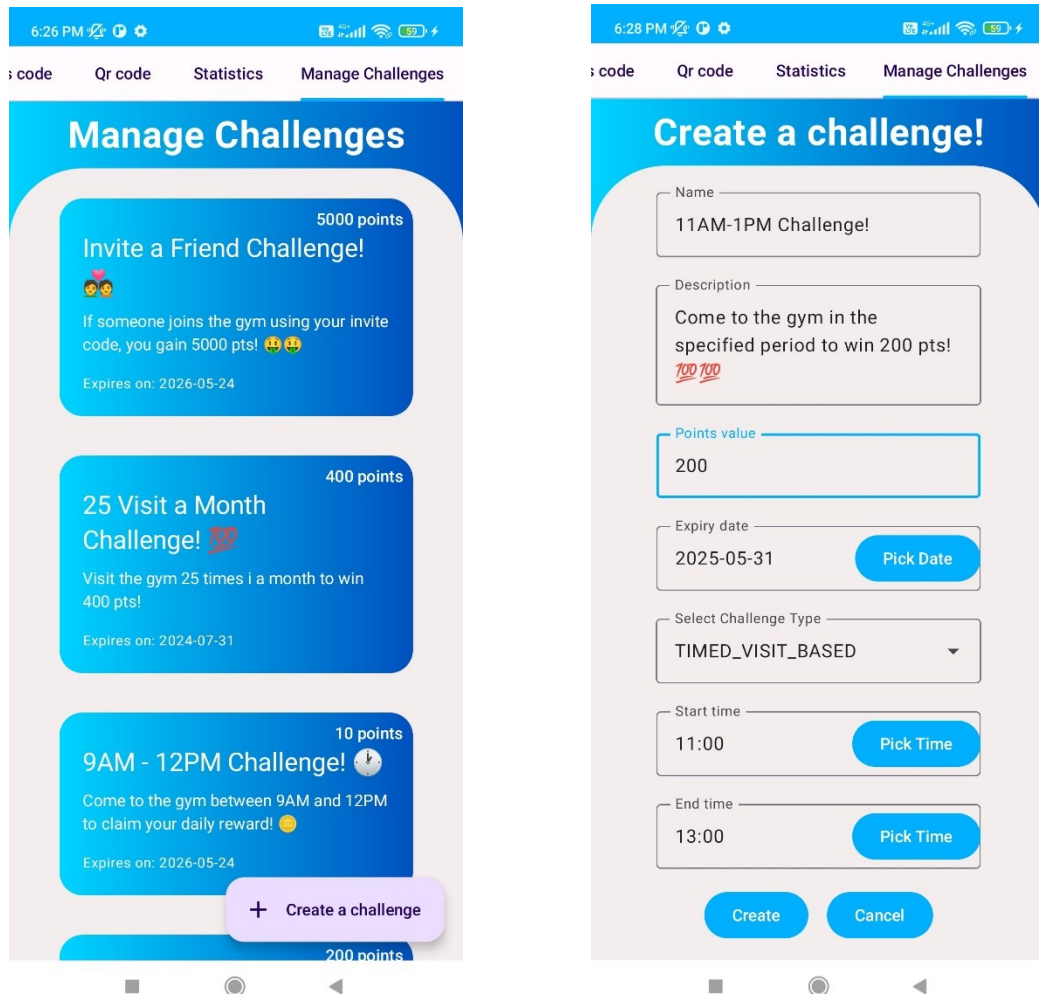
Postoje 3 vrste izazova čiju izradu aplikacija pruža:

- **Frekvencijski izazov** – Frekvencijski izazov postavlja kriterij da član, kako bi ga riješio, mora posjetiti teretanu određeni broj puta unutar mjesec dana. Ova

vrsta izazova motivira članove na redovito vježbanje, što može poboljšati njihovu fizičku kondiciju i zadovoljstvo uslugama teretane.

- **Vremenski izazov** - Ovaj izazov postavlja kriterij da član mora doći u teretanu u određenom vremenskom periodu unutar dana kako bi dobio bodove. Na primjer, član može biti nagrađen bodovima ako posjeti teretanu između 14:00 i 16:00 sati, kada je teretana obično manje posjećena. Ova vrsta izazova pomaže vlasnicima teretana da uravnoteže posjećenost tijekom dana, smanjujući gužve u najprometnijim terminima i poboljšavajući iskustvo svih članova.
- **Izazov pozivanja novih članova** - Ovaj izazov može se smatrati riješenim kada se novi član pridruži teretani koristeći osobni kod postojećeg člana. Stari član dobiva bodove za svakog novog člana koji se pridruži teretani koristeći njegov kod. Ovakva vrsta izazova potiče članove da aktivno sudjeluju u promociji teretane i dovedu nove članove, što može značajno pridonijeti rastu broja članova te povećanju prihoda.

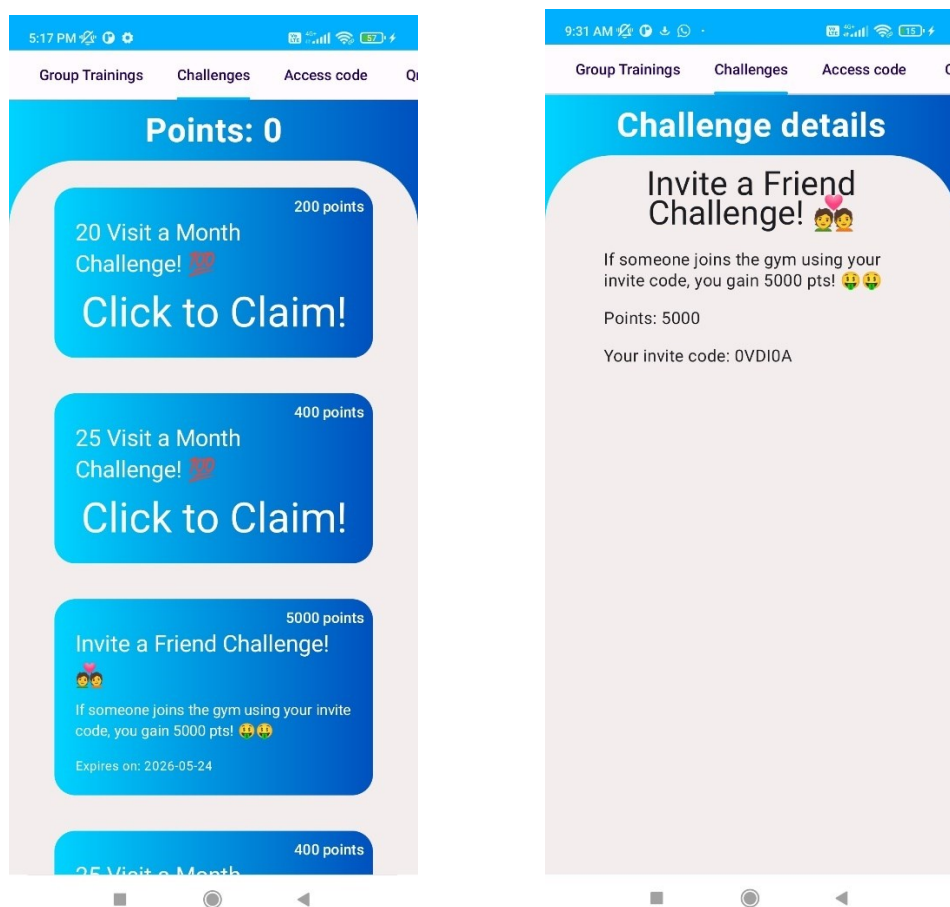
Vizualni prikaz izazova i njihove izrade može se vidjeti na slici Slika 5.13. Na ovom prikazu, vlasnik može vidjeti koji su izazovi aktivni u njegovoj teretani, klikom na gumb '*Create a challenge*' pojavljuje mu se ekran za izradu novog izazova, a klikom na bilo koji od aktivnih izazova prikazuje se ekran na kojemu vlasnik može mijenjati detalje izazova poput imena, opisa, datuma isteka te broja bodova, a na istom ekranu vlasnik može i zaustaviti izabrani izazov.



Slika 5.13: Prikaz aktivnih izazova te izrade novog izazova

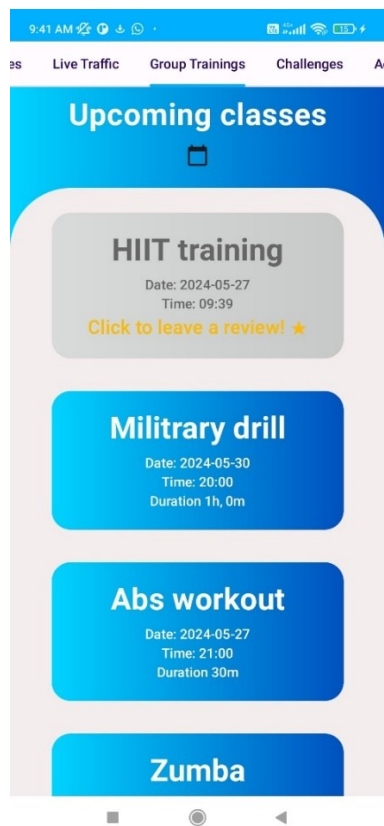
Nakon što smo objasnili sustav izazova i kako oni mogu motivirati članove teretane, važno je detaljno istu, ali i ostale funkcionalnosti sa strane članova teretane. Članovi mogu vidjeti popis dostupnih izazova, pratiti svoj napredak u izazovima te prihvatiti bodove za dovršene izazove. Svaki izazov je jasno prikazan s informacijama o tome koliko puta je potrebno posjetiti teretanu ili u kojem vremenskom razdoblju kako bi se izazov dovršio. Kada član dovrši izazov, može prihvatiti svoje bodove koji se zatim dodaju na njegov ukupni iznos bodova koji je također vidljiv na ekranu izazova, što članovima omogućava praćenje njihovog napretka i motivaciju za dalje sudjelovanje.

Navedene funkcionalnosti prikazane su na slici Slika 5.14. Također, ako je aktivan 'Invite a Friend' izazov, član može kliknuti na taj izazov kako bi dobio detaljne informacije, uključujući svoj jedinstveni pozivni kod. Ovaj kod članovi mogu dijeliti s prijateljima, a za svakog novog člana koji se pridruži teretani koristeći taj kod, postojeći član dobiva bodove.



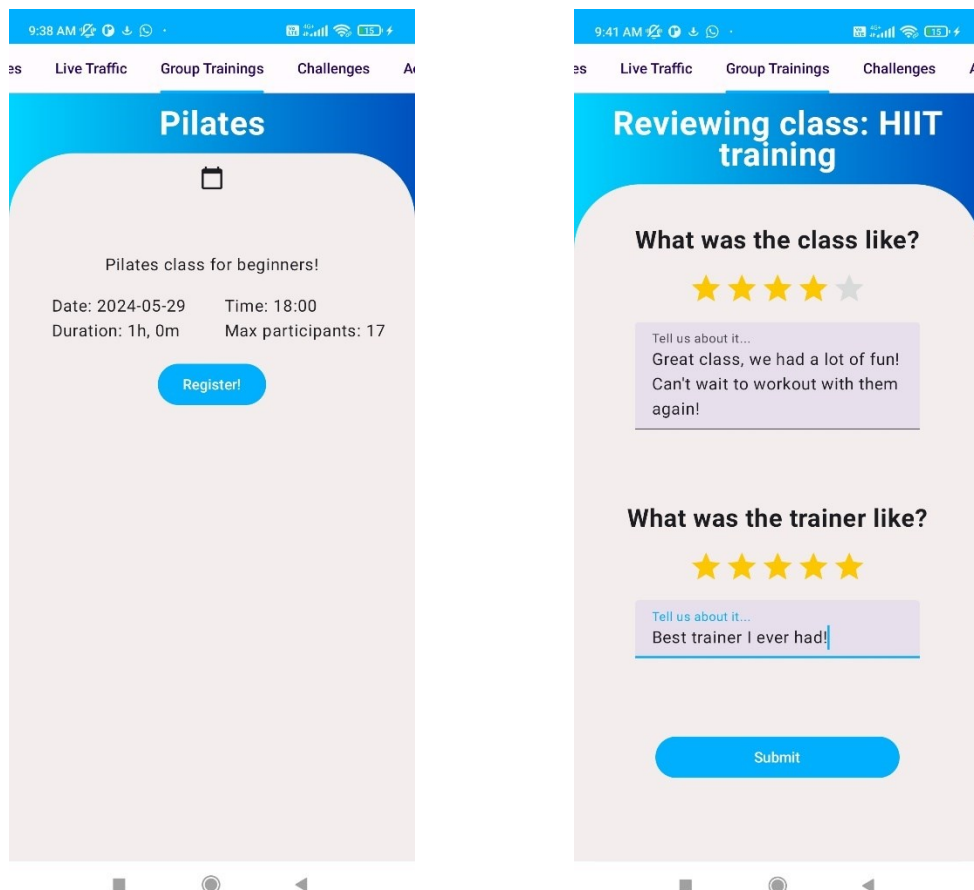
Slika 5.14: Prikaz ekrana izazova te detalja o pozirajućem izazovu

Nakon opisa sustava izazova, sljedeća važna funkcionalnost koju aplikacija nudi članovima je pregled grupnih treninga. Ova funkcionalnost omogućava članovima da lako prate i sudjeluju u različitim treninzima koje teretana nudi. Članovima je ponuđen poseban zaslona na kojem članovi mogu vidjeti sve grupne treninge koji se odvijaju u teretani. Na tom ekranu prikazani su treninzi zajedno s osnovnim informacijama kao što su naziv treninga, datum i vrijeme održavanja te trajanje treninga. Ako je trening ponavljajući, na ovom ekranu će biti prikazan sljedeći termin koji se održava, omogućujući članovima da planiraju svoje aktivnosti unaprijed. Također, ukoliko je član sudjelovao na nekom od treninga, nakon što je trening završio članu će biti dostupno recenziranje treninga. Za detaljniji pregled svih budućih termina treninga, članovi mogu koristiti ekran s kalendarom. Kalendar pruža pregled svih zakazanih termina, omogućujući članovima da vide cijeli raspored treninga na jednom mjestu. Na ovaj način, članovi mogu lako planirati svoje vrijeme i odabrati termine koji im najviše odgovaraju. Navedeni ekrani mogu se vidjeti na slici Slika 5.15.



Slika 5.15: Prikaz nadolazećih treninga u običnom te kalendarskom prikazu

Klikom na bilo koji termin treninga, bilo da se radi o terminu prikazanom na ekranu s popisom sljedećih treninga ili na kalendaru, otvara se prozor s detaljnijim informacijama o treningu. U ovom prozoru članovi mogu saznati više informacija o treningu. Također, iz ovog prozora članovi mogu jednostavno rezervirati svoje mjesto na odabranom terminu treninga. Osim toga, nakon što je trening završen, članovi imaju mogućnost ostaviti recenziju treninga i trenera. Na ekranu sa slike Slika 5.16 pojavljuju se i oni treninzi koji su spremni za recenziju, odnosno oni treninzi na kojima je trenutno prijavljeni član bio participant, a klikom na trening, otvara se ekran gdje član može ocijeniti trening i trenera te ostaviti povratne informacije. Ove recenzije su ključne za poboljšanje kvalitete usluga, jer vlasnici teretana i treneri mogu koristiti ove povratne informacije za kontinuirano unapređenje treninga i prilagođavanje programa potrebama članova.



Slika 5.16: Prikaz ekrana s informacijama o treningu te ekrana za recenziranje treninga

Navedene funkcionalnosti pregleda informacija o treningu te recenziranju gotov treninga može se vidjeti na slici. Spomenuta funkcionalnost je ključna za poboljšanje korisničkog iskustva, jer omogućuje članovima da lako pronađu i rezerviraju treninge prema svojim interesima i rasporedu, a nakon toga i ostave povratne informacije koje doprinose ukupnoj kvaliteti usluga u teretani. Transparentnost i jednostavnost pristupa informacijama o treninzima dodatno potiču angažman članova, omogućujući im da maksimalno iskoriste ponudu teretane.

Nakon što smo opisali pregled grupnih treninga i sustava recenziranja, sljedeća važna funkcionalnost koju aplikacija nudi članovima je ekran *Live Traffic*. Ovaj ekran pruža članovima bitne informacije o trenutnoj i prosječnoj posjećenosti teretane, što im pomaže u planiranju posjeta i izbjegavanju gužvi. Na tom ekranu članovi mogu vidjeti podatke o prosječnoj posjećenosti teretane za taj dan, prikazane po satima. Grafički prikaz prosječne posjećenosti po satima pruža jasan uvid u uzorke posjećenosti, što je korisno za optimizaciju vremena dolaska u teretanu. Još jedan važan podatak koji je prikazan na istome ekranu je trenutni broj ljudi u teretani. Ova informacija omogućuje članovima da u svakom trenutku

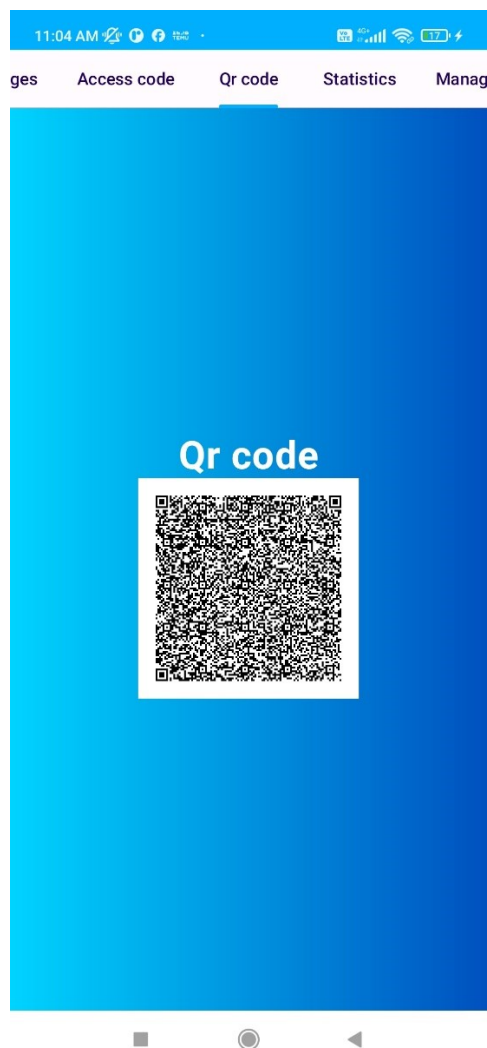
znaju koliko je ljudi trenutno prisutno u teretani. Pored toga, ekran prikazuje postotak koliko je trenutni broj ljudi veći ili manji od uobičajenog broja ljudi za taj dan i to vrijeme. Na taj način, članovi mogu brzo procijeniti hoće li u teretani biti gužva ili ne, i prilagoditi svoje planove u skladu s tim. Na slici Slika 5.17 prikazan je je izgled ekrana *Live Traffic*, uključujući graf prosječne posjećenosti po satima te informacija o trenutnom broju prisutnih članova u teretani.



Slika 5.17: Prikaz ekrana *Live Traffic*

Osim prethodno opisanih funkcionalnosti, aplikacija nudi i mogućnost korištenja ekrana sa QR kodom za jednostavno i efikasno bilježenje posjeta teretani što je ključno za praćenje aktivnosti članova i prikupljanje podataka koji se koriste u različitim statističkim analizama te u sustavu izazova. Na tom ekranu članovi mogu generirati svoj jedinstveni QR kod koji se skenira na ulazu u teretanu. Skeniranjem QR koda, njihov posjet biva zabilježen u sustavu. Ovaj proces je brz i jednostavan, te osigurava točnost podataka o posjetama. Podaci o posjetima koriste se za prikazivanje na ekranu *Live Traffic*, omogućujući ostalim članovima uvid u trenutnu popunjenost teretane. Također, skeniranjem QR koda pri dolasku, aplikacija provjerava je li član svojim dolaskom ispunio neki od aktivnih izazova, ukoliko takvih ima. Na primjer, ako je član posjetio teretanu određeni broj puta ili došao u specificirano vrijeme,

njegov posjet će biti zabilježen kao dovršenje izazova i bit će nagrađen odgovarajućim bodovima. Za svaki posjet, u sustavu se bilježi datum i vrijeme dolaska, kao i ukupno trajanje boravka u teretani. To znači da članovi moraju skenirati svoj QR kod i pri odlasku iz teretane, čime se osigurava precizno praćenje vremena provedenog u teretani. QR kod rješava i problem potrebe za nošenjem posebne kartice koju teretane obično izrađuju pri prvom ućlanjenju. U tradicionalnim sustavima, ako član dođe bez kartice, osoblje teretane mora ručno potražiti člana u sustavu kako bi provjerili ima li važeću članarinu. Ovaj proces je spor i neefikasan. Međutim, gotovo nitko ne dolazi u teretanu bez mobitela, što znači da članovi uvijek mogu koristiti QR kod generiran u aplikaciji za prijavu i odjavu pri ulasku i izlasku iz teretane. Na ovaj način, QR kod značajno pojednostavljuje proces ulaska i izlaska, štedi vrijeme osoblju i osigurava točnost podataka. Vizualni prikaz ekrana s QR kodom može se vidjeti na slici Slika 5.18



Slika 5.18: Prikaz ekrana s QR kodom

6. Zaključak

U ovom radu cilj je bio razviti mobilnu aplikaciju koja će olakšati upravljanje teretanama, optimizirati rad trenera i poboljšati iskustvo članova. Motivacija za ovaj projekt proizašla je iz potrebe za modernizacijom poslovanja teretana i odgovorom na rastuće zahtjeve korisnika. Ova aplikacija omogućava članovima pregled sadržaja teretane, prijavu na treninge, praćenje popunjenosti teretane u stvarnom vremenu i evidenciju dolazaka putem QR koda. Implementiran je sustav za recenziranje sadržaja, kao i funkcionalnost automatskog obnavljanja članstva. Za trenere, aplikacija omogućava upravljanje kalendarom, definiranje i uređivanje termina vođenih treninga te slanje *push* notifikacija članovima o promjenama u rasporedu što pomaže trenerima da učinkovito planiraju i organiziraju svoje aktivnosti te prate povratne informacije od članova. Vlasnicima teretane aplikacija nudi detaljan uvid u poslovanje kroz skup izvještaja koji uključuju podatke o članovima, iskorištenosti termina i učinkovitosti trenera, što omogućuje vlasnicima da donose informirane odluke i optimiziraju poslovne procese, čime se povećava ukupna učinkovitost i zadovoljstvo korisnika.

Jedan od najznačajnijih izazova razvoja ovog projekta bio je rad s tehnologijama s kojima nisam prethodno imao iskustva. Jedina tehnologija s kojom sam bio upoznat bila je Spring. Suočavanje s novim alatima, kao što su Stripe i Auth0, predstavljalo je veliku prepreku baš zbog vrlo kompleksnih sučelja servisa Stripe i Auth0. Stripe API, sa svojom širokom primjenjivošću, bio je posebno izazovan. Omogućiti plaćanje i stvaranje pretplata kroz Stripe zahtijevalo je detaljno proučavanje njihove dokumentacije. Također, bilo je važno razumjeti način razmjene podataka s Auth0 za pravilno funkcioniranje autentifikacije i autorizacije. Iako Auth0 pojednostavljuje mnoge aspekte sigurnosti, bilo je potrebno razumjeti kako njihovi servisi upravljaju korisničkim podacima i kako ih integrirati u sustav.

Budući rad mogao bi se fokusirati na dodatnu personalizaciju korisničkog iskustva, uvođenje novih elemenata igifikacije i proširenje funkcionalnosti aplikacije na druge aspekte upravljanja teretanom. Moguće je unaprijediti aplikaciju dodavanjem izazova koji nagrađuju korisnike određenim brojem bodova za kontinuirano članstvo u teretani od dva, šest ili dvanaest mjeseci, ili za redovito sudjelovanje u grupnim treninzima. To bi doprinijelo boljem korisničkom iskustvu, motivirajući članove na dugoročnu angažiranost i redovitost. Nadalje, nagrade bi mogle biti dostupne unutar aplikacije, omogućujući korisnicima pregled i preuzimanje nagrada čim dosegnu potreban broj bodova, što bi dodatno povećalo zadovoljstvo korisnika. Također, vlasnicima teretana mogli bi se omogućiti pristup dodatnim

statističkim podacima, kao što su prosječno vrijeme članova provedeno u teretani, prosječno trajanje članstva i prosječna potrošnja članova prije nego što prestanu biti aktivni. Ove informacije bi vlasnicima omogućile bolje razumijevanje ponašanja članova i optimizaciju poslovanja. Članovima bi se mogla pružiti mogućnost pregleda osobnih statistika, uključujući graf posjeta teretani kroz određeni vremenski period i prosječno vrijeme zadržavanja. Pružanje ovih podataka korisnicima omogućilo bi im praćenje vlastitog napretka i dodatno ih motiviralo na aktivnost, što bi znatno poboljšalo njihovo korisnički iskustvo. Aplikacija bi također mogla sadržavati *feed* za objave trenera i vlasnika, u kojem bi se mogli dijeliti novosti o teretani i popisi nadolazećih događaja što bi članove na vrijeme informiralo o svim aktivnostima i događanjima, poboljšavajući njihovu povezanost s teretanom i osjećaj zajednice. Sve navedene funkcionalnosti zajedno bi rezultirale značajnim poboljšanjem cjelokupnog korisničkog iskustva za članove, trenere te vlasnike.

7. Literatura

- [1] Jet Brains, *Kotlin Docs*, Dokumentacija, 2024. Poveznica: <https://kotlinlang.org/docs/home.html>; Pristupljeno 23. svibnja 2024.
- [2] Spring, *Spring Boot*, Dokumentacija, 2024. Poveznica: <https://spring.io/projects/spring-boot>; Pristupljeno 23. svibnja 2024.
- [3] Spring, *Spring Data*, Dokumentacija, 2024. Poveznica: <https://spring.io/projects/spring-data>; Pristupljeno 24. svibnja 2024.
- [4] Spring, *Spring Data Jpa*, Dokumentacija, 2024. Poveznica: <https://spring.io/projects/spring-data-jpa>; Pristupljeno 24. svibnja 2024.
- [5] Spring, *Spring Security*, Dokumentacija, 2024. Poveznica: <https://spring.io/projects/spring-security>; Pristupljeno 25. svibnja 2024.
- [6] Backlinko, *iPhone vs. Android Market Share: 2024 Statistics.*, Članak, 2024. Poveznica: <https://backlinko.com/iphone-vs-android-statistics>; Pristupljeno 25. svibnja 2024.
- [7] Google, *Guide to app architecture*, Dokumentacija, 2024. Poveznica: <https://developer.android.com/topic/architecture>; Pristupljeno 25. svibnja 2024.
- [8] Google, *Build better apps faster with Jetpack Compose*, Priručnik, 2024. Poveznica: <https://developer.android.com/develop/ui/compose>; Pristupljeno 26. svibnja 2024.
- [9] Google, *Firebase Cloud Messaging*, Dokumentacija 27. svibnja 2024. Poveznica: <https://firebase.google.com/docs/cloud-messaging>; Pristupljeno 27. svibnja 2024.
- [10] Stripe, *Api reference*, Dokumentacija 2024. Poveznica: <https://docs.stripe.com/api>; Pristupljeno 28. svibnja 2024.
- [11] Auth0, *Learn about Auth0*, Dokumentacija 2024. Poveznica: <https://auth0.com/docs>; Pristupljeno 29. svibnja 2024.

Mobilna aplikacija za teretane

Sažetak

Ovaj rad razvija sveobuhvatno programsko rješenje za unapređenje upravljanja teretanama integracijom tehnologija poput Kotlin-a za razvoj Android mobilnih aplikacija i Spring Frameworka za razvoj backenda. Funkcionalnosti aplikacije usmjerene su na tri glavna korisnika, a to su vlasnici, treneri te članovi teretana. Aplikacija članovima omogućava rezervaciju i pregled treninga, kao i pristup informacijama o popunjenosti prostora u stvarnom vremenu, dok je za trenere razvijena funkcionalnost upravljanja terminima treninga. Vlasnici koriste aplikaciju za pristup detaljnim izvještajima o korištenju resursa teretane. Također, uvedeni su elementi igrifikacije za povećanje angažmana korisnika, te sustav preporuka koji nagrađuje članove za dovođenje novih korisnika.

Mobile application for gyms

Abstract

This paper develops a comprehensive software solution for enhancing gym management by integrating technologies such as Kotlin for developing Android mobile applications and Spring Framework for backend development. The application's functionalities are targeted at three main users: gym owners, trainers, and members. It allows members to reserve and review training sessions, as well as access real-time information about facility occupancy, while trainers have functionalities for managing training schedules. Owners use the application to access detailed reports on the use of gym resources. Additionally, elements of gamification have been introduced to increase user engagement, along with a recommendation system that rewards members for bringing in new users.