



VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
KOMPIUTERIJOS KATEDRA

Baigiamasis bakalauro darbas

Sporto programos sudarymo ir treniruočių paieškos sistema

Atliko:

Albertas Kružintaitis

parašas

Vadovas:

dr. Joana Katina

Vilnius

2020

Turinys

Santrauka	4
Summary	5
Iyadas	6
1. Analizė	8
1.1. Naudotojų poreikiai	8
1.2. Panašios sistemos	8
1.2.1. Panašios sistemos Lietuvos rinkoje	9
1.2.2. Panašios sistemos užsienio rinkoje	11
2. Sistemos teorinė dalis	14
2.1. Planuojamas sistemos funkcionalumas	14
2.2. Aktoriai ir objektai	15
2.3. UML diagramos	17
2.3.1. Panaudos atvejų diagrama	17
2.3.2. Veiklos diagrama	18
2.3.3. Duomenų bazių diagrama	19
2.4. Sisteminė aplinka	20
2.5. Naudojamos technologijos	21
2.5.1. Express.js	21
2.5.2. React.js	21
2.5.3. MongoDB	23
3. Sistemos funkcionalumo įgyvendinimas	24
3.1. Sporto programos generavimas	24
3.2. Kūno masės indekso apskaičiavimas	32
3.3. Procentinis kūno riebalų kiekis	33
3.4. Profilio sudarymas	34
3.5. Grupinių treniruočių skelbimas	34
3.6. Trenerio rezervacija	34
3.7. Asmeninių ir grupinių treniruočių pirkimas	35
3.8. Rezultatų stebėjimas diagramose	35
3.9. Saugumas	36
4. Išvados ir rekomendacijos	37

Priedai	39
A. Treniruočių programos generavimo algoritmas	39
B. Treniruočių programos generavimo forma	49
C. Treniruočių programos atvaizdavimas	50

Santrauka

Šiame darbe yra aprašoma individualios treniruočių programos generavimo sistema, kartu su panašių darbų analize ir projektavimo bei kūrimo procesais. Darbo tikslas – sukurti žmonėms automatizuotą sistemą, kuri pagal vartotojo nurodytus duomenis sugeneruotų individualią, būtent tam vartotojui pritaikytą, sporto programą. Darbo metu buvo įgyvendinta internetinė svetainė su keturiomis skirtingomis aplinkomis nuo kurių priklauso sistemos funkcionalumas: neprisijungusio naudotojo, prisijungusio trenerio, prisijungusio sportininko ir prisijungusio administratoriaus. Treneriai prisijungę prie sistemos galės susidaryti savo profilį su darbo tvarkaraščiu ir taip pat skelbti grupines treniruotes. Prisijungę sportininkai galės susikurti profilį, generuoti treniruočių programą, pirkti treniruotes ir vertinti trenerius.

Summary

System of workout program generation and coach selection

This thesis describes the system for generating an individual training program followed by the analysis of similar programs and the development process. The aim of this paper is to build an automated system allowing the generation of a unique workout program for each individual, which is based on user data specifications. During the process a particular website with four separate environments has been set up: user (not registered), user (registered as coach), user (registered as athlete) and the system administrator. Users who log into the system as coach, will be able to create their profile with a work schedule and group workouts they would be interested in. Users who log into as athletes, will be able to create a profile and then be assigned a unique workout program with a possibility of purchasing chosen workouts and booking coaches for individual workouts.

Ivyadas

Šiais laikais daugėjant ofisų ir atsirandant darbuotojų, kurie didžiąją laiko dalį praleidžia sėdėdami priešais kompiuterio ekraną, visuomenėje atsiranda labai didelė sveikatos sutrikimų problema. Žmonės vis dažniau susiduria su stuburo problemomis, antsvoriu ir kitomis ligomis, kurias įtakoja sėdimas darbas ir per mažas fizinis krūvis. Dabartinė sporto trenerių paslaugų teikimo sritis vis dar nėra labai gerai išvystyta ir pritaikyta šiuolaikiniam gyvenimo ritmui, todėl nemažas dalis žmonių neturi laiko ar galimybių pasinaudoti šių specialistų pagalba, ieškodami jų sporto klubuose ar interneto forumuose, kur dažnai pateikiama klaidinanti informacija ar treneriais skelbiasi jokio išsilavinimo neturintys savamoksliai ar apsimetėliai. Remiantis *"World Health Organization"* straipsniu [1], pasaulyje viršsvorio problemą turinčių žmonių skaičius patrigubėjo nuo 1975m., taip pat 2016m. buvo 1.9 mlrd. žmonių, kuriems buvo daugiau nei 18 metų ir kurie turėjo viršsvorio, t.y. jų kūno masės indeksas (KMI) buvo virš 25. Iš šių žmonių, 650 mln. turėjo nutukimo problemą, t.y. jų kūno masės indeksas (KMI) buvo virš 30. Kita rimta priežastis kurti sistemą susijusią su sportu yra per mažas fizinis aktyvumas pandemijos ar karantino atveju. Šiais laikais atsirandant vis naujoms ligoms ar virusams, žmonės būna priversti neišeiti iš namų, daugėja bedarbystės procentas. Kaip rašoma moksliniame straipsnyje [2], visa tai lemia žmonių patiriamą stresą ir atsiranda daugiau problemų susijusių su psichine sveikata. Tačiau psichinė sveikata yra susijusi su fiziniu aktyvumu ir norint sumažinti keliamą stresą ar gydyti tam tikras ligas pasireiškusias pandemijos ar karantino metu, reikia reguliariai atlikti fizinius pratimus. Kaip teigiama straipsnyje [3], fizinis aktyvumas gerina imuninę sistemą ir padeda apsaugoti organizmą nuo įvairių ligų. Visi šie duomenys tik dar labiau įrodo, kad problema yra didelė ir šioje srityje trūksta inovatyvių sprendimų, kurie pritrauktų žmones sportuoti ir lengvai rasti trenerį su aiškiu jo profiliu.

Šio darbo tikslas - atsižvelgiant į dažniausias šiuolaikinių žmonių problemas ir pasinaudojant šių dienų technologijomis sukurti internetinę sistemą, kuri teiktų treniruočių generavimo ir asmeninių bei grupinių treniruočių paslaugas, taip padėdama žmonėms gerinti fizinę formą ir sveikatą tiek sportuojant iš namų, tiek sporto klubuose. Baigiamajam bakalauro darbui buvo iškeltos šios užduotys:

- Atlikti panašių sistemų analizę
- Apibrėžti sistemos funkcionalumą
- Suprojektuoti sistemą
- Sukurti pilnai veikiančią sporto sistemą

Naujai kuriama sistema turėtų būti saugi ir prieinama iš betkokių įrenginių palaikančio interneto naršyklę, todėl sistemos kūrimui pasirinkta Javascript programavimo kalba dėl jos paprastumo kuriant internetines svetaines tiek iš serverio pusės (angl. *backend*), tiek iš išorinės pusės (angl. *frontend*). Kadangi naudojama Javascript programavimo kalba, buvo nuspręsta naudoti ir Javascript karkasą, nes būtent karkaso pagalba sistemos kūrimo procesas tampa paprastesnis ir greitesnis bei neapkrauna programinio kodo. Atlikus išsamią karkasų analizę, buvo nuspręsta naudoti Express.js karkasą serverio pusėje ir React.js karkasą išorinėje pusėje. Analizė yra pateikiama pagrindinėje darbo dalyje.

Kadangi sistema leis naudotis įvairiomis funkcijomis tiek treneriams, tiek sportuojantiems žmonėms, bus reikalingos ir kelios aplinkos, skirtos skirtingų tipų naudotojams. Pirmoji aplinka bus visiems prieinama, kuri atspindės sistemos veiklą ir bus reprezentatyvaus tipo su galimybe užsiregistruoti. Antroji aplinka skirta treneriams - joje kuriamos grupinės treniruotės ir pildomas trenerio darbo tvarkaraštis bei profilis. Trečioji aplinka skirta sportuojantiems žmonėms besinaudojantiems trenerio paslaugomis. Šioje aplinkoje naudotojai galės pirkti tiek grupines, tiek individualias treniruotes, susigeneruoti unikalią treniruočių programą ir vertinti trenerius. Ketvirtoji aplinka yra skirta sistemos administratoriui, kuris gali trinti ir keisti sistemos naudotojų informaciją. Kiekviena aplinka yra išsamiai aprašyta pagrindinėje darbo dalyje.

Pasiekti rezultatai: sukurta funkcionuojanti internetinė svetainė, kurioje naudotojai gali pasirinkti vieną iš dviejų sistemos naudotojų tipų: sportininko arba trenerio. Sportininkai gali susigeneruoti unikalią treniruočių programą, pirkti grupines treniruotes, rezervuoti trenerį individualiai treniruotei, matyti trenerio profilį su jo asmeniniais pasiekimais ir socialiniais tinklais, tokiais kaip instagram ir facebook, per kuriuos naudotojas gali susisiekti ir komunikuoti su treneriu. Taip pat profilyje galima rasti trenerio youtube kanalą, kuriame naudotojas gali matyti trenerio keliamus treniruočių klipus ir sportuoti pagal juos. Kitas sistemos privalumas - naudotojas trenerio profilyje gali matyti trenerio skype vardą, jog esant poreikiui, naudotojas turėtų galimybę susisiekti su treneriu ir atlikti treniruotę pasinaudojant tiesiogine transliacija. Treneris sistemoje gali susidaryti savo darbo tvarkaraštį, pagal kurį vyksta rezervacijos. Kitos funkcijos treneriui - galimybė skelbti grupines treniruotes ir stebėti užsiregistravusius į treniruotę žmones lentelėje.

1. Analizė

Skyriuje "Analizė" bandoma išanalizuoti vartotojų poreikius ir išdėstyti funkcinius reikalavimus, paremtus jais. Taip pat tiriama rinka, analizuojamos panašios sistemos.

1.1. Naudotojų poreikiai

Gyvename tokiam amžiuje, kai fizinį darbą keičia sėdimas. Daugėjant ofisų skaičiui, daugėja ir žmonių, kurie didžiąją laiko dalį praleidžia dirbdami kompiuteriu. Taip leidžiant laiką, suprastėja ne tik fizinė, bet ir psichologinė sveikata. Remiantis moksliniu straipsniu [4] yra įrodyta, jog fizinė veikla yra būtina norint palaikyti gerą smegenų veiklą ir išlaikyti sveiką psichinę būklę, kuri yra ypač svarbi dirbant ofise ar namuose naudojantis kompiuteriu. Taigi ši sporto sistema ir buvo sukurta atsižvelgiant į šiuolaikinių žmonių problemas. Ji leis naudotojams paprastai ir greitai susidaryti individualią sporto programą pagal norimus tikslus, tačiau jeigu naudotojas yra pradedantysis ir jam būtina trenerio pagalba, jis galės susirasti asmeninį trenerį ir sportuoti kartu su juo arba dalyvauti grupinėje treniruotėje kartu su kitais asmenimis.

1.2. Panašios sistemos

Atlikus išsamią analizę internete pavyko rasti keletą panašių sistemų, iš kurių buvo atrinktos tik panašiausią funkcionalumą turinčios sistemos detalesnei analizei. Panašios sistemos buvo atrenkamos atsižvelgiant į šiuos kriterijus:

- Galimybė rasti ir rezervuoti asmeninį trenerį
- Galimybė rasti ir pirkti grupinę treniruotę
- Galimybė vertinti trenerį
- Galimybė stebėti trenerio profilį
- Individualios treniruočių programos sudarymas
- Rezultatų sekimas diagramose

Darbo autorius domėdamasis naudotojų poreikiais atrado keletą panašių sistemų, kurios galėtų padėti žvelgiant į sistemos plėtojimą ateityje. Visų pirma pradėta ieškoti sistemų, kurios turėtų visus šešis anksčiau minėtus kriterijus. Kiti, ne tokie svarbūs kriterijai, pagal kuriuos buvo atliekama paieška t.y.: profilio sudarymas, socialinių tinklų dalinimasis, atsiliepimų apie trenerį rašymas, treniruočių filtravimas. Panašių sistemų buvo ieškota tiek Lietuvos, tiek tarptautinėje užsienio

rinkoje, todėl atskirai detaliau yra aprašoma, kokias sistemas pavyko rasti, kokią specifinį funkcionalumą jos turėjo ir kuo buvo panašios į kuriamą sistemą.

1.2.1. Panašios sistemos Lietuvos rinkoje

Pradedant ieškoti esamų panašių sistemų Lietuvos rinkoje buvo susidurta su keletu problemų. Visų pirma daugelis sistemų buvo sukurtos tam tikrų sporto klubo tinklų, tokių kaip Impuls, GymPlius ar LemonGym. Šios įmonės savo internetinėse svetainėse turėjo klube dirbančių trenerių anketas, kur naudotojas galėjo išsirinkti sau labiausiai patinkantį ir sportuoti jo atstovaujame klube. Tačiau tokių sistemų, kur individualiai dirbantys treneriai, nepriklausomai nuo sporto klubų, galėtų skelbti savo veiklą ir sportuoti įvairiuose sporto klubuose buvo nedaug. Tuo labiau rasti sistemą, kurioje yra galimybė susikurti sau individualią programą ir susirasti trenerį vienoje vietoje buvo didelis iššūkis.

www.kilo.lt

Pirmoji rasta sistema, kuri labiausiai atitinka kuriamos sistemos funkcionalumą yra internetinė svetainė www.kilo.lt. Ši sistema yra viena iš populiariausių ir labiausiai paplitusių sporto sistemų visoje Lietuvoje. Vieni iš pagrindinių panašumų su kuriama sistema buvo tai, jog www.kilo.lt svetainėje yra galimybė susikurti individualią sporto programą su detaliais aprašymais ir paveikslėliais, tačiau šis funkcionalumas internetinėje svetainėje turi vieną trūkumą, treniruočių programos yra mokamos ir jas sudaro treneriai, taigi jeigu naudotojas nori gauti treniruočių programą tą pačią dieną, tai gali sudaryti sunkumų. Kitas panašumas yra naudotojų profiliai su asmeniniais prisijungimais. Sistemos naudotojai gali prisijungti prie sistemos ir stebėti savo rezultatus, matyti sudarytą programą ir nebijoti jog sudarytos treniruotės failas išsitrins iš kompiuterio ar dings. Viskas yra saugojama internete ir matoma prisijungus prie paskyros. Tačiau www.kilo.lt sistemos naudotojas savo progresą mato tik iš nuotraukų kurios yra įkeliamos.

www.lympo.lt

Kita panaši internetinė svetainė yra www.lympo.lt. Šioje sistemoje galima sukurti savo asmeninį norinčio sportuoti arba norinčio treniruoti vartotojo profilį. Analizuodamas sistemą, darbo autorius užsiregistravo kaip norintysis sportuoti ir rasti trenerį naudotojas, taigi tai suteikė galimybę nurodyti ūgį, svorį, pasirinkti atstovaujamą sporto šaką. Atlikus visus anksčiau minėtus veiksmus buvo galima pradėti ieškoti asmeninio trenerio, kuris turi nurodytą kainą asmeninei ir grupinei treniruotei. Prie trenerio profilio yra atvaizduojamas jo įvertinimas gautas iš kitų naudotojų, treniruojama sporto šaka ir vieta kurioje jis dirba. Dažniausiu atveju treneriai nurodo, jog gali atvykti į nurodytą vietą, kas yra labai didelis privalumas. Tačiau be visų šių privalumų, ši sistema turi ir vieną trūkumą, naudojantis ja, negalima susidaryti individualios sporto programos.

www.getfit.lt

Dar viena panaši internetinė svetainė yra www.getfit.lt. Atlikus šios sistemos analizę pasirodė, jog šios sistemos veikla yra vykdoma vadinamoje "Delfi Sporto Centras" arenoje, kurioje jie skelbia įvairias funkcinės treniruotes su skirtingais treneriais ir jų priežiūra. Nors narystė yra mokama, tačiau naudotojas gali rinktis iš įvairiausių treniruočių. Tai ir yra vienas iš kuriamos sistemos panašumų, jog naudotojai galės rinktis grupines treniruotes palaikančioje ir motyvuojančioje bendruomenėje. Šios sistemos trūkumas yra tas, jog visos funkcinės treniruotės yra grupinės, sistemoje negalima rasti asmeninio trenerio, kuris dirbtų individualiai su sportuojančiu žmogumi.

Tarp kitų analizuotų ir peržiūrėtų internetinių svetainių funkcijų nebuvo panašių į kuriamos sistemos, nes dauguma internetinių svetainių susijusių su sportu yra įvairūs dienoraščiai (angl. blog), forumai ir informaciniai portalai, taip pat šiuo metu labai populiari visa turima naudinga informacija dalintis įvairiuose socialiniuose puslapiuose ar mobiliuose aplikacijose.

1.2.2. Panašios sistemos užsienio rinkoje

Kitas keliamas tikslas buvo surasti panašių sistemų užsienio rinkoje, kaip ir buvo tikėtasi, jų pavyko rasti kur kas daugiau nei Lietuvos rinkoje, tačiau dauguma jų buvo su vienodu funkcionalumu ir netgi panašiu dizainu. Taigi didesnis kiekis aplikacijų ir internetinių svetainių leido pasirinkti sistemas pagal aukštesnius kriterijus. Tai užtruko ganėtinai ilgą laiko tarpą, prireikė daugiau raktažodžių paieškose ir dauguma sistemų buvo panašios į lietuviškąsias sporto sistemas. Norint surasti šiek tiek įdomesnę ir išsiskiriantį funkcionalumą turinčias sistemas, kurios išsiskirtų savo unikalumu iš visų konkurencingų lietuviškų ir užsienietišκών sistemų prireikė daugiau laiko.

www.mirror.co

Šioje internetinėje svetainėje žmonės gali įsigyti realų produktą, kuris yra kartu ir veidrodis, ir treniruočių sistema, kurią vartotojai vadina - interaktyvia namų sporto sale. Šis produktas yra vienas iš inovatiškiausių analizuojamų sistemų/produktų, kadangi kol jis yra išjungtas, jis atrodo lyg veidrodis, tačiau jis turi įmontuotą LCD ekraną, kurį įjungus galima pasirinkti įvairių žanrų treniruotes. Nors tai gali atrodyti lyg paprastas sporto treniruočių vaizdo klipų rinkinys, tačiau iš tiesų taip nėra, kadangi veidrodis turi įmontuotą kamerą, jis stebi atliekamus naudotojo veiksmus ir įspėja, jeigu pratimas atliekamas su prasta technika. Taip pat veidrodis gali sinchronizuotis su mobiliąja aplikacija ir išmaniuoju laikrodžiu, taip sistema stebi širdies ritmą ir parenka atitinkamo lygio pratimus.

www.bodbot.com

Šioje svetainėje vartotojai gali susigeneruoti sau individualią sporto programą užpildant anketoje tam tikrus asmeninius duomenis, kurie reikalingi kiekvienai sporto programai - amžius, svoris, lytis. Be asmeninių rodiklių pradedantysis sportuoti turi nurodyti treniruotės tikslą, treniruotės trukmę, intensyvumą ir raumenų grupes į kurias turi būti atsižvelgta. Taip pat privaloma nurodyti kokią sporto įrangą turi naudotojas ir kuri bus naudojama sportuojant. Ši sistema yra ganėtinai pažangi, kadangi atsižvelgia į daug svarbių kriterijų, tokių kaip turimas sporto inventorius ir atsiliekančios raumenų grupės, tai leidžia sudaryti tikslią ir naudingą sporto programą. Dauguma kriterijų, prašomų įvesti siekiant sudaryti sporto programą sutampa ir su kuriamoje sistemoje, tik yra vienas skirtumas, jog nėra galimy-

bės pasirinkti turimą sporto inventorių, pažymėti atsiliekančias raumenų grupes, kurioms reikėtų skirti didesnę dėmesį treniruojantis ar pasirinkti treniruotės trukmę. Analizuojamoje sistemoje taip pat yra daugybė kitų naudingų funkcijų, tačiau dauguma jų yra gaunamos tik tuo atveju, kai naudotojas nusiperka mokamą narystę.

Nike Training Club

Tai viena iš populiariausių aplikacijų užsienyje susijusių su sportu. Analizuojant sporto sistemas, internete dauguma jų buvo aplikacijos mobiliems telefonams, tačiau ši aplikacija iš visų kitų išsiskyrė tuo, jog turėjo didžiausią įvertinimą. Ji yra viena iš daugiausiai naudojamų ir turi keletą įdomių ir unikalių funkcijų. Aplikacija yra mokama ir ją išsigiję naudotojai gauna prieigą prie įvairiausių pratimų, pradedant fitnessu ir baigiant ties joga. Aplikacijoje yra nemažai trenerių, kurie teikia su sportu susijusias paslaugas, taigi tai yra vienas iš pagrindinių panašumų susijusių su mano kuriamą sistema, kadangi vartotojai gali gauti ne tik individualias treniruotes, bet ir trenerių pagalbą išskylant įvairiems nesklandumams ar susiduriant su žinių trūkumu sporto pratimų srityje.

www.workoutscheduler.net

Dar viena panaši sistema yra taip pat internetinė svetainė pavadinimu *Workout-Scheduler*. Ši sistema turi taip pat ir mobiliąją aplikaciją, kuri yra pritaikyta tiek Android, tiek iOS operacinėms sistemoms. Ši sistema taip pat leidžia vartotojams gauti sporto treniruočių programas suvedant asmeninius fizinius duomenis, tačiau sistema pati negeneruoja programų, o tik parenka labiausiai tinkamą iš duomenų bazės, kurioje jau yra keletas sukurtų skirtingų sporto programų. Tai reiškia, kad naudotojas negaus individualiai pritaikytos sporto sistemos, ji tik dalinai atitiks reikalavimus. Prie šio funkcionalumo, sistema taip pat turi prisijungimo ir registravimosi funkcionalumą.

8fit

Paskutinė sistema, kurią aprašysiu yra taip pat labai populiari užsienyje ir vadinasi *8fit*. Tai yra mobilioji aplikacija ir yra pritaikyta dvejoms operacinėms sistemoms - Android ir iOS. Ši sistema yra kaip ir daugelis kitų mobiliųjų aplikacijų, kadangi yra skirta sportui namuose, tačiau turi vieną išskirtinumą, kad visas jos funkcionalumas yra automatizuotas ir nereikalaujantis trenerių pagalbos. Norint susigeneruoti individualią sporto programą reikia nurodyti savo asmeninius fizinius duomenis: amžių, svorį, ūgį, kūno riebalų masę, fizinį aktyvumą ir sporto patirtį. Sportas namuose reiškia, jog vartotojui neprireiks jokios sportinės įrangos ir visi pratimai bus atliekami namų sąlygomis ir panaudojant naudotojo svorį, tai yra vienas iš pagrindinių aplikacijos privalumų lyginant su kuriama sistema, kurioje pratimus rekomenduojama atlikti sporto salėje ar pasinaudojant trenerio pagalba, jeigu naudotojas yra pradedantysis ir prieš tai nesportavęs. Kitos funkcijos, kurias turi 8fit aplikacija yra įvairių patiekalų pateikimas, nurodant juose esamas kalorijas ir kitas maistingas medžiagas, tikslo užsibrėžimas nurodant norimą pasiekti kūno svorį, tačiau šios funkcijos yra gaunamos tik su mokama 8fit prenumerata.

Apibendrinimui, buvo sukurta lentelė 1, kurioje yra pateiktas tam tikrų specifinių funkcijų atitikimai ir skirtumai tarp išanalizuotų sistemų ir sporto programos generavimo ir trenerių paieškos internetinės svetainės. Kaip papildomas funkcijas, pagal kurias atvaizdavau skirtumus ir panašumus buvo: galimybė rasti tinkantį trenerį pagal miestą, sporto šaką ir kitus kriterijus, nusipirkti grupinę ar asmeninę treniruotę tam tikrame mieste ir tam tikru laiku, susikurti individualią sporto programą, sekti rezultatus diagramose, rašyti atsiliepimus ir vertinti trenerį, susikurti trenerio arba sportininko profilį.

Funkcijos Sistema	Sporto programos sudarymas	Vartotojo progreso stebėjimas	Trenerių paieška	Treniruočių paieška	Vartotojų registracija	Papildomos funkcijos
www.kilo.lt	Ne	Ne	Taip	Ne	Taip	Dalinai
www.lympo.lt	Ne	Ne	Taip	Ne	Taip	Dalinai
www.getfit.lt	Ne	Ne	Taip	Taip	Ne	Ne
www.mirror.co	Ne	Ne	Taip	Taip	Taip	Dalinai
www.bodbot.com	Taip	Taip	Ne	Ne	Taip	Ne
Nike Training Club	Ne	Dalinai	Taip	Dalinai	Taip	Dalinai
Workout Scheduler	Dalinai	Dalinai	Ne	Ne	Taip	Ne
8fit	Taip	Dalinai	Ne	Dalinai	Taip	Dalinai

1 lentelė. Išanalizuotų panašių esamų sistemų analizė

2. Sistemos teorinė dalis

Siekiant sukurti sistemą, kuri būtų plačiai naudojama tarp daugybės skirtingų naudotojų bei lengvai plėtojama, kūrimo pradžioje labai svarbu priimti teisingus sprendimus susijusius su dizainu ir pasirinkti labiausiai tinkamas technologijas atsižvelgiant į numatytą funkcionalumą. Taigi šiame skyriuje yra pateikta visa kuriamos sistemos planavimo bei modeliavimo teorinė dalis. Toliau poskyriuose yra detalčiai aprašytas prototipų kūrimas ir sistemos funkcionalumas. Taip pat yra atvaizduojami aktoriai ir objektai, jų funkcijos ir būsenos, pateiktos pagrindinės UML diagramos bei aprašytas sistemos duomenų bazės modelis.

2.1. Planuojamas sistemos funkcionalumas

Prieš pradėdant visus darbus ir išanalizavus daugumą panašių sistemų, buvo sudarytas pradinis projekto planas, kuriame detalčiai aprašiau visas planuojamas įgyvendinti sistemos funkcijas:

- Naudotojų registracija
- Naudotojų prisijungimas
- Naudotojų profilio sudarymas ir redagavimas
- Naudotojų atpažinimas pagal naudotojo tipą (sportininkas, treneris, administratorius)
- Administratoriaus funkcijos (naudotojų duomenų redagavimas ir trynimas)
- Kūno masės indekso skaičiavimas
- Procentinis kūno riebalų kiekio skaičiavimas
- Treniruočių pratimų peržiūra
- Sporto programos sudarymas pagal naudotojo pateiktus duomenis
- Grupinių treniruočių paieška ir pirkimas
- Trenerių paieška individualioms treniruotėms ir rezervavimas
- Naudotojų nusipirkusių individualią ar grupinę treniruotę atvaizdavimas

2.2. Aktoriai ir objektai

Šioje sistemoje yra planuojami keturi skirtingi naudotojų tipai (aktoriai): anoniminiai (neprisijungę) naudotojai, prisijungę kaip treneriai, prisijungę kaip sportininkai ir sistemos administratoriai. Nuo šių tipų priklausys ir tai, kokias specifines sistemos funkcijas naudotojai galės vykdyti:

- Anoniminiai naudotojai gali naudotis šiomis sistemos funkcijomis:
 - Užsiregistruoti sistemoje
 - Patvirtinti savo registraciją paspaudus ant gautos nuorodos elektroniniame pašte
 - Registruojantis pasirinkti savo norimą sistemos tikslą (Treneris ar Atletas)
- Prisijungę kaip sportuojantys ir norintys pasiekti tam tikrų fizinių rezultatų naudotojai gali atlikti anonominių naudotojų funkcijas ir papildomai turi galimybes:
 - Prisijungti prie sistemos
 - Redaguoti savo profilio duomenis
 - Susigeneruoti individualią sporto programą pagal tam tikrus kriterijus
 - Stebėti KMI pasikeitimus diagramoje
 - Stebėti procentinio kūno riebalų kiekio pasikeitimus diagramoje

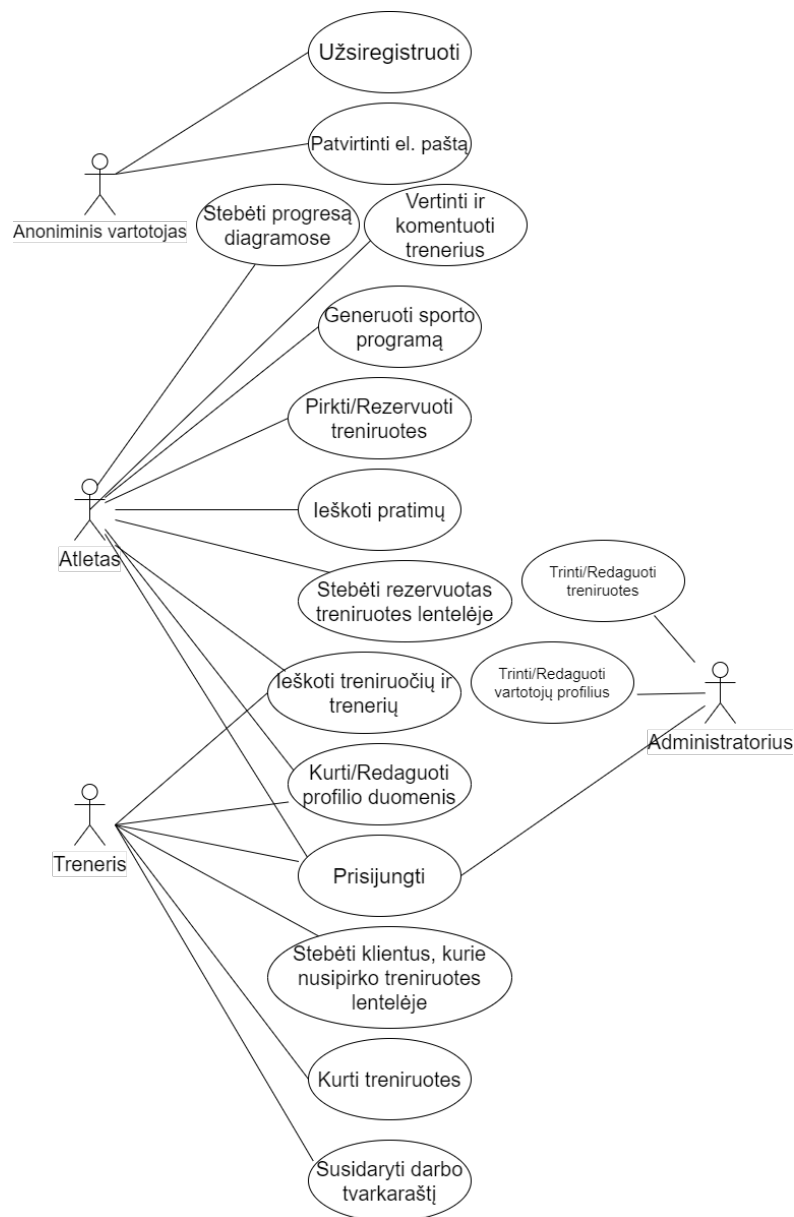
- Stebėti klubų, juosmens ir kaklo apimčių pasikeitimus diagramoje
 - Ieškoti trenerio pagal atitinkamus kriterijus
 - Rezervuoti/Nusipirkti trenerį, pagal pasirinktą laiką, individualioms treniruotėms
 - Ieškoti grupinės treniruotės pagal atitinkamus kriterijus
 - Rezervuoti/Nusipirkti grupinę treniruotę
 - Peržiūrėti trenerio profilį
 - Rašyti atsiliepimą apie trenerį ir įvertinti penkiabalėje sistemoje
 - Peržiūrėti treniruotės pilną aprašymą
 - Matyti nusipirkta/rezervuota treniruotės lentelėje
 - Matyti ir filtruoti visus pratimus pagal atitinkamus kriterijus
- Prisijungę su trenerio anketa ir norintys rasti klientų naudotojai gali atlikti anoniminių naudotojų funkcijas ir papildomai turi galimybes:
 - Prisijungti prie sistemos
 - Redaguoti savo profilio duomenis
 - Pasidalinti savo socialiniais tinklais profilyje
 - Pridėti, atnaujinti arba ištrinti savo išsilavinimą profilyje
 - Pridėti, atnaujinti arba ištrinti savo profesinius pasiekimus profilyje
 - Skelbti grupines treniruotes su galimybe sportininkams jas pirkti
 - Susidaryti savo darbo tvarkaraštį pagal atitinkamas dienas ir laiką
 - Matyti jau praėjusias treniruotes archyvo skytyje
 - Trinti jau pasibaigusias treniruotes
 - Matyti atletus su jų nupirktomis treniruotėmis lentelėje
 - Matyti kitų trenerių profilius
 - Taip pat sistemoje bus administratoriaus paskyra kurioje jis galės:
 - Matyti visus užsiregistravusius naudotojus
 - Pakeisti naudotojų rolę iš sportininko į trenerį ir atvirkščiai
 - Pakeisti naudotojų vardą, elektroninį paštą
 - Ištrinti naudotoją
 - Ištrinti treniruotę
 - Redaguoti treniruotę

2.3. UML diagramos

Sukūrus išsamų funkcionalumo planą ir norint kuo aiškiau suprasti sistemos ar tam tikro proceso veikimą, buvo sukurtos pagrindinės ir plačiausiai naudojamos UML diagramos.

2.3.1. Panaudos atvejų diagrama

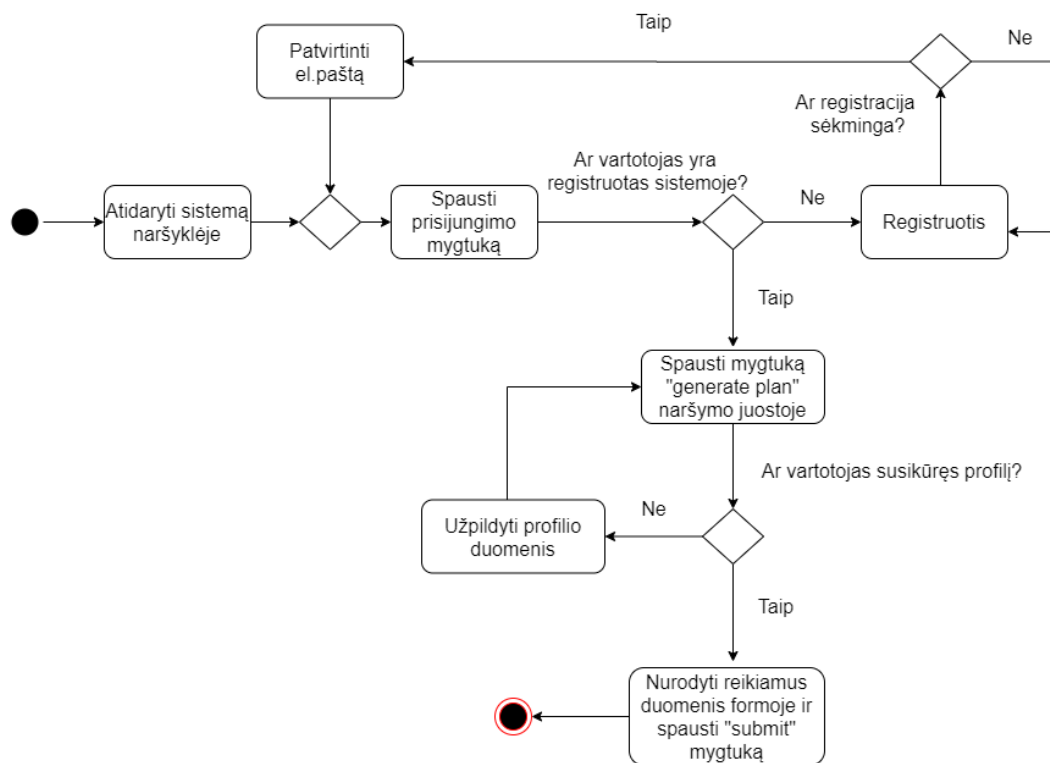
Norint kuo aiškiau suprasti sistemos funkcionalumą tam tikram naudotojui, buvo sukurta panaudos atvejų diagrama 2 pav. Diagramoje yra atvaizduojami praeitame skyriuje aptarti aktoriai su jiems priskirtomis funkcijomis.



2 pav. Panaudos atvejų diagrama

2.3.2. Veiklos diagrama

Kadangi buvo nuspręsta sukurti daug funkcijų turinčią sistemą, veiklos diagramai aprašyti buvo pasirinkta tik vieną iš daugelio funkcijų - unikalios sporto programos generavimą. Ši diagrama atvaizduoja sistemoje vykstančių veiksmų seką norint gauti unikalią treniruočių programą. Tai leis išsigilinti ir suprasti sistemos veikimo principą.



3 pav. Veiklos diagrama

Kaip yra pavaizduota 3 pav., naudotojas visų pirma turi atsidaryti sistemą naršyklėje ir paspausti prisijungimo mygtuką. Tačiau, jeigu naudotojas nėra anksčiau naudojęsis sistema ir nėra atlikęs registracijos, jis visų pirma turės užpildyti registracijos formą. Užpildžius formą ir atlikus teisingą registraciją, naudotojas gaus elektroninį paštą patvirtinantį laišką, kuriame gaus adresą, kurį privalės atidaryti. Atidarius laišką gautą adresą, naudotojas bus nukreipiamas į pradinį sistemos langą, kuriame galės spausti prisijungimo mygtuką ir prisijungti. Po šių veiksmų, naudotojas bus nukreipiamas į profilio sudarymo anketą, kuriame naudotojas turės nurodyti savo asmeninius duomenis. Tačiau jeigu naudotojas jau bus atlikęs visus anksčiau minėtus veiksmus, jis galės spausti "Workout Plan" mygtuką naršymo juostoje, užpildyti reikiamus duomenis, pagal kuriuos bus generuojama programa ir spausti *Submit* mygtuką unikalios treniruočių generavimui.

2.3.3. Duomenų bazių diagrama

Duomenų bazių diagramoje yra pavaizduoti visi planuojami komponentai, kurie bus naudojami sistemoje, pagrindiniai iš jų bus *User* (naudotojas), *Coach* (treneris) ir *Athlete* (atletas). Diagramoje taip pat pavaizduota kuris komponentas, kurį komponentą paveldės. Dažniausiai paveldimas komponentas bus *User* (naudotojas), kadangi kiekvienas prisiregistravęs žmogus taps naudotoju ir vėliau susikūręs profilį, paveldės naudotojo duomenis t.y vardą, elektroninį paštą, rolę ir paveikslėlį, kuris bus gaunamas iš elektroninio pašto. Naudotojas susikūręs trenerio profilį savyje turės sistemos naudotojo duomenis, taip pat galės turėti kelis *Achievements* (pasiekimus), *Education* (išsilavinimą), kur galės nurodyti visus savo turimus diplomus ar sertifikatus ir *Social* (socialinių tinklų adresus). Tačiau vienas iš svarbiausių trenerio komponentų bus *WorkSchedule* (darbo tvarkaraštis), kur treneris galės saugoti savo darbo valandas ir sukurtas grupines treniruotes. Naudotojas pasirinkęs atleto vaidmenį paveldės naudotojo (*User*) duomenis ir *Workout* (treniruočių) eilutėje turės nupirktas treniruotes, kurios bus grupinės arba individualios, taip pat sugeneruotas individualias treniruotes su pratimų sąrašais saugos *WorkoutPlan* eilutėje.

įrankius, bei motyvaciją juos pritaikyti. Sporto generavimo sistema bus pakankamai sudėtinga internetinė sistema, todėl be standartinių programavimo kalbų, kaip HTML, CSS bei JavaScript, bus pritaikomi papildomi įrankiai norint užtikrinti visapusiškai greitesnį darbą.

2.5. Naudojamos technologijos

Kuriant sistemą, bus pasirinkta JavaScript programavimo kalba, kadangi ji turi nepaprastai platų funkcionalumą turinčias bibliotekas ir lengvai suprantamą sintaksę. Taip pat JavaScript yra labai lanksti programavimo kalba, leidžianti kurti visiškai skirtingo pobūdžio sistemas bei programas. Taigi visų pirma buvo atsižvelgta į būtent šios programavimo kalbos technologijas. Vėliau buvo ieškota karkasų kurie yra plačiausiai naudojami ir yra vieni iš populiariausių tarp įvairių programuotojų ar didelių kompanijų, kas leistų šią programą plėtoti ateityje ir integruoti į kitas sistemas. Taip pat išanalizuota kokia jų nauda kuriant atitinkamą funkcionalumą tiek sistemos vidinėje pusėje (angl. *back-end*), tiek išorinėje naudotojo sąsajoje (angl. *front-end*).

2.5.1. Express.js

Kadangi sistemos pagrindinis funkcionalumas bus vidinėje sistemos pusėje, visų pirma paieška vyko būtent šios dalies karkasui. Atlikus analizę puslapiuose, tokiuose kaip *Github*, *The State of JavaScript* ir *NPM trend*, buvo aiškūs vieni pagrindinių ir daugiausiai naudojamų javascript karkasų serverio pusėje, tai yra: Express, Next.js, Nuxt. Tačiau galutinis pasirinkimas buvo priimtas - pasirinkti Express karkasą. Kadangi šis karkasas yra išrinktas labiausiai lūkesčius atitinkantis ir tenkinantis reikalavimus karkasas tris metus iš eilės *The State of JavaScript* svetainėje [5]. Toliau StackOverFlow atliktoje programuotojų apklausoje buvo matyti, kad Express yra penktoje vietoje iš visų populiariausių WEB karkasų ir pirmoje vietoje iš visų serverio pusės karkasų [6]. Taip pat Express pirmauja ir NPM trends diagramoje [7], kurioje matyti, jog Express karkasas aktyviausiai pirmauja tarp kitų anksčiau paminėtų karkasų. Taigi sistemos serverio pusė bus atliekama panaudojant Node.js (javascript vykdymo aplinką) ir Express karkasą.

2.5.2. React.js

Kita labai svarbi dalis yra išorinė naudotojo sąsaja, kuri lemia sklandžią žmogaus ir kompiuterio sąveiką. Kuriant naudotojo sąsają labai svarbu pasirinkti tinkamą dizainą, pritaikyti derančias spalvas ir atvaizduoti sistemos pranešimus taip,

kad naudotojui būtų aišku ir paprasta naudotis visais sistemos funkcionalumais. Norint atlikti šiuos dalykus nepriekaištingai, reikia pasirinkti tokį naudotojo sąsajos karkasą, kuris leistų įgyvendinti visus suplanuotus funkcionalumus interneto naršyklėje. Vieni iš populiariausių ir plačiausiai naudojamų karkasų šiuo metu yra trys: Angular, React.js ir Vue.js. Kadangi šie trys karkasai iš esmės yra labai panašūs, nes React ir Vue yra parašyti javascript kalba, React yra sukurtas Facebook kompanijos ir keletos savanorių programuotojų, Vue – Google darbuotojo. Angular yra parašytas TypeScript kalba ir plėtojamas Google kompanijos. Visos trys yra populiarios tarp įvairių didelių kompanijų. Analizuojant šiuos karkasus populiariuose puslapiuose tokiuose kaip *State of Javascript* ar *NPM Trends*, buvo matyti, jog React yra pirmaujantis karkasas tarp dviejų anksčiau minėtų, kadangi *State of Javascript Survey 2019* apklausoje [8], React pirmavo 3 metus iš keturių, šis karkasas nusileido Vue tik 2018 metais. Toliau *NPM trends* sistemos parsisiuntimų diagramoje matyti [9], jog React yra absoliutus lyderis tarp kitų karkasų. Tačiau pasirinkti karkasą neužtenka atsižvelgti tik į diagramas, papildomai reikia atsižvelgti į tai, kuris karkasas yra labiausiai tinkamas planuojamiems sistemos reikalavimams.

- Angular

- Didelės apimties ir daug funkcijų turinčios programos
- Realus laiko aplikacijos
- Programuojama TypeScript kalba
- Taikomas objektinis programavimas

- React

- Nedaug vietos užimančios, greitai sukuriamos ir naudojamos įvairaus dydžio įmonių, sistemos
- Užtikrinančios saugius svetainių kūrimo sprendimus
- Kuriamos multi platforminės programos, kurios yra pritaikomos tiek mobiliosioms aplikacijoms, tiek vieno puslapio internetinėms svetainėms

- Vue

- Išmanios, greitos ir našios programos
- Mažos ir nedaug vietos užimančios programos

Pereinant prie išvadų, buvo pasirinktas React naudotojo sąsajos karkasas. Kadangi naudojant šį karkasą, programos yra kuriamos naudojant javascript kalbą, taip pat kuriant sporto programos generavimo sistemą yra norima sukurti vieno puslapio saugią sistemą, kuri būtų lengvai naudojama visuose įrenginiuose, palaikančiuose internetinę naršyklę, taip pat būtų lengvai plėtojama ir pritaikoma didesnėse kompanijose.

2.5.3. MongoDB

Renkantis duomenų bazę, vienas didžiausių sprendimų yra pasirinkti reliacinę (SQL) ar nereliacinę (NoSQL) duomenų struktūrą. Nors abu variantai yra plačiai naudojami įvairiose sistemose, tačiau yra keletas esminių skirtumų, į kuriuos darbo autorius privalėtų atsižvelgti.

Reliacinėms (SQL) duomenų bazėms aprašyti yra taikomas reliacinis duomenų aprašymo modelis, kuris remiasi predikatų logika, bei aibių teorija. Šio modelio pagrindinis elementas yra reliacinė lentelė. Kuri susideda iš stulpelių (atributų) ir eilučių (duomenų bazių įrašų). SQL duomenų bazės yra galingos, saugios ir puikiai atlieka sudėtingas užklausas. Tačiau norint pradėti naudoti SQL duomenų bazes, privaloma iš anksto nusistatyti duomenų struktūrą.

NoSQL duomenų bazės neturi griežtos struktūros ir nėra reliacinė. Šios duomenų bazės duomenys yra saugojami dokumentuose JSON (angl. JavaScript Object Notation) formatu, sudarytu iš atributo ir reikšmės porų. Verta paminėti ir tai, kad ši duomenų bazė saugo dokumentus kolekcijose, skirtingai nei reliacinėse duomenų bazėse, kurios saugo dokumentus duomenų lentelėse. Kolekcija yra grupė susijusių dokumentų, kurie turi jiems priskirtus indeksus. Taigi naudojant NoSQL duomenų bazę gaunami papildomi privalumai, tokie kaip:

- Galima kurti dokumentus be nurodytos duomenų struktūros
- Kiekvienas dokumentas gali turėti savo unikalią struktūrą
- Sinatksė įvairiose duomenų bazėse gali skirtis
- Galima keisti duomenų struktūrą betkuriuo projekto vystymo momentu

Atsižvelgus į visus NoSQL duomenų bazės privalumus, buvo nuspręsta pasirinkti NoSQL duomenų bazę. Kitas žingsnis, kuris yra taip pat labai svarbus, pasirinkti atitinkamą NoSQL tipo duomenų bazę. Vienos iš populiariausių NoSQL duomenų bazių yra *MongoDB*, *Apache Cassandra*, *Google Cloud BigTable*. Apžvelgus visas šias duomenų bazes, buvo nuspręsta pasirinkti MongoDB duomenų bazę, kadangi ji yra nemokama, atviro kodo dokumentinė duomenų bazės

valdymo sistema, kuri yra plačiai naudojama kartu su Node.js (javascript vykdymo aplinka). Dažną MongoDB naudojimą kartu su Node.js lemia tai, jog šios technologijos turi *mongoose* objektų duomenų modeliavimo (*ODM*) biblioteką, kuri valdo ryšius tarp duomenų bei supaprastina kodo ir duomenų bazių sąveiką.

3. Sistemos funkcionalumo įgyvendinimas

Suplanavus visą sistemos funkcionalumą, pasirinkus tinkamiausias technologijas ir nusibraižius reikalingiausias diagramas, buvo pereita prie sistemos kūrimo. Kuriamoje internetinėje svetainėje pagal naudotojo tipą bus ir skirtingas sistemos funkcionalumas. Vienas iš pagrindinių funkcijų sportuojantiems žmonėms, kuris ją išskiria iš anksčiau minėtų ir išanalizuotų sistemų yra unikalus sporto programos generavimas pagal atitinkamus naudotojo kriterijus. Ši pagrindinė funkcija susideda iš keletos mažesnių funkcijų, kurios atlieka tam tikrus skaičiavimus ir grąžina galutinį rezultatą. Visi funkcijų skaičiavimai yra susiję su moksliniais straipsniais ir literatūriniais šaltiniais, tam tikrose vietose yra taikomos matematinės formulės, kurios jau ilgą laiką yra naudojamos apskaičiuoti žmogaus kūno masės indeksą ar procentinį kūno riebalų santykį. Be šios funkcijos, sportuojančio žmogaus vaidmenį sistemoje pasirinkę naudotojai, taip pat galės pirkti grupines treniruotes ir rezervuoti trenerio laiką asmeninėms treniruočioms. Naudotojams, kurie pasirinkę trenerio vaidmenį sistemoje, pagrindinis funkcionalumas bus darbo valandų nustatymas, pagal kurį sportininkai galės rezervuoti trenerį asmeninėms treniruočioms. Taip pat treneris galės skelbti grupines treniruotes ir taip pritraukti daugiau sportuojančių asmenų į renkamas grupines treniruotes. Taigi šiame skyriuje detaliai aprašysiu kaip ir kodėl naudojami tam tikri skaičiavimai ir formulės individualioms programoms kurti ir komponentai rezervacijoms ir pirkimams vykdyti.

3.1. Sporto programos generavimas

Sporto programos generavimas yra viena iš pagrindinių programos funkcionalumų, ji bus prieinama tik naudotojams, pasirinkusiems sportininko vaidmenį sistemoje ir susikūrus naudotojo profilį. Verta paminėti ir tai, kad naudotojams sistema suteiks galimybę generuoti sporto programą daugiau nei vieną kartą, kadangi naudotojas, norintis gauti treniruočių programą, privalės užpildyti formą su duomenimis, kurie ateityje gali keistis.

Šią funkciją galima rasti paspaudus mygtuką sistemos naršyklės juostoje "Wor-

kout Plan" ir patekus į treniruočių programos generavimo ir programos atvaizdavimo puslapį. Naudotojas patekęs į šį puslapį gali matyti apskaičiuotą rekomenduojamą pulsą, atliekant ištvermės reikalaujančius pratimus. Maksimalus pulsas yra apskaičiuojamas pasinaudojant šia formule:

$$\text{Maksimalus pulsas} = 220 - \text{amžius}$$

Knygoje "*Heart Rate Training*" [10] yra rašoma, jog norint pasiekti geriausių rezultatų atliekant kardio pratimus, rekomenduojamas pulsas yra 60-75% nuo maksimalaus pulso, taigi kuriamoje sistemoje buvo nuspręsta naudoti šio skaičiaus vidurkį - 67%. Toliau treniruočių programos puslapio formoje, naudotojas pasirenka norimą treniruočių sudėtingumo lygį, kuris privalo būti įvertintas pagal naudotojo patirtį. Pasirinkimuose yra pradedančiojo ir turinčio patirtį parinktys. Kita parinktis yra dienų skaičius per savaitę (kiek dienų per savaitę naudotojas gali skirti sportui). Naudotojas gali rinktis nuo dviejų iki šešių dienų per savaitę, kadangi sportuojant vieną dieną per savaitę, pasiekti tam tikrų rezultatų neužtenka, o sportuojant septynias dienas per savaitę, gali įvykti persitreniravimas ir tokiu momentu rezultatai gali smukti žemyn. Kadangi intensyviai sportuojant ir sunkiai apkraunant raumenis, daugiau laiko reikia skirti poilsiui ir jėgų atstatymui, nes raumenų augimas ir fizinių duomenų progresavimas vyksta tik poilsio metu. Paskutinis pasirinkimas yra treniruočių tikslas, šioje vietoje naudotojas turi tris pasirinkimus: svorio metimas, raumeninės masės auginimas, fizinės būklės gerinimas. Sekančius reikalingus duomenis sistema pasiims iš jau sukurto sportininko profilio, kadangi norint sugeneruoti sau unikalią programą, naudotojas visų pirma privalo užpildyti sportininko profilio anketą, kurioje užpildoma forma su gimimo data, lytim, svoriu, ūgiu, kaklo, juosmens ir klubų apimtimis, kurios reikalingos norint apskaičiuoti kūno masės indeksą ir procentinį kūno riebalų indeksą. Taigi, atlikus visus šiuos veiksmus galima spausti *Submit* mygtuką. Šiuo momentu visi duomenys nukeliauja į sporto programos generavimo algoritmą, kuris yra serverio pusėje - *Helpers* direktorijoje, kur ir saugojamos visos generavimo funkcijos.

Visų pirma, gavus visus duomenis yra kviečiama *GeneratePlan* funkcija, šioje funkcijoje yra aprašomi objektai, kurie gauna visus pratimus iš duomenų bazės. Pirmas iš objektų *isolationExercises*, kuriame yra filtruojami visi izoliuoti pratimai skirti krūtinės, pečių, kojų, nugaros, trigralviui ir dvigralviui raumenims ir pažengusiems naudotojams. Izoliuoti pratimai yra tie, kuriuos atliekant dirba tik vienas tam tikras raumuo. Toliau yra aprašomas bazinių pratimų objektas. Baziniai pratimai yra vadinami tie, kuriuos atliekant dirba keli sąnariai ir daugiau

nei viena raumenų grupė. Toliau yra aprašomi lygiai tokie pat objektai izoliuotų ir bazinių pratimų, tik jiems priskiriamas dar vienas papildomas kriterijus, kuris yra skirtas pradedantiesiems sportininkams. Kiti objektai yra apšilimo ir ištvermės pratimai. Apšilimo pratimai yra daromi kiekvieną treniruotę, norint išvengti traumų ir gauti geriausią rezultatą iš treniruotės. Ištvermės pratimų dažnumas ir trukmė bus parinkti atsižvelgiant į naudotojo tikslus. Po visų šių objektų, funkcijoje taip pat aprašytas masyvas, su pagrindinėmis raumenų grupėmis tiek moterims, tiek vyrams:

- Krūtinė
- Nugara
- Kojos
- Pečiai
- Dvigalvis raumuo
- Trigalvis raumuo

Šis masyvas yra skirtas tam, jog atitinkamai dienai būtų parinktas tam tikras raumuo ir kad raumenų grupės nesidubliuotų. 5 paveiksle, galite matyti kaip yra filtruojami izoliuoti pratimai iš visų pratimų sąrašo. Norint gauti bazinius pratimus ar pratimus skirtus pradedantiesiems, papildomai yra pridedamas sunkumo lygio (*level*) atributas su nustatyta *beginner* reikšme ir pratimo tipas (*mechanicsType*) pakeičiamas į *compound* reikšmę.

```
const isolationExercisesForBeginner = {
  chest: exercise.filter(x =>
    x.muscles === 'chest' && x.mechanicsType === 'isolation'),
  back: exercise.filter(x =>
    (x.muscles === 'middle back' || x.muscles === 'lower back' || x.muscles === 'lats')
    && x.mechanicsType === 'isolation'),
  shoulders: exercise.filter(x =>
    x.muscles === 'shoulders' && x.mechanicsType === 'isolation'),
  legs: exercise.filter(x =>
    (x.muscles === 'calves' || x.muscles === 'legs')
    && x.mechanicsType === 'isolation'),
  triceps: exercise.filter(x =>
    x.muscles === 'triceps' && x.mechanicsType === 'isolation'),
  biceps: exercise.filter(x =>
    x.muscles === 'biceps' && x.mechanicsType === 'isolation'),
  legsForWomen: exercise.filter(x =>
    (x.muscles === 'calves' || x.muscles === 'legs' || x.muscles === 'glutes')
    && x.mechanicsType === 'isolation'),
}
```

5 lentelė. Izoliuotų pratimų filtravimas

Visi šie pratimai bus skirti tiek vyrams, tiek moterims, kadangi remiantis [11], buvo išaiškinta jog nėra skirtumo tarp to, kas atlieka pratimą, ar vyras, ar moteris. Didžiausias skirtumas tarp merginų ir vyrų treniruočių yra pasirenkami svoriai ir pakartojimų bei serijų skaičius, kurie bus aptarti sekančiu žingsniu.

Po aprašytų objektų ir masyvo yra kviečiama kita funkcija *HowManyMusclesPerDay*, kuri apskaičiuoja kiek raumenų grupių naudotojui teks treniruoti per dieną. Ši funkcija tikrina, kiek kartų per savaitę sportininkas sportuos ir atitinkamai parenka treniruojamų raumenų grupių skaičių per dieną.

```
let muscleNames = ['chest', 'back', 'shoulders', 'legs', 'triceps', 'biceps']

function HowManyMusclesPerDay(){
  if(daysPerWeek == 2){
    musclesPerDay = muscleNames.length / 2;
  } else if(daysPerWeek == 3){
    musclesPerDay = muscleNames.length / 3;
  } else if(daysPerWeek == 6){
    musclesPerDay = muscleNames.length / 6;
  }
}
```

6 lentelė. Raumenų grupių skaičiaus per dieną apskaičiavimas

Kaip yra pavaizduota 6 lentelėje, *HowManyMusclesPerDay* funkcija dalina *muscleNames* masyvo dydį iš atitinkamo skaičiaus, kadangi norima, jog sportininkas spėtų per savaitę ištreniruoti visas raumenų grupes taip, jog ramuo nebūtų per daug apkrautas ar išdirbtas per mažai. Taigi jeigu naudotojas sportuos tik dvi dienas per savaitę, jis turės treniruoti tris atskirus raumenis per vieną dieną, tai leis optimaliai išnaudoti visas darbinės savaitės dienas. Jeigu sportininkas sportuos 3 dienas, jam teks atlikti po dvi raumenų grupes ir atitinkamai jeigu sportuos 6 dienas, jis treniruos tik vieną raumens grupę per dieną. Šioje funkcijoje skaičiuojamos tik trys dienos, visas kitas dienas apskaičiuos kita funkcija, kuri pavaizduota 7 lentelėje. Ši funkcija yra naudojama tam, jog kas antrą dieną sportininkas gautų skirtingą raumenų grupių skaičių. Tarkim sportuojantis žmogus pirmą treniruotės dieną treniruos dvi raumenų grupes, antra dieną - vieną raumenų grupę, trečią dieną vėl dvi. Panaši logika yra naudojama ir sportuojant penkias dienas per savaitę, tik tuomet atletas treniruos dvi raumenų grupes tik pirmą treniruočių dieną, visas kitas dienas, atletas treniruos tik po vieną raumenų grupę.

```

if(daysPerWeek == 4){
  for(let i = 1; i <= daysPerWeek; i++){
    if(i % 2 === 0){
      for(let k = 1; k <= 2; k++){
        // Vienos dienos programos generavimas su dviem raumenų grupėmis
      }
    } else {
      for(let k = 1; k <= 1; k++){
        // Vienos dienos programos generavimas su viena raumenų grupe
      }
    }
  }
}

```

7 lentelė. Raumenų grupių skaičiaus per dieną apskaičiavimas

Toliau pereinama prie vienos dienos treniruotės generavimo. Šio ciklo viduje yra kviečiama funkcija *GetCardioExercise*, kuri yra pavaizduota 8 paveiksle. Šios funkcijos viduje yra tikrinamas naudotojo tikslas. Jeigu naudotojo tikslas yra numesti svorio, programa sportininkui parinks atsitiktinį ištvermės tipo pratimą kiekvieną treniruotės dieną. Jeigu naudotojo tikslas yra raumeninės masės auginimas, tuomet programa jam parinks ištvermės pratimą tik kas trečią treniruotės dieną ir jeigu sportininko tikslas tapti stipresniam ir sveikesniam, tuomet sistema sugeneruos ištvermės pratimą, kas antrą treniruotės dieną. Suradus atsitiktinį ištvermės pratimą ir nustačius pratimo trukmę, šie duomenys yra išsaugojami į vienos dienos treniruočių programą.

```

if(level === 'beginner'){
  if(goal === 'lose fat'){
    min = 35;
    cardio = cardioExercises[Math.floor(Math.random() * cardioExercises.length)]
    dailyExercises.push({ min, exercise: cardio })
  } else if(goal === 'build muscle'){
    min = 15;
    // kviečiama funkcija kur cardio pratimas įdedamas kas 3 dienas
  } else if(goal === 'get stronger'){
    min = 20;
    // kviečiama funkcija kur cardio pratimas įdedamas kas 2 dienas
  }
} else if(level === 'intermediate'){
  if(goal === 'lose fat'){
    min = 45;
    cardio = cardioExercises[Math.floor(Math.random() * cardioExercises.length)]
    dailyExercises.push({ min, exercise: cardio })
  } else if(goal === 'build muscle'){
    min = 20;
    // kviečiama funkcija kur cardio pratimas įdedamas kas 3 dienas
  } else if(goal === 'get stronger'){
    min = 30;
    // kviečiama funkcija kur cardio pratimas įdedamas kas 2 dienas
  }
}

```

8 lentelė. Ištvermės pratimų parinkimas

Pasibaigus *GetCardioExercise* funkcijai, pereinama prie kitos funkcijos *GetAbsExercise*. Kadangi pilvo preso raumenys yra vieni iš greičiausiai atsigaunančių žmogaus raumenų, buvo nuspręsta jog ši funkcija bus kviečiama kas antrą treniruočių dieną. Ši funkcija parinks du atsitiktinius pratimus šiai raumenų grupei iš visų pilvo raumenų pratimų sąrašo ir pridės į dienos pratimų masyvą.

Išvykdžius *GetCardioExercise* ir *GetAbsExercise* funkcijas, toliau yra pereinama prie ciklo, kurio viduje yra kviečiama funkcija *WorkoutForOneDayGeneration*. Ši funkcija atsitiktine tvarka parenka vieną raumenų grupę kuri bus treniruojama ir, kuri dar nėra treniruota būtent tą savaitę. Po šio veiksmo, šis raumuo yra ištrinamas iš raumenų grupių masyvo, kad kitą kartą vykdant funkciją, būtų išvengiama to paties raumens grupės treniruočių pasikartojimo. Kadangi šios sistemos sugeneruotose treniruotėse viena raumenų grupė bus treniruojama tik vieną kartą per savaitę. Toliau yra sąlyga, kuri tikrina ar tai yra pirmas pratimas dienoje ir jeigu sąlyga yra teisinga, pridedami apšilimo ir tempimo pratimai. Jeigu sąlyga neteisinga ir apšilimo pratimai jau yra parinkti, tuomet pereinama prie kitos funkcijos *HowManyExercisesPerMuscle*, kurios dalis yra pavaizduota 9 pav.

```
if (daysPerWeek == 3){
    if(gender === 'male'){
        if(level === 'beginner'){
            if(goal === 'lose fat'){
                isolationNumber = 3;
                compoundNumber = 0;
            } else if(goal === 'build muscle'){
                isolationNumber = 1;
                compoundNumber = 2;
            } else if(goal === 'get stronger'){
                isolationNumber = 2;
                compoundNumber = 1;
            }
        } else if(level === 'intermediate'){
            if(goal === 'lose fat'){
                isolationNumber = 4;
                compoundNumber = 1;
            } else if(goal === 'build muscle'){
                isolationNumber = 2;
                compoundNumber = 3;
            } else if(goal === 'get stronger'){
                isolationNumber = 3;
                compoundNumber = 2;
            }
        }
    }
}
```

9 lentelė. Izoliuotų ir bazinių pratimų skaičiaus parinkimas

Ši sąlyga yra vykdoma *HowManyExercisesPerMuscle* funkcijos viduje, iš karto po apšilimo ir ištvermės pratimų funkcijų. Ši sąlyga atsižvelgia į dienų skaičių, kurių pasirinkęs naudotojas, naudotojo lytį, patirtį ir tikslą. Sąlygose yra nustatoma kiek izoliuotų ir bazinių pratimų privalės atlikti sportininkas. Izoliuotų ir bazinių pratimų skaičius priklauso nuo naudotojo patirties ir tikslo.

Baziniai pratimai yra labiau skirti pažengusiems ir norintiems priaugti raumeninės masės sportininkams [12], kadangi juos atliekant yra apkraunami keli sąnariai ir kelios raumenų grupės, tai leidžia naudoti didesnius svorius juos atliekant ir taip greičiau didinti raumeninę masę, kadangi naudojant didesnius svorius yra greitesnis raumeninės masės prieaugis, tačiau ir traumų tikimybė padidėja. Taigi norint išvengti šios problemos, pradedantiesiems atletams, kurių tikslas taip pat yra priaugti raumeninės masės ir kurie turi atlikti didesnę skaičių bazinių pratimų, jiems bus parenkami tokie baziniai pratimai, kurie nereikalauja papildomo svorio ar štangos - prisitraukimai, atsispaudimai, pritūpimai ar lygiagretės. Visi šie paminėti pratimai yra baziniai, tačiau traumų tikimybė yra daug mažesnė, nei atliekant kitus bazinius pratimus, kuriems reikalingi papildomi svoriai - pritūpimai su štangą ar mirties trauka, skirtą nugaros raumenims.

Izoliuoti pratimai yra labiau skirti pradedantiesiems atletams ir norintiems atsi-kratyti papildomų kilogramų. Izoliuoti pratimai leidžia akcentuoti vieną raumens grupę ir atlikti daugiau pakartojimų, bei serijų, kadangi išvengiama traumos tikimybė. Atliekant didesnę pakartojimų skaičių yra deginama daugiau kalorijų, taip yra lengviau mesti svorį ir įgauti geresnę fizinę formą.

Po pratimų skaičiaus parinkimo yra pereinama prie ciklo, kuris pavaizduotas 10 pav. Šiame paveiksle yra pavaizduotas izoliuotų pratimų parinkimo ciklas, lygiai toks pat yra naudojamas ir baziniams pratimams gauti, kuri vyksta iš karto po izoliuotų pratimų ciklo. Šio ciklo viduje yra kviečiamos funkcijos *GetIsolationExerciseForBeginner* ir *GetIsolationExercise*, atitinkamai nuo atleto patirties. Skirtingi pratimai pradedantiesiems ir pažengusiems sportininkams yra parenkami dėl to, jog yra pratimų, kurie turi didesnę traumų tikimybę, tokie kaip mirties trauka. Šiam pratimui atlikti reikia daugiau patirties, nes atliekant šį pratimą yra apkraunamas beveik visas kūnas ir yra naudojami dideli svoriai.

```

for(let j = 1; j <= (nustatytas izoliuotų pratimų skaičius); j++){
    if(level === 'beginner'){
        GetIsolationExerciseForBeginner()
    } else {
        GetIsolationExercise()
    }
}

```

10 lentelė. Izoliuotų pratimų parinkimas

Parinkus atsitiktinius bazinius ir izoliuotus pratimus, skirtus tam tikroms raumenų grupėms, yra pereinama prie funkcijų - skirtų serijų ir pakartojimų skaičiui, bei poilsio trukmei gauti, kurie bus parenkami atitinkamai nuo pratimo tipo, ar tai būtų izoliuotas, ar bazinis pratimas. 11 pav. yra pavaizduotas pakartojimų, serijų ir poilsio trukmės sekundėmis nustatymas, atsižvelgiant į sportininko patirtį ir tikslą. Pakartojimų ir serijų skaičius yra parenkamas atsižvelgus į straipsnį [13] iš patikimo fitneso tinklalapio *fitness.com*. Pradedantiesiems sportuoti bus rekomenduojama naudoti mažesnius svorius ir atlikti daugiau pakartojimų bei mažiau serijų, kad raumuo iš lėto priprastų prie krūvio ir būtų išvengta persitreniravimo pirmosiomis sporto savaitėmis. Pažengusiems naudotojams bus priskiriamos serijos ir pakartojimai pagal atitinkamus kriterijus, norint priaugti raumeninės masės bus rekomenduojama rinktis didesnius svorius ir atlikti mažiau pakartojimų, tačiau daugiau serijų su ilgesnėmis pertraukomis [14]. Norint sulieknėti ar įgauti geresnę fizinę formą bus parenkamos mažesnės pertraukos, daugiau pakartojimų ir serijų. Verta paminėti ir tai, jog būtent dvigalviui raumeniui bazinių pratimų nėra, taigi bazinių pratimų parinkimo funkcijoje yra tikrinama ar parinktas raumuo nėra dvigalvis raumuo, tačiau jeigu yra parinktas būtent šis raumuo, tuomet bazinių pratimų parinkimas yra praleidžiamas ir pereinama prie izoliuotų pratimų, skirtų dvigalviui raumeniui treniruoti.

```

if(gender === 'male'){
  if(level === 'beginner'){
    if(goal === 'lose fat'){
      sets = 3;
      reps = 10;
      rest = 60;
    } else if(goal === 'build muscle'){
      sets = 4;
      reps = 6;
      rest = 120;
    } else if(goal === 'get stronger'){
      sets = 3;
      reps = 8;
      rest = 90;
    }
  }
}

```

11 lentelė. Serijų, pakartojimų ir poilsio trukmės parinkimas

Pasibaigus šioms funkcijoms į vienos dienos masyvą yra įrašomi visi pratimai ir ciklas toliau kartojasi. Galiausiai turime visas dienas, kurias naudotojas gali sportuoti ir visi pratimai yra grąžinami *workout* masyve. Norint labiau išsigilinti į treniruočių generavimo logiką, yra pateikta pavyzdinė lentelė 12, kuri atvaizduoja treniruotės duomenų nustatymą, pagal atitinkamus naudotojo duomenis.

days	3								
gender	Male								
level	beginner				Intermediate				
goal	Lose	Gain	Stay		Lose	Gain	Stay		
number of compound	1	4	3		1	4	3		
number of isolation	5	2	3		5	2	3		
muscles per day	2				2				
exercise per muscle	3				3				
exercise type	compound isolation	compound isolation	compound	isolation	compound isolation	compound isolation	compound	isolation	
sets	3 3	4 3	3	3	3 3	5 4	4	3	
reps	10 12	6 8	8	10	10 15	5 8	6	10	
rest	60 45	120 120	90	90	45 45	120 120	90	90	

12 lentelė. Treniruotės generavimo informacija

3.2. Kūno masės indekso apskaičiavimas

Kūno masės indeksas yra skaičiuojamas panaudojant matematiko *Adolphe Quetelet* formulę [15]:

$$KMI = \frac{s}{u^2}$$

- s - Svoris kilogramais
- u - Ūgis metrais

Ši formulė yra kviečiama tuomet, kai naudotojas užpildo sportininko profilį, kuriame nurodo reikalingus parametrus ir spaudžia *Submit* mygtuką. Įvykus sėkmingam formos užpildymui, naudotojas matys vieną iš trijų rodiklių:

- Normali masė
- Antsvoris
- Svorio trūkumas

Sportininkas, norėdamas generuoti sau unikalią sporto programą, galės remtis rodikliu ir pagal tai pasirinkti treniruočių tikslą. Kadangi kūno masės indekso formulė buvo sukurta tarp 1830 m. ir 1850 m. ji nėra labai tiksli ir tam tikrais atvejais rodiklis gali būti klaidingas. Dėl šios priežasties buvo priimtas sprendimas KMI rezultatą pritaikyti tik kaip rekomendaciją ir leisti naudotojui pasirinkti treniruočių tikslą pačiam, o ne automatiškai parinkti pagal KMI rodiklį. Taip pat sportininkas galės stebėti savo KMI rodiklius ir progresą diagramoje.

3.3. Procentinis kūno riebalų kiekis

Procentinis kūno riebalų kiekis yra gaunamas pagal šiuos parametrus: naudotojo lytį, ūgį (centimetrais), kaklo, juosmens ir klubų apimtis (centimetrais). Visus šiuos parametrus sistema gaus tuomet, kai naudotojas užpildys sportininko profilį ir paspaus *Submit* mygtuką. Tuomet sistema iškvies funkciją kuri atliks visus matematinius skaičiavimus pagal šią formulę [16], priklausomai ar naudotojas yra vyras, ar moteris:

$$\text{Vyrams } KRD = \frac{495}{(1.0324 - 0.19077(\log(j - ka)) + 0.15456(\log(u))) - 450}$$

$$\text{Moterims } KRD = \frac{495}{(1.29579 - 0.35004(\log(j + kl - ka)) + 0.22100(\log(u))) - 450}$$

- j - juosmuo
- ka - kaklas
- u - ūgis
- kl - klubai

Funkcijai atlikus skaičiavimus, bus grąžinama procentinė kūno riebalų dalis ir suapvalinama iki dešimtosios dalies. Šis rezultatas bus įrašomas į naudotojo profilį ir vaizduojamas diagramoje su visais kitais rodikliais (išskyrus ūgio). Naudotojui bus rekomenduojama kas mėnesį atnaujinti apimtis ir taip stebėti kiekvieno iš jų pasikeitimus atliekant sistemos sukurta sporto programą.

3.4. Profilio sudarymas

Kadangi sistema turi du skirtingus sistemos naudotojus, jie turės ir atskirus profilius su skirtingais privalomais užpildyti laukais. Naudotojas tik užsiregistravęs, patvirtinęs registraciją ir prisijungęs prie sistemos pirmą kartą, turės užpildyti profilio duomenis. Trenerio profilis susidarys iš gimimo datos, išsilavinimo ir profesinių pasiekimų, kurie galės būti nuolatos atnaujinami ir šiuo metu ypač populiarių, socialinių tinklų, tokių kaip Youtube, Instagram bei Facebook. Tokį sprendimą, kaip dalinimąsi socialiniais tinklais, priėmiau dėl to, jog labai daug trenerių nuolatos dalinasi savo pasiekimais bei vykdoma veikla būtent šiuose puslapiuose, taigi iš trenerio profilio, vienu mygtuko paspaudimu, naudotojas bus nukreipiamas į pasirinktą trenerio socialinį puslapį. Sportuojančio asmens profilis susidarys iš jo asmeninių fizinių rodiklių, tokių kaip svoris, ūgis, amžius ir tikslai.

3.5. Grupinių treniruočių skelbimas

Treneriams susikūrus savo profilį, jiems atsiranda galimybė kurti grupines treniruotes. Norėdami sukurti grupinę treniruotę, jiems privaloma nurodyti treniruotės pavadinimą, trumpą aprašymą ir kiek vietų skirta grupinėje treniruotėje, jog naudotojai perkantys treniruotę galėtų matyti kiek vietų yra likę. Kiti duomenys apie treniruotę yra tikslus adresas, treniruotės sunkumas, kuris prasideda nuo pradedančiojo, vėliau pereinama prie vidutinio ir baigiasi ties eksperto lygiu. Taip pat nurodomas tikslus laikas, kuriuo vyks treniruotė. Treniruotei pasibaigus, ji bus perkeliama į trenerio, sukūrusio treniruotę, archyvą. Taip pat labai svarbus akcentas, kurį treneris privalės nurodyti yra treniruotės kaina.

3.6. Trenerio rezervacija

Susikūrus trenerio profilį, taip pat galima nurodyti ir savo darbo valandas, bei dienas, kuriomis treneris dirba individualiai su klientais. Nurodžius šią informaciją, naudotojas turintis sportininko profilį gali užsirezervuoti trenerį tuo metu, kai trenerio darbo valandos nėra užimtos kitų naudotojų. Rezervacijos/Pirkimo vykdymas bus atliekamas panaudojant Stripe įskiepi, kuris yra saugus ir turi aiškią dokumentaciją bei yra lengvai naudojamas kartu su Node karkasu. Sėkmingos rezervacijos atveju, šios valandos nebus prieinamos kitiems sportininkams ir rezervaciją atlikęs sportininkas yra įtraukiamas į asmeninę trenerio lentelę, kurioje jis gali matyti visus savo klientus su jiems priskirtu laiku.

3.7. Asmeninių ir grupinių treniruočių pirkimas

Asmeninių ir grupinių treniruočių pirkimas bus prieinamas naudotojams, kurie bus užsiregistravę kaip sportininkai. Pirkimas vyks tokiu pat principu, kaip vyks ta prieš tai paminėtame trenerio rezervacijos skyriuje, panaudojant Stripe įskiepi. Grupinės treniruotės bus randamos treniruočių skiltyje. Naudotojas galės išsirinkti sau tinkamą treniruotę pagal tipą, datą ar sudėtingumo lygį. Norintis gauti individualią treniruotę, naudotojas ją galės rasti trenerių skiltyje. Radęs trenerį, naudotojas galės peržiūrėti visus atsiliepimus iš kitų sportininkų apie trenerį, matyti jo patirtį, išsilavinimą bei trenerio socialinius tinklus. Jeigu treneris atitiks sportininko lūkesčius, jis galės pasiskirti sau laiką, kada bus laisvas treneris. Taigi išsirinkęs norimą treniruotę ar trenerį naudotojas privalės atlikti apmokėjimą savo banko kortele. Įvykus sėkmingam apmokėjimui, naudotojas gaus patvirtinimą į savo elektroninį paštą su treniruotės pavadinimu, adresu, laiku ir pinigų suma, kurią sumokėjo. Jeigu sportininkas rezervavo trenerį, tuomet laiškas bus su trenerio vardu, laiku ir kaina.

3.8. Rezultatų stebėjimas diagramose

Galimybė stebėti rezultatus bus prieinama tik sportininko profilį susikūrusiems naudotojams. Pildant sportininko profilį bus privaloma nurodyti naudotojo, kuriančio anketą svorį, ūgį, gimimo datą, lytį, bei kaklo, juosmens ir klubų apimtis iš kurių ir bus apskaičiuojamas kūno masės indeksas ir procentinis kūno riebalų santykis. Užpildžius anketą diagramos susikurs automatiškai ir jas bus galima stebėti naudotojo profilyje. Diagramoms kurti buvo naudojama ZingChart biblioteka, kuri turi puikią dokumentaciją ir gerą suderinamumą su React karkasu. Norint stebėti savo progresą, naudotojui bus rekomenduojama kas mėnesį atnaujinti savo duomenis užpildant formas. Vienoje formoje, kuri skaičiuoja kūno masės indeksą užteks įvesti savo esamą svorį ir ūgį. Sprendimas leisti atnaujinti savo ūgio rodiklius buvo priimtas dėl to, jog sistema galės naudotis ir jaunesni žmonės, kurių ūgis vis dar keičiasi kas keletą mėnesių. Kita forma yra procentinis kūno riebalų santykis, norint atnaujinti šią diagramą, naudotojas privalės nurodyti naujus duomenis susijusius su kaklo, juosmens, klubų apimtimis bei ūgiu. Taip pat apskaičiuoti šiai formai bus reikalingas ir lyties parametras, tačiau jis bus gaunamas iš sportininko profilio ir jo atnaujinti daugiau nereikės.

3.9. Saugumas

Kuriant internetinę sistemą vienas iš svarbiausių aspektų yra saugumas. Kibernetiniai incidentai gali turėti didelės įtakos kuriamai sistemai, kadangi jie gali sutrikdyti ne tik sistemos veiklą, tačiau ir nutekinti konfidencialius naudotojų duomenis į viešumą, kas lems naudotojų pastikėjimo praradimą naudojantis internetine svetaine. Norint užtikrinti sistemos saugumą buvo priimti keli sprendimai, tiek iš sistemos vidinės pusės (angl. *backend*), tiek iš išorinės naudotojo sąsajos (angl. *front-end*).

Išorinėje naudotojo sąsajoje yra naudojamas *protectedRoutes* komponentas. Naudotojui bandant patekti į tam tikrą puslapį, komponentas visada patikrina ar naudotojo rolė atitinka puslapio, į kurį norima patekti, rolę. Kadangi norint aprašyti tam tikrą puslapį programinėje dalyje, reikia nurodyti ir vaidmenį, kurį turintis naudotojas galės patekti į jį. Sistemoje gali būti trys prisijungę vaidmenys ir naudotojas gali turėti tik vieną iš jų:

- Administratorius
- Treneris
- Sportininkas

Kaip galite pastebėti 13 pav., komponentas tikrina ar naudotojas yra prisijungęs ir jeigu yra prisijungęs, tuomet komponentas filtruoja visus adresus, kurie sutampa su naudotojo role sistemoje. Jeigu naudotojas nėra prisijungęs, jis yra grąžinamas į pagrindinį puslapį naudotojo, kuris yra prieinamas visiems tos grupės naudotojams. Sistemos vidinėje pusėje naudotojų autentifikacijai ir autorizacijai yra naudojami JWT (*JSON Web Token*) ir slapukai (angl. *Cookies*), kurie užtikrina sklandų ir saugų naudojimąsi internetine svetaine.

```
session ?
routes.filter(x => x.role === session.role).map(x =>
  <Route path={x.path} component={x.component} exact={x.exact} />
): <Redirect to="/" />
```

13 lentelė. Serijų, pakartojimų ir poilsio trukmės parinkimas

4. Išvados ir rekomendacijos

Darbo metu analizuojant panašias sistemas, jų projektinius sprendimus ir naudotojų poreikius, buvo nustatyta didelė sistemos paklausa tiek Lietuvos, tiek užsienio rinkoje. Remiantis panašiais projektiniais sprendimais, buvo sukurta internetinė sistema naudotojams, norintiems pasiekti geresnių rezultatų tiek sporto, tiek treniravimo srityje. Sistemos pagrindinė funkcija - vienoje sistemoje naudotojams leisti generuoti unikalias treniruočių programas, bei pirkti trenerių paslaugas asmeninėms treniruotėms, naudojantis Stripe programos įskiepiu. Siekiant pagerinti sistemos veiklą ir patrauklumą, buvo pridėta trenerių reitingavimo, atsiliepimų rašymo bei peržiūrėjimo funkcionalumas, tai leis naudotojams išsirinkti geresnį trenerį ir išvengti nesusipratimų. Ši sistema galėtų būti pritaikoma ir toliau plėtojama įvairių sporto klubų, kurie plėtoja asmeninių trenerių ir grupinių treniruočių veiklą.

Siekiant užtikrinti sistemos sėkmę tiek Lietuvos, tiek užsienio rinkoje, ateityje planuojama papildyti sistemos funkcionalumą ir leisti naudotojams susidaryti unikalų mitybos planą. Mityba yra vienas svarbiausių dalykų norint pasiekti užsibrėžtus tikslus. Tačiau visame pasaulyje daugėjant žmonių populiacijai atsiranda ir didėjantis maisto trūkumas. Kaip teigiama *"Global Report on Food Crises"* ataskaitoje [17], per 2019 metus maisto išteklių trūkumas išaugo daugiau nei 20mln. ir pasiekė 135mln. gyventojų, kuriems kritiškai trūko maisto. Taigi atsižvelgiant į tai, atsiranda poreikis leisti naudotojams sistemoje kurti mitybos planą atsisakant tam tikrų produktų, tokių kaip - mėsa, pieno produktai ir naudoti daugiau augalinės kilmės maisto produktus, kurie gali būti naudojami ir nepasiturinčiose valstybėse. Be maisto produktų trūkumo yra žmonių, kurie propaguoja vegetarizmą ar veganizmą, taigi tai leistų žmonėms susigeneruoti mitybos planą iš jų naudojamų maisto produktų ir taip gauti visas reikiamas maistines medžiagas. Taip pat atsižvelgiant į tai, jog visame pasaulyje daugėja virusinių susirgimų, kas lemia tai, jog yra uždaromi visi sporto klubai ir stabdomos trenerių veiklos, sistemoje veiktų tiesioginės transliacijos iš trenerių pusės, tokiu būdu naudotojai galėtų toliau naudotis trenerių paslaugomis ir sportuoti iš namų kartu su asmeniniu treneriu.

Literatūra

- [1] World Health Organization. Facts about overweight and obesity.
- [2] Betty Pfeferbaum and Carol S. North. Mental health and the covid-19 pandemic. *New England Journal of Medicine*, 0(0):null, 0.
- [3] Michael Gleeson. Immune function in sport and exercise. *Journal of Applied Physiology*, 103(2):693–699, 2007. PMID: 17303714.
- [4] M. Hamer and Y. Chida. Physical activity and risk of neurodegenerative disease: a systematic review of prospective evidence. *Psychological Medicine*, 39(1):3–11, 2009.
- [5] The State Of Javascript. Back end frameworks, 2019.
- [6] StackOverFlow. Developer survey results 2019.
- [7] Npm Trends. Express framework, 2020.
- [8] The State Of Javascript. Front end frameworks, 2019.
- [9] Npm Trends. Angular vs react vs vue, 2020.
- [10] R. Benson and D. Connolly. *Heart Rate Training*. Human Kinetics, Incorporated, 2019.
- [11] Arin Gragossian. Training men vs women.
- [12] Fitness Academy. What are compound exercises, October 31, 2018.
- [13] The Global Fitness Community. How much weight? how many reps?
- [14] Paige Waehner. How many sets you should be doing in a workout, October 17, 2019.
- [15] Diabetes.ca. Body mass index (bmi) calculator.
- [16] James Hodgdon and Marcie Beckett. Prediction of percent body fat for u.s. navy women from body circumferences and height. 06 1984.
- [17] Food Security Information Network. *Global report on food crises*. FSIN, 2020.

Priedai

A. Treniruočių programos generavimo algoritmas

```
1
2  function GeneratePlan(athlete, exercise, daysPerWeek, level, goal){
3    let sets = null;
4    let reps = null;
5    let rest = null;
6    let musclesPerDay = null;
7    let gender = athlete.gender;
8    const isolationExercises = {
9      chest: exercise.filter(x => x.muscles === 'chest' && x.mechanicsType
10         === 'isolation'),
11      back: exercise.filter(x => (x.muscles === 'middle back' || x.muscles
12         === 'lower back' || x.muscles === 'lats') && x.mechanicsType === '
13         isolation'),
14      shoulders: exercise.filter(x => x.muscles === 'shoulders' && x.
15         mechanicsType === 'isolation'),
16      legs: exercise.filter(x => (x.muscles === 'calves' || x.muscles === '
17         legs') && x.mechanicsType === 'isolation') ,
18      triceps: exercise.filter(x => x.muscles === 'triceps' && x.
19         mechanicsType === 'isolation'),
20      biceps: exercise.filter(x => x.muscles === 'biceps' && x.mechanicsType
21         === 'isolation'),
22      legsForWomen: exercise.filter(x => (x.muscles === 'calves' || x.
23         muscles === 'legs' || x.muscles === 'glutes') && x.mechanicsType
24         === 'isolation'),
25    }
26    const compoundExercises = {
27      chest: exercise.filter(x => x.muscles === 'chest' && x.mechanicsType
28         === 'compound'),
29      back: exercise.filter(x => (x.muscles === 'middle back' || x.muscles
30         === 'lower back' || x.muscles === 'lats') && x.mechanicsType === '
31         compound'),
32      shoulders: exercise.filter(x => x.muscles === 'shoulders' && x.
33         mechanicsType === 'compound'),
34      legs: exercise.filter(x => (x.muscles === 'calves' || x.muscles === '
35         legs') && x.mechanicsType === 'compound') ,
36      triceps: exercise.filter(x => x.muscles === 'triceps' && x.
37         mechanicsType === 'compound'),
38      biceps: exercise.filter(x => x.muscles === 'biceps' && x.mechanicsType
39         === 'compound'),
40    }
41    const isolationExercisesForBeginner = {
42      chest: exercise.filter(x => x.muscles === 'chest' && x.mechanicsType
43         === 'isolation' && x.level === 'beginner'),
44      back: exercise.filter(x => (x.muscles === 'middle back' || x.muscles
45         === 'lower back' || x.muscles === 'lats') && x.mechanicsType === '
46         isolation' && x.level === 'beginner'),
47      shoulders: exercise.filter(x => x.muscles === 'shoulders' && x.
48         mechanicsType === 'isolation' && x.level === 'beginner'),
```

```

29 legs: exercise.filter(x => (x.muscles === 'calves' || x.muscles === '
    legs')) && x.mechanicsType === 'isolation' && x.level === 'beginner'
    ) ,
30 triceps: exercise.filter(x => x.muscles === 'triceps' && x.
    mechanicsType === 'isolation' && x.level === 'beginner'),
31 biceps: exercise.filter(x => x.muscles === 'biceps' && x.mechanicsType
    === 'isolation' && x.level === 'beginner'),
32 legsForWomen: exercise.filter(x => (x.muscles === 'calves' || x.
    muscles === 'legs' || x.muscles === 'glutes') && x.mechanicsType
    === 'isolation'),
33 }
34 const compoundExercisesForBeginner = {
35 chest: exercise.filter(x => x.muscles === 'chest' && x.mechanicsType
    === 'compound' && x.level === 'beginner'),
36 back: exercise.filter(x => (x.muscles === 'middle back' || x.muscles
    === 'lower back' || x.muscles === 'lats') && x.mechanicsType === '
    compound' && x.level === 'beginner'),
37 shoulders: exercise.filter(x => x.muscles === 'shoulders' && x.
    mechanicsType === 'compound' && x.level === 'beginner'),
38 legs: exercise.filter(x => (x.muscles === 'calves' || x.muscles === '
    legs')) && x.mechanicsType === 'compound' && x.level === 'beginner')
    ,
39 triceps: exercise.filter(x => x.muscles === 'triceps' && x.
    mechanicsType === 'compound' && x.level === 'beginner'),
40 biceps: exercise.filter(x => x.muscles === 'biceps' && x.mechanicsType
    === 'compound' && x.level === 'beginner'),
41 }
42 const stretchingExercises = exercise.filter(x => x.exerciseType === '
    stretching');
43 const cardioExercises = exercise.filter(x => x.exerciseType === '
    cardio');
44 const absExercises = exercise.filter(x => x.muscles === 'abs')
45 let muscleNames = ['chest', 'back', 'shoulders', 'legs', 'triceps', '
    biceps']
46
47 HowManyMusclesPerDay();
48
49 if(daysPerWeek == 2 || daysPerWeek == 3 || daysPerWeek == 6){ // How
    many days per week athlete can workout
50 let workout = [] // all workout will be passed in this workout array
51 for(let i = 1; i <= daysPerWeek; i++){ // for every workout day
52 let dailyExercises = []; // one day workout array
53 GetCardioExercise(i, dailyExercises)
54 if(i % 2 === 0){ // get abs workout every two workout days
55 GetAbsExercise(absExercises, dailyExercises)
56 }
57 for(let k = 1; k <= musclesPerDay; k++){ // for every muscle per day
58 WorkoutForOneDayGeneration(dailyExercises, k, daysPerWeek)
59 }
60 workout.push({ day: i, exercises: dailyExercises })
61 }
62 return workout;
63 } else if(daysPerWeek == 4){
64 let workout = []
65 for(let i = 1; i <= daysPerWeek; i++){

```

```

66 let dailyExercises = []; // Array of daily exercises
67 GetCardioExercise(i, dailyExercises)
68 if(i % 2 === 0){
69   for(let k = 1; k <= 2; k++){ // for every muscle per day
70     WorkoutForOneDayGeneration(dailyExercises, k, 3) // Two muscles per
       day every two days 1 2 1 2
71   }
72 } else {
73   GetAbsExercise(absExercises, dailyExercises)
74   for(let k = 1; k <= 1; k++){ // How many muscles per day
75     WorkoutForOneDayGeneration(dailyExercises, k, 6) // One muscle per day
       every two days 1 2 1 2
76   }
77 }
78 workout.push({ day: i, exercises: dailyExercises })
79 }
80 return workout;
81 } else if(daysPerWeek == 5){
82   let workout = []
83   for(let i = 1; i <= daysPerWeek; i++){
84     let dailyExercises = [];
85     GetCardioExercise(i, dailyExercises)
86     if(i === 1){
87       for(let k = 1; k <= 2; k++){
88         WorkoutForOneDayGeneration(dailyExercises, k, 3) // in first day of
           workout - 2 muscles with 3 exercises per muscle
89       }
90     } else {
91       GetAbsExercise(absExercises, dailyExercises)
92       for(let k = 1; k <= 1; k++){
93         WorkoutForOneDayGeneration(dailyExercises, k, 6) // other days one
           muscle per day with 6 exercises per muscle
94       }
95     }
96     workout.push({ day: i, exercises: dailyExercises })
97   }
98   return workout;
99 }
100
101 function HowManyMusclesPerDay(){
102   if(daysPerWeek == 2){
103     musclesPerDay = muscleNames.length / 2;
104   } else if(daysPerWeek == 3){
105     musclesPerDay = muscleNames.length / 3;
106   } else if(daysPerWeek == 6){
107     musclesPerDay = muscleNames.length / 6;
108   }
109 }
110
111 function WorkoutForOneDayGeneration(dailyExercises, k, daysPerWeek){
112   const tempExercise = muscleNames[Math.floor(Math.random() *
     muscleNames.length)] // Random muscle selection
113   muscleNames = muscleNames.filter(x => x !== tempExercise); // Remove
     that random value from array, to prevent duplication
114   if(k == 1) { // if first exercise of the day, choose warm up exercises

```

```

115 GetWarmUpExercise(stretchingExercises, dailyExercises);
116 }
117 HowManyExercisesPerMucle(tempExercise, dailyExercises, daysPerWeek) //
    get numbers for compound and isolation exercises
118 }
119
120 function HowManyExercisesPerMucle(tempExercise, dailyExercises,
    daysPerWeek){
121     let isolationNumber = null;
122     let compoundNumber = null;
123     if(daysPerWeek == 6){ // it means 1 muscle per day
124         if(gender === 'male'){
125             if(level === 'beginner'){
126                 if(goal === 'lose fat'){
127                     isolationNumber = 5;
128                     compoundNumber = 1;
129                 } else if(goal === 'build muscle'){
130                     isolationNumber = 2;
131                     compoundNumber = 4;
132                 } else if(goal === 'get stronger'){
133                     isolationNumber = 3;
134                     compoundNumber = 3;
135                 }
136             } else if(level === 'intermediate'){
137                 if(goal === 'lose fat'){
138                     isolationNumber = 5;
139                     compoundNumber = 1;
140                 } else if(goal === 'build muscle'){
141                     isolationNumber = 2;
142                     compoundNumber = 4;
143                 } else if(goal === 'get stronger'){
144                     isolationNumber = 3;
145                     compoundNumber = 3;
146                 }
147             }
148         } else if (gender === 'female'){
149             if(level === 'beginner'){
150                 if(goal === 'lose fat'){
151                     isolationNumber = 6;
152                     compoundNumber = 0;
153                 } else if(goal === 'build muscle'){
154                     isolationNumber = 4;
155                     compoundNumber = 2;
156                 } else if(goal === 'get stronger'){
157                     isolationNumber = 1;
158                     compoundNumber = 5;
159                 }
160             } else if(level === 'intermediate'){
161                 if(goal === 'lose fat'){
162                     isolationNumber = 5;
163                     compoundNumber = 1;
164                 } else if(goal === 'build muscle'){
165                     isolationNumber = 4;
166                     compoundNumber = 2;
167                 } else if(goal === 'get stronger'){

```

```

168     isolationNumber = 5;
169     compoundNumber = 1;
170 }
171 }
172 }
173 } else if (daysPerWeek == 2){ // 3 muscles per day
174     if(gender === 'male'){
175         if(level === 'beginner'){
176             if(goal === 'lose fat'){
177                 isolationNumber = 2;
178                 compoundNumber = 0;
179             } else if(goal === 'build muscle'){
180                 isolationNumber = 1;
181                 compoundNumber = 1;
182             } else if(goal === 'get stronger'){
183                 isolationNumber = 1;
184                 compoundNumber = 1;
185             }
186         } else if(level === 'intermediate'){
187             if(goal === 'lose fat'){
188                 isolationNumber = 2;
189                 compoundNumber = 1;
190             } else if(goal === 'build muscle'){
191                 isolationNumber = 1;
192                 compoundNumber = 3;
193             } else if(goal === 'get stronger'){
194                 isolationNumber = 2;
195                 compoundNumber = 2;
196             }
197         }
198     } else if (gender === 'female'){
199         if(level === 'beginner'){
200             if(goal === 'lose fat'){
201                 isolationNumber = 2;
202                 compoundNumber = 0;
203             } else if(goal === 'build muscle'){
204                 isolationNumber = 1;
205                 compoundNumber = 1;
206             } else if(goal === 'get stronger'){
207                 isolationNumber = 1;
208                 compoundNumber = 1;
209             }
210         } else if(level === 'intermediate'){
211             if(goal === 'lose fat'){
212                 isolationNumber = 3;
213                 compoundNumber = 0;
214             } else if(goal === 'build muscle'){
215                 isolationNumber = 2;
216                 compoundNumber = 2;
217             } else if(goal === 'get stronger'){
218                 isolationNumber = 2;
219                 compoundNumber = 1;
220             }
221         }
222     }

```

```

223 } else if (daysPerWeek == 3){ // 2 muscles per day
224 if(gender === 'male'){
225 if(level === 'beginner'){
226 if(goal === 'lose fat'){
227 isolationNumber = 3;
228 compoundNumber = 0;
229 } else if(goal === 'build muscle'){
230 isolationNumber = 1;
231 compoundNumber = 2;
232 } else if(goal === 'get stronger'){
233 isolationNumber = 2;
234 compoundNumber = 1;
235 }
236 } else if(level === 'intermediate'){
237 if(goal === 'lose fat'){
238 isolationNumber = 4;
239 compoundNumber = 1;
240 } else if(goal === 'build muscle'){
241 isolationNumber = 2;
242 compoundNumber = 3;
243 } else if(goal === 'get stronger'){
244 isolationNumber = 3;
245 compoundNumber = 2;
246 }
247 }
248 } else if (gender === 'female'){
249 if(level === 'beginner'){
250 if(goal === 'lose fat'){
251 isolationNumber = 3;
252 compoundNumber = 0;
253 } else if(goal === 'build muscle'){
254 isolationNumber = 1;
255 compoundNumber = 2;
256 } else if(goal === 'get stronger'){
257 isolationNumber = 2;
258 compoundNumber = 1;
259 }
260 } else if(level === 'intermediate'){
261 if(goal === 'lose fat'){
262 isolationNumber = 4;
263 compoundNumber = 0;
264 } else if(goal === 'build muscle'){
265 isolationNumber = 2;
266 compoundNumber = 3;
267 } else if(goal === 'get stronger'){
268 isolationNumber = 3;
269 compoundNumber = 2;
270 }
271 }
272 }
273 }
274 for(let j = 1; j <= isolationNumber; j++){ // get exercises, sets,
    reps, and rest for isolation exercise
275 if(level === 'beginner'){

```

```

276 GetIsolationExercise(tempExercise, isolationExercisesForBeginner,
    dailyExercises);
277 } else {
278 GetIsolationExercise(tempExercise, isolationExercises, dailyExercises)
    ;
279 }
280 }
281 for(let l = 1; l <= compoundNumber; l++){ // get exercises, sets, reps
    , and rest for compound exercise
282 if(level === 'beginner'){
283 GetCompoundExercise(tempExercise, compoundExercisesForBeginner,
    dailyExercises);
284 } else {
285 GetCompoundExercise(tempExercise, compoundExercises, dailyExercises);
286 }
287 }
288 }
289
290 function GetWarmUpExercise(stretchingExercises, dailyExercises){
291 for(let i = 1; i <= 3; i++){
292 let stretching = stretchingExercises[Math.floor(Math.random() *
    stretchingExercises.length)] // random stretching exercises
293 dailyExercises.push({ exercise: stretching })
294 }
295 }
296
297 function GetAbsExercise(absExercises, dailyExercises){
298 for(let i = 1; i <= 2; i++){
299 let abs = absExercises[Math.floor(Math.random() * absExercises.length)
    ] // random abs exercises
300 dailyExercises.push({ exercise: abs })
301 }
302 }
303
304 function GetCardioExercise(thisDay, dailyExercises){
305 let min = null;
306 let cardio = null;
307 if(level === 'beginner'){
308 if(goal === 'lose fat'){ // if user is beginner and goal is lose fat,
    cardio exercises will be every day
309 min = 35;
310 cardio = cardioExercises[Math.floor(Math.random() * cardioExercises.
    length)]
311 dailyExercises.push({ min, exercise: cardio })
312 } else if(goal === 'build muscle'){
313 min = 15;
314 BuildMuscleCardion(thisDay)
315 } else if(goal === 'get stronger'){
316 min = 20;
317 GetStrongerCardio(thisDay)
318 }
319 } else if(level === 'intermediate'){
320 if(goal === 'lose fat'){
321 min = 45;

```

```

322   cardio = cardioExercises[Math.floor(Math.random() * cardioExercises.
      length)]
323   dailyExercises.push({ min, exercise: cardio })
324   } else if(goal === 'build muscle'){
325     min = 20;
326     BuildMuscleCardion(thisDay)
327   } else if(goal === 'get stronger'){
328     min = 30;
329     GetStrongerCardio(thisDay)
330   }
331   }
332
333   function BuildMuscleCardion(day){ // Push cardio exercises by goal
334   if(day % 3 > 1){
335     let cardio = cardioExercises[Math.floor(Math.random() *
      cardioExercises.length)]
336     dailyExercises.push({ min, exercise: cardio })
337   }
338   }
339
340   function GetStrongerCardio(day){ // Push cardio exercises by goal
341   if(day % 2){
342     let cardio = cardioExercises[Math.floor(Math.random() *
      cardioExercises.length)]
343     dailyExercises.push({ min, exercise: cardio })
344   }
345   }
346   }
347   }
348
349   function GetIsolationExercise(tempExercise, isolationExercises,
      dailyExercises){
350     let isolationExercise = null;
351     if(tempExercise === 'legs' && gender === 'female'){
352       isolationExercise = isolationExercises['legsForWomen'][Math.floor(Math
        .random() * isolationExercises['legsForWomen'].length)]
353     } else {
354       isolationExercise = isolationExercises[tempExercise][Math.floor(Math.
        random() * isolationExercises[tempExercise].length)]
355     }
356     if(gender === 'male'){
357       if(level === 'beginner'){
358         if(goal === 'lose fat'){
359           sets = 3;
360           reps = 12;
361           rest = 45;
362         } else if(goal === 'build muscle'){
363           sets = 3;
364           reps = 8;
365           rest = 120;
366         } else if(goal === 'get stronger'){
367           sets = 3;
368           reps = 10;
369           rest = 90;
370         }

```

```

371 } else if(level === 'intermediate'){
372   if(goal === 'lose fat'){
373     sets = 3;
374     reps = 15;
375     rest = 45;
376   } else if(goal === 'build muscle'){
377     sets = 4;
378     reps = 8;
379     rest = 120;
380   } else if(goal === 'get stronger'){
381     sets = 3;
382     reps = 10;
383     rest = 90;
384   }
385 }
386 } else if(gender === 'female'){
387   if(level === 'beginner'){
388     if(goal === 'lose fat'){
389       sets = 3;
390       reps = 12;
391       rest = 45;
392     } else if(goal === 'build muscle'){
393       sets = 3;
394       reps = 8;
395       rest = 120;
396     } else if(goal === 'get stronger'){
397       sets = 3;
398       reps = 10;
399       rest = 90;
400     }
401   } else if(level === 'intermediate'){
402     if(goal === 'lose fat'){
403       sets = 3;
404       reps = 15;
405       rest = 45;
406     } else if(goal === 'build muscle'){
407       sets = 3;
408       reps = 10;
409       rest = 120;
410     } else if(goal === 'get stronger'){
411       sets = 3;
412       reps = 12;
413       rest = 90;
414     }
415   }
416 }
417
418 return dailyExercises.push({ sets, reps, rest, exercise:
    isolationExercise })
419 }
420
421 function GetCompoundExercise(tempExercise, compoundExercises,
    dailyExercises){
422   let compound = null;
423   if(tempExercise === 'biceps'){

```

```

424 compound = isolationExercises['biceps'][Math.floor(Math.random() *
    isolationExercises['biceps'].length)]
425 } else {
426 compound = compoundExercises[tempExercise][Math.floor(Math.random() *
    compoundExercises[tempExercise].length)]
427 }
428 if(gender === 'male'){
429 if(level === 'beginner'){
430 if(goal === 'lose fat'){
431 sets = 3;
432 reps = 10;
433 rest = 60;
434 } else if(goal === 'build muscle'){
435 sets = 4;
436 reps = 6;
437 rest = 120;
438 } else if(goal === 'get stronger'){
439 sets = 3;
440 reps = 8;
441 rest = 90;
442 }
443 } else if(level === 'intermediate'){
444 if(goal === 'lose fat'){
445 sets = 3;
446 reps = 10;
447 rest = 45;
448 } else if(goal === 'build muscle'){
449 sets = 5;
450 reps = 5;
451 rest = 120;
452 } else if(goal === 'get stronger'){
453 sets = 4;
454 reps = 6;
455 rest = 90;
456 }
457 }
458 } else if(gender === 'female'){
459 if(level === 'beginner'){
460 if(goal === 'lose fat'){
461 sets = 3;
462 reps = 12;
463 rest = 45;
464 } else if(goal === 'build muscle'){
465 sets = 4;
466 reps = 8;
467 rest = 120;
468 } else if(goal === 'get stronger'){
469 sets = 3;
470 reps = 10;
471 rest = 90;
472 }
473 } else if(level === 'intermediate'){
474 if(goal === 'lose fat'){
475 sets = 3;
476 reps = 10;

```

```

477   rest = 45;
478   } else if(goal === 'build muscle'){
479     sets = 4;
480     reps = 8;
481     rest = 120;
482   } else if(goal === 'get stronger'){
483     sets = 3;
484     reps = 8;
485     rest = 90;
486   }
487   }
488   }
489   return dailyExercises.push({ sets, reps, rest, exercise: compound })
490   }
491   }
492
493   module.exports = {GeneratePlan};

```

1 išėities kodas. Treniruočių programos generavimo algoritmas

B. Treniruočių programos generavimo forma

The screenshot shows the 'Plan' screen of a 'Fitness App'. The top navigation bar includes 'Profile', 'Coaches', 'Exercises', 'Workouts', 'Workout Plan', and a 'Logout' button. The main content area has a dark background with white text and form elements. It starts with the title 'Plan' and the label 'Workout Level'. Below this is a dropdown menu showing 'Beginner'. The next question is 'How many days per week you can workout?' with a dropdown menu showing '2'. The third question is 'Workout Goal' with a dropdown menu showing 'Lose fat'. A blue 'Submit' button is below the goal selection. Below the button, it says 'Your heart ratio during cardio should be around' followed by a large red heart icon, the number '129', and another red heart icon. At the bottom, there is a dark grey bar with the text 'Workout Plan'.

14 lentelė. Treniruočių programos generavimo forma

C. Treniruočių programos atvaizdavimas

Workout Plan			
1	Minutes	Sets x Reps	Rest
Kneeling hip-flexor (groin)(stretching)			
Seated spinal twist (glutes)(stretching)			
Ground quadricep stretch(stretching)			
Reverse Flyes Machine(shoulders)(isolation)		4 x 8	120sec.
Reverse Flyes(shoulders)(isolation)		4 x 8	120sec.
Standing Dumbbell Press(shoulders)(compound)		5 x 5	120sec.
Seated Dumbbell Shoulder Press(shoulders)(compound)		5 x 5	120sec.
Seated Barbell Military Press(shoulders)(compound)		5 x 5	120sec.
Lying Leg Curls(legs)(isolation)		4 x 8	120sec.
Leg Extensions(legs)(isolation)		4 x 8	120sec.
Box Squat(legs)(compound)		5 x 5	120sec.
Romanian Deadlift(legs)(compound)		5 x 5	120sec.
Clean from Blocks(legs)(compound)		5 x 5	120sec.

15 lentelė. Treniruočių programos 1 dalis

2	Minutes	Sets x Reps	Rest
Jump Rope(cardio)	20		
Hanging Pike(abs)			
Decline Crunch(abs)			
Calf raises(stretching)			
Seated spinal twist (glutes)(stretching)			
hamstring stretch(stretching)			
Incline Cable Chest Press(chest)(isolation)		4 x 8	120sec.
Butterfly Machine(chest)(isolation)		4 x 8	120sec.
Decline Barbell Bench Press(chest)(compound)		5 x 5	120sec.
Bench Press(chest)(compound)		5 x 5	120sec.
Pushups(chest)(compound)		5 x 5	120sec.
EZ-Bar Skullcrusher(triceps)(isolation)		4 x 8	120sec.
Tricep Kickback(triceps)(isolation)		4 x 8	120sec.
Bench Dip(triceps)(compound)		5 x 5	120sec.
Weighted Bench Dip(triceps)(compound)		5 x 5	120sec.
Bench Dip(triceps)(compound)		5 x 5	120sec.

16 lentelė. Treniruočių programos 2 dalis

3	Minutes	Sets x Reps	Rest
Chest Opener(stretching)			
Kneeling hip-flexor (groin)(stretching)			
Kneeling hip-flexor (groin)(stretching)			
Preacher Curls(biceps)(isolation)		4 x 8	120sec.
EZ-Bar Curl(biceps)(isolation)		4 x 8	120sec.
Hammer Curls(biceps)(isolation)		5 x 5	120sec.
Incline Hammer Curls(biceps)(isolation)		5 x 5	120sec.
Cable Curls(biceps)(isolation)		5 x 5	120sec.
Seated One-arm Cable Pulley Rows(middle back)(isolation)		4 x 8	120sec.
One-Arm Dumbbell Row(middle back)(isolation)		4 x 8	120sec.
Rowing, Stationary(middle back)(compound)(cardio)		5 x 5	120sec.
One-Arm Long Bar Row(middle back)(compound)		5 x 5	120sec.
Pullups(lats)(compound)		5 x 5	120sec.

17 lentelė. Treniruočių programos 3 dalis