

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Reio Sedrik 155429IAPB

MOBIILIRAKENDUS VIRTUAALSE JÕUSAALI PERSONAALTREENERINA

Bakalaureusetöö

Juhendaja: Priit Järv
PhD

Tallinn 2020

Autorideklaratsioon

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Reio Sedrik

18.05.2020

Annotatsioon

Käesoleva bakalaureusetöö käigus loodi mobiilirakendus, mis käitub kasutaja personaaltreenerina, pakkudes talle treeningplaane ning andes soovitusi harjutuste sooritamiseks. Samuti on kasutajal võimalus ise treeningplaane koostada ning jälgida oma arengut. Lõputöö raames valmis antud rakendus kasutades Flutter raamistikku.

Üheks peamiseks eesmärgiks oli luua parema kasutusmugavusega rakendus võrreldes teiste olemasolevate rakendustega samas kategoorias. Lisaks rakenduse loomisele võrreldakse rakendust sarnaste olemasolevate rakendustega ning analüüsitakse kasutusmugavust ja funktsionaalsust.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 29 leheküljel, 5 peatükki, 20 joonist, 2 tabelit.

Abstract

Mobile application as a personal gym trainer

During this thesis was created a mobile application that works as a personal gym trainer for the user, provides workout plans and gives recommendations for performing exercises. Also the user has the option to create workout plans and keep track of the workout progress. The application was made using the Flutter framework.

The main purpose of this thesis was to create an application with better user experience than other similar applications in the same category. In addition to creating the application, it is compared to other similar applications and the user experience and functionality is analyzed.

The thesis is in Estonian and contains 29 pages of text, 5 chapters, 20 figures, 2 tables.

Lühendite ja mõistete sõnastik

Android	Google poolt loodud operatsioonisüsteem mobiilsetele seadmetele
Set	Treeningu käigus järjestikult ilma puhkepausita sooritatud kordused
Kordus	Ühekordne harjutuse sooritus
Äpp	Mobiilirakendus
Algoritm	Täpne eeskirjade kogum kindlate ülesannete lahendamiseks
Logima	Treeningandmeid salvestama
Navigatsioon	Liikumine rakenduse ühest osast teise
UI	Visuaalne pool rakendusest, mille abil kasutaja suhtleb arvutiga
Žest	(ingl. k gesture) Käeliigutus teatud tegevuse sooritamiseks rakenduses
Element	Üks osa rakendusest, näiteks nupp või tekstiväli
Debugimine	Rakenduse jooksutamise vigade ja probleemide leidmiseks
ARM	Mikroprotsessor, mida kasutavad enamik tänapäeva nutitelefonid
Kompilaator	Arvutiprogramm, mis teisendab käsud madala taseme keelde, et need oleks arvuti poolt loetavad
Protokoll	Ametlik eeskiri, kuidas süsteem peab käituma
Autoriseerima	Heaks kiitma, andma luba millegi teostamiseks
Laiendus	Lisaosa, et sooritada täiendavat funktsiooni
Emulaator	Tarkvara, mis laseb arvutisüsteemil käituda teise arvutisüsteemina
Google Play	Koht rakenduste leidmiseks ning allalaadimiseks, mis on mõeldud Android seadmetele
App Store	Sarnane süsteem Google Play-le, mis on mõeldud iOS seadmetele
Domeen	Üks osa või valdkond internetist, mis kuulub kindlasse kategooriasse või mida omab teatud organisatsioon või individuaal
Lambda süntaks	Anonüümne funktsioon, võimaldab kirjutada lühemat ning hästi loetavat koodi

Sisukord

1 Sissejuhatus	9
2 Taust.....	11
2.1 Jõusaalis kasutatavad rakendused	11
2.2 Allikad kasutajaliideste disaini kohta.....	13
3 Meetodid.....	15
3.1 Funktsionaalsed nõuded	15
3.2 Kasutusmugavus	20
3.2.1 Olemasolevate rakenduste analüüs.....	21
3.2.2 Funktsionaalsuse võrdlus teiste rakendustega.....	23
3.3.4 Kasutusmugavuse saavutamine.....	25
3.3 Kasutatud tehnoloogiad	26
3.3.1 Flutter.....	27
3.3.2 Firebase	29
3.3.3 Cloud Firestore.....	29
3.3.4 Firebase Authentication	29
4 Tulemused	30
5 Kokkuvõte	32

Jooniste loetelu

Joonis 1. FitnessAI.....	12
Joonis 2. My Workout Plan.....	12
Joonis 3. Avavaade	15
Joonis 4. Põhivaade.....	15
Joonis 5. Aktiivse treeningu <i>set</i> -i vaade.....	17
Joonis 6. Aktiivse treeningu harjutuse vaade	17
Joonis 7. Kalendri vaade	18
Joonis 8. Treeningplaanide vaade.....	18
Joonis 9. Treeningplaani muutmise vaade	19
Joonis 10. Seadete vaade.....	19
Joonis 11. Profiili vaade.....	20
Joonis 12. Dart koodinäide harjutuse jaoks maksimaalse raskuse leidmiseks	28
Joonis 13. Flutter koodinäide Dartis põhivaate profiilinupust kastuades olemasolevaid UI komponente	28
Joonis 14. Struktuur andmebaasi treeningutest	35
Joonis 15. Andmebaasi struktuur treeningplaanidest.....	36
Joonis 16. Andmebaasi struktuur kasutajate profiilidest	36
Joonis 17. Andmebaasi struktuur autentitud kasutajatest	37
Joonis 18. Andmebaasi struktuur harjutustest	37
Joonis 19. Koodinäide eelseisva treeningu pärimiseks Firestore'ist.....	38
Joonis 20. Koodinäide treeningseisiooni salvestamisest Firestor	39

Tabelite loetelu

Tabel 1. Loodud rakenduse, FitnessAI ja My Workout Plan võrdlus	23
Tabel 2. Tähtsamate funktsionaalsuste kasutusmugavuse võrdlus teiste rakendustega .	31

1 Sissejuhatus

Käesoleva lõputöö raames valmis mobiilirakendus, mis käitub kasutaja personaaltreenerina. Rakendus võimaldab kasutajal koostada treeningplaani või kasutada rakenduse poolt pakutud treeningplaane. Kasutaja saab salvestada oma treeninguid ning jälgida arengut. Lisaks rakendus koostab peale igat treeningut kasutajale soovitusi järgmiseks trenniks.

Peamiseks eesmärgiks oli luua parema kasutusmugavusega rakendus võrreldes konkurentidega. Selle põhjuseks on see, et mitmeid sarnaseid rakendusi juba eksisteerib, kuid autor leidis, et need ei ole kasutusmugavuse poolest kõige paremad ning esineb mõningaid funktsionaalseid puudusi.

Rakendus koosneb kahest osast - mobiilirakendus koos pilveteenusega ning masinõppe osa, mis analüüsib treeninguid ning pakub kasutajale tema jaoks optimeeritud harjutusi. Antud töö raames valmis mobiilirakendus koos andmebaasiga. Masinõppe poolega tegeleb kaastudeng Rasmus Sikk oma lõputöö raames. Lõputöö raames koostatakse rakenduse esialgne versioon, tulevikus on plaanis seda edasi arendada ning ka turule minna.

Rakendus sai valitud kuna sport ja tervislik eluviis on pikalt populaarsust kogunud ning inimesed otsivad pidevalt viise, kuidas ennast rohkem motiveerida ja treeninguid lihtsustada või mugavamaks muuta. Kuna treenimine ise on küllaltki raske, siis kasutajad tahavad, et mobiilirakendus, mida nad kasutavad, teeks treenimise lihtsamaks, mitte ei teeks juurde lisatööd, et oma arengut jälgida. Sarnaseid rakendusi on turul juba palju, kuid esineb mõningaid puudusi. Neid kirjeldatakse töö käigus.

Teises peatükis antakse ülevaade taustast ning sarnastest rakendustest ja samuti kirjeldatakse disainiprintsiipe, mille järgi kasutusmugavust kujundatakse.

Kolmandas peatükis võrreldakse antud rakendust olemasolevate rakendustega, tuuakse välja erinevused ning sarnasused, analüüsitakse kasutusmugavust ning kirjeldatakse rakenduse funktsionaalsust.

Neljandas peatükis kirjeldatakse tööga saavutatud tulemusi ning viiendas peatükis antakse kokkuvõtte koostatud tööle.

2 Taust

Tervislik eluviis ning sportimine on muutunud viimastel aastatel üha populaarsemaks. Järjest rohkem inimesi saab aru kui suurt rolli tervis nende elus mängib. Et lihtsustada inimestel tervise eest hoolitsemist, on loodud mitmeid rakendusi eri valdkondades, rakendused, mis pakuvad toitumiskavasid ning retsepte, mediteerimise rakendused ning treeningkavasid pakuvad ja treeninguid abistavad rakendused. Antud töö raames loodav rakendus keskendub jõusaalitreeningutele. Olemasolevaid rakendusi, mis abistavad jõusaalitreeninguid, on juba mitmeid.

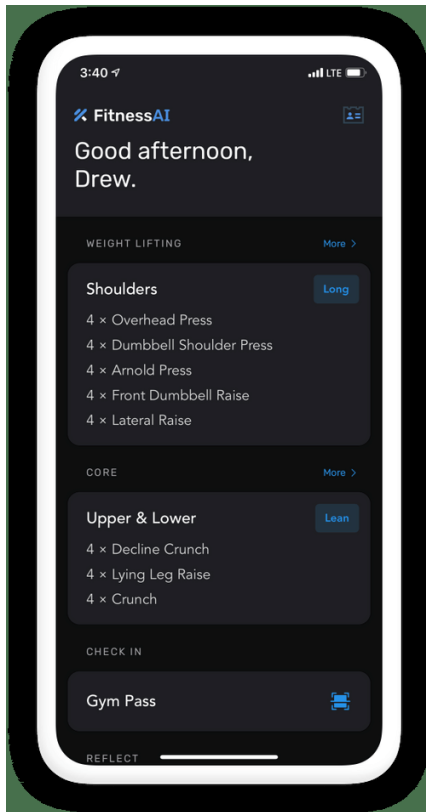
Idee antud rakenduse loomiseks tekkiski muid sarnaseid rakendusi kasutades. Treeninguid juhendavaid ja abistavaid rakendusi on palju, kuid neil esineb mõningaid puudusi või ei vasta täpselt kasutaja vajadustele. Lisaks paljud rakendused üritavad katta hästi laia kasutusala, hõlmates kõikvõimalikke treeningvaldkondi, toitumist ja muud treenimisega seonduvat. Selle tõttu tekkis mõte luua rakendus, mis oleks kasutusmugavuse poolest parem teistest sarnastest rakendustest ning hõlmaks ainult jõusaalitreeninguid.

2.1 Jõusaalis kasutatavad rakendused

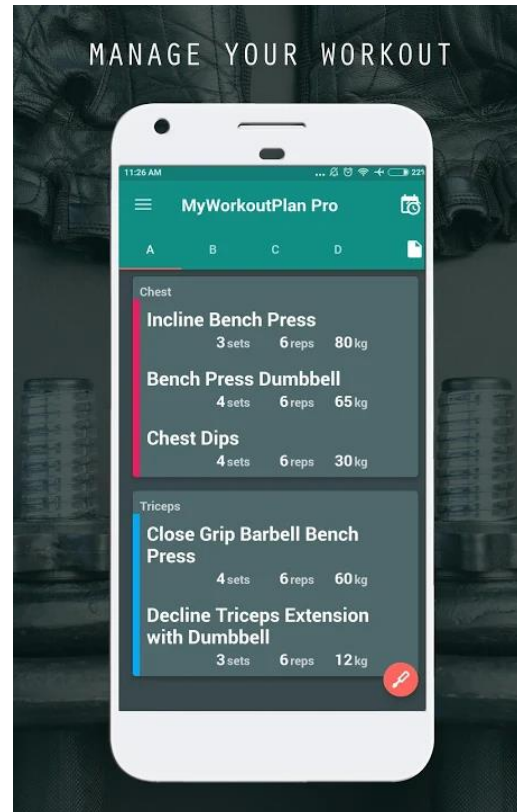
On mitmeid olemasolevaid mobiilirakendusi, mida jõusaalis kasutatakse treeningute abistamiseks. On lihtsamaid rakendusi ainult treeningute logimiseks[16], et kasutajatel oleks ülevaade oma treeningutest, kuid on ka suurema funktsionaalsusega rakendusi, mis analüüsivad kasutaja treeninguid, andes tagasiside või pakkudes soovitusi treeningute osas[1]. On rakendusi, mis pakuvad inimestele treeningkavasid, mis on kasutajate[2] või treenerite enda loodud või loodud kasutades tehnoloogiaid nagu masinõpe ja analüüsitud suurt hulka andmeid treeningsessioonidest[1].

Käesolevas töös võrreldakse osasid neist käesoleva rakendusega. Antud peatükis tuuakse lühitutvustused võrreldavatest rakendustest ning järgmises peatükis

analüüsitakse rakendusi põhjalikumalt ja võrreldakse antud rakendusega. Autor valis võrreldavad rakendused kuna need olid Google Play Store'is või App Store'is ühed populaarseimad või parima hinnanguga.



Joonis 1. FitnessAI



Joonis 2. My Workout Plan

FitnessAI

FitnessAI (Joonis 1) on rakendus, mis pakub kasutajale treeningkavasid ning kasutab masinõppe algoritmi, et pakkuda kasutajale raskusi ja kordusi kõige optimaalsemaks arenguks. FitnessAI sai alguse sellest, et rakenduse looja oli varem loonud äpi, mis võimaldas treeninguid logida[5]. Mõne aasta jooksul oli rakendus kogunud umbes 6 miljoni treeningu andmed. Autor hakkas andmeid analüüsima, et leida kõige optimaalsem viis arenguks, ja koostas algoritmi, mida kasutab loodud iPhone-i rakenduses.

My Workout Plan

My Workout Plan (Joonis 2) on rakendus, mis võimaldab kasutajal luua treeningplaane või järgida teiste kasutajate loodud treeningplaane. Rakendus pakub palju võimalusi plaanide ja treeningute kohandamiseks, kuid see nõuab kasutajalt lisa vaeva ning aega, kuna treeningplaanid on vaja ise koostada. Kõik harjutused tuleb üksikhaaval valida ning ka raskused ja kordused paika panna. Samuti ei paku rakendus võimalust oma treeninguid ette plaanida. Lähemalt analüüsitakse My Workout Plan rakendust ning eelnevas peatükis mainitud rakendust FitnessAI peatükis 3.3.1.

2.2 Allikad kasutajaliideste disaini kohta

Peamiseks allikaks, mida järgiti antud rakenduse kasutajaliidese kasutusmugavuse disainimisel, oli Material Design [3]. Material Design on visuaalne keel, mis ühendab hea disaini klassikalised põhimõtted tehnoloogia innovatsiooniga.

Kuna kasutajatel on erineval tasemel domeeniteadmised, teadmised tehnoloogiast ja samuti füüsilised võimed, näiteks võivad kasutajatel esineda kehvad kognitiivsed oskused, vigastused või halb nägemine, siis peab jälgima, et rakendus oleks kasutatav kõigile.

Peamised Material Design printsiibid kasutusmugavuse osas

- Elementide paigutus peab olema selge koos eristuvate *call to action*-itega (üleskutse tegevusele)
- Rakendus peab toetama mitmesuguseid kasutajaid
- Abistavate tehnoloogiate kasutamine (näiteks ekraanilugejad) - seda antud töö raames ei jälgita
- Hierarhia kasutamine navigatsiooni lihtsustamiseks
 - Selgelt nähtavad elemendid
 - Piisav kontrast ja suurus
 - Elemendid on eristatavad vastavalt tähtsusele
 - Põhiinformatsioon on lihtsasti kättesaadav ja leitav
 - Elemendid on loogilises järjekorras

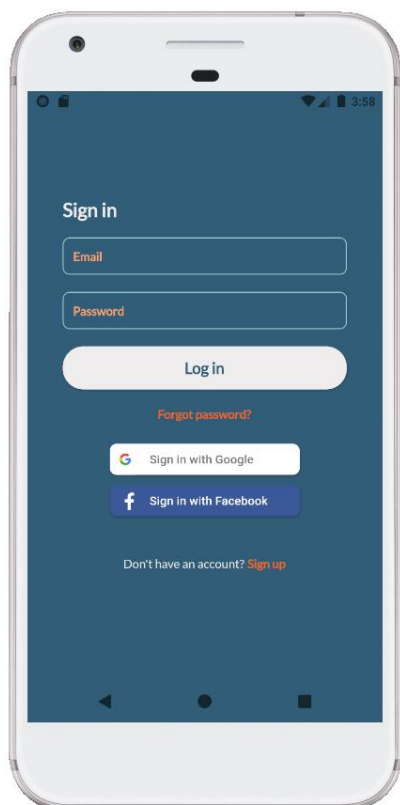
- *Touch targets* (elemendid, mis on vajutatavad) peaks olema piisavalt suure alaga, et toetada inimesi, kellel on raskusi väikeste elementide vajutamisega
- Paindlik arendus - elementide paigutus on reageeriv erinevatele ekraanisuurustele, et vältida informatsiooni kadu väiksemate ekraanidega
- Seotud elemendid on grupeeritud kokku

3 Meetodid

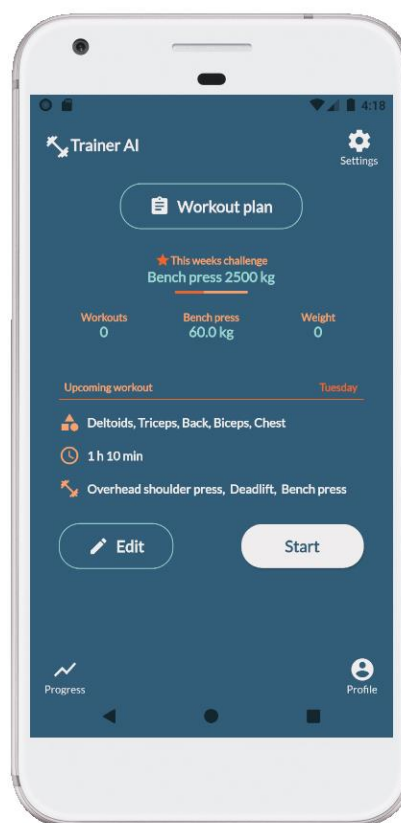
3.1 Funktsionaalsed nõuded

Avavaade (Joonis 3)

1. Kasutaja saab luua kasutaja, millega rakendust kasutada.
 - a. Kasutaja loomiseks saab kasutada emaili, Google'i või Facebook'i kontot
 - b. Sisselogimise vaate kaudu on võimalik parooli muuta. Sisestades email, saadetakse kasutajale emailile link parooli muutmiseks.
2. Kasutajat luues saab kasutaja luua profiili - nimi, vanus, kehakaal, sugu, treeningeesmärgid.



Joonis 3. Avavaade



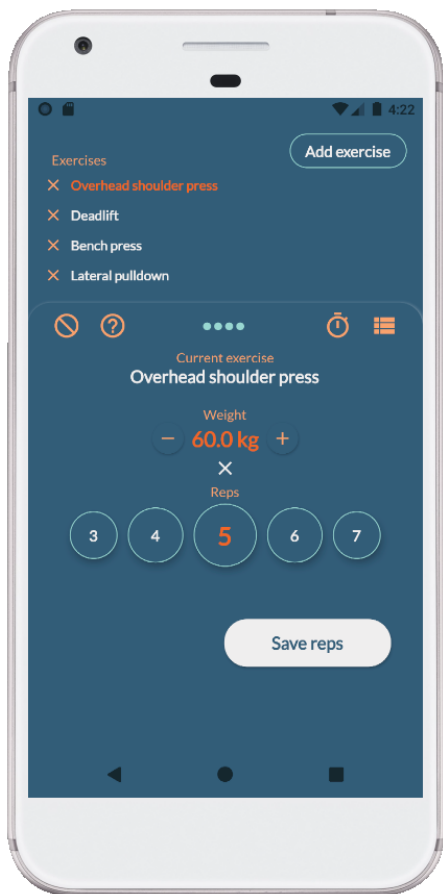
Joonis 4. Põhivaade

Põhivaade (Joonis 4)

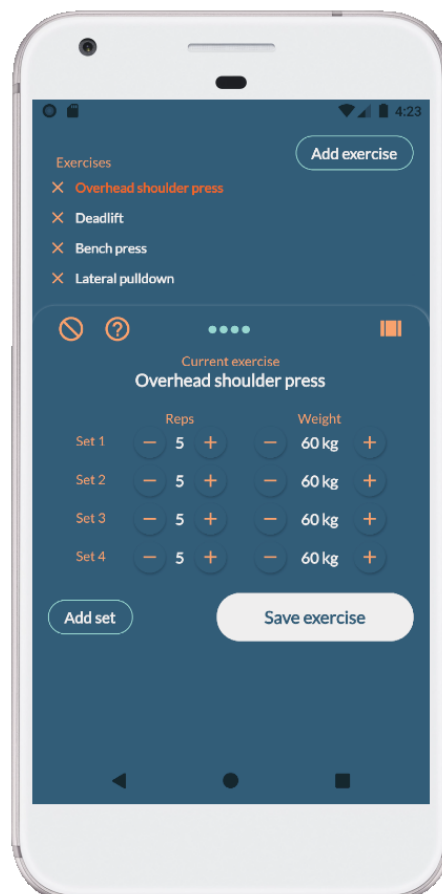
1. Põhivaates on kasutajal ülevaade eelseisvast treeningust.
 - a. Kasutaja näeb eesseisvaid harjutusi ning lihasgruppe, mida antud harjutused hõlmavad. Lisaks on nähtaval päev, mis ajaks on harjutus planeeritud ja samuti harjutuse kestus.
 - b. Kasutaja saab alustada treeningut või muuta treeningu harjutusi
2. Kasutaja saab navigeerida treeningplaani ja kalendri, seadete, profiili ning progressi vaadetesse.

Aktiivse treeningu vaade (Joonis 5 ja 6)

1. Kasutaja saab valida hetke *set*-i vaate ja hetke harjutuse vaate vahel.
 - a. *Set*-i vaates saab kasutaja muuta ja sisestada sooritatud *set*-i raskust ning korduste arvu
 - b. Harjuse vaates saab kasutaja muuta kogu harjutuse *set*-ide raskusi ning kordusi, samuti saab *set*-e juurde lisada
2. Kasutajal on ülevaade kõikidest harjutustest, mida treening sisaldab
3. Kasutaja näeb, mis harjutus on aktiivne ning mis harjutused on lõpetatud.
4. Kasutaja saab treeningust harjutusi eemaldada ja lisada.
5. Harjutuse lisamiseks avaneb otsinguväljaga dialoog, kust saab valida harjuse, mida lisada treeningule
6. Kasutaja saab aktiivse harjutuse vahele jätta
7. Kasutaja saab valida, kas näidata taimerit *set*-ide vahel
8. Harjutuse lõppedes saab kasutaja anda tagasisidet, kuidas talle treening meeldis



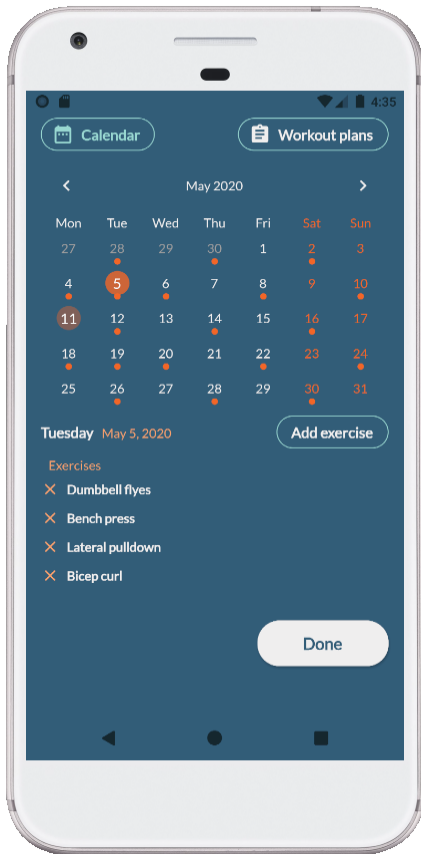
Joonis 5. Aktiivse treeningu *set*-i vaade



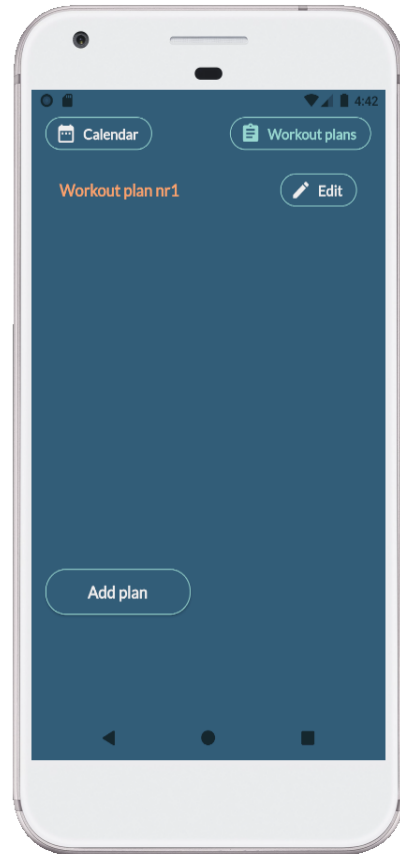
Joonis 6. Aktiivse treeningu harjutuse vaade

Kalendri vaade (Joonis 7)

1. Kasutaja saab kalendrist valida kuid ning päevi, mida vaadata
2. Kasutaja näeb, milliste päevade peal on treeningud ja milliseid harjutusi treeningud sisaldavad
3. Kasutajal on võimalik päeva treeningule harjutusi lisada või harjutusi eemaldada



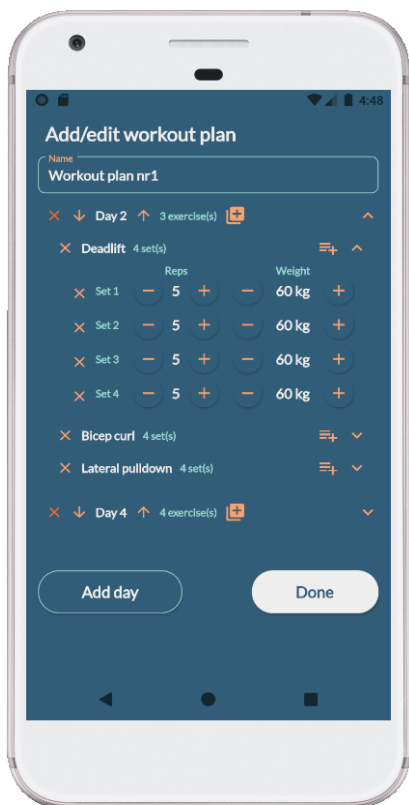
Joonis 7. Kalendri vaade



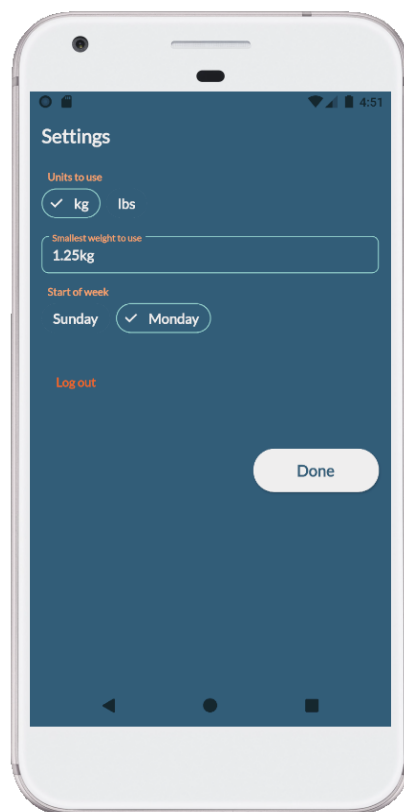
Joonis 8. Treeningplaanide vaade

Treeningplaanide vaade (Joonis 8)

1. Kasutajal on ülevaade kõigist treeningplaanidest.
2. Kasutaja saab muuta või lisada treeningplaane.
3. Treeningplaani lisamine/muutmine (Joonis 9)
 - a. Treeningplaanile saab lisada ning eemaldada päevi
 - b. Päevade vahemikke saab muuta
 - c. Päevadele saab lisada harjutusi
 - d. Harjutustele saab lisada *set*-e
 - e. *Set*-ide kordusi ning raskusi saab muuta
 - f. Treeningplaani nime saab muuta



Joonis 9. Treeningplaani muutmise vaade



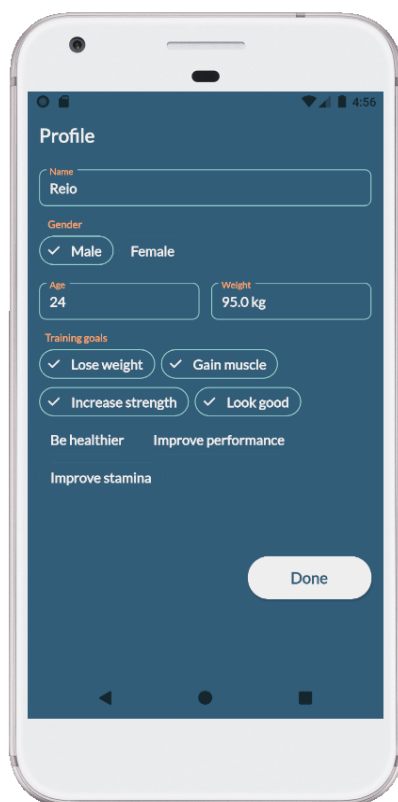
Joonis 10. Seadete vaade

Seadete vaade

1. Kasutaja saab valida, mis raskusühikuid kasutada
2. Kasutaja saab valida, mis on väikseim raskus, mille kaupa raskusi suurenda/vähendada
3. Kasutaja saab valida, mis päeva loetakse nädala alguseks
4. Kasutaja saab kasutajast välja logida

Profili vaade

1. Kasutaja saab muuta nime.
2. Kasutaja saab valida sugu.
3. Kasutaja saab muuta vanust ning kehakaalu.
4. Kasutaja saab valida treeningeesmärgid.



Joonis 11. Profiili vaade

Progressi vaade

1. Kasutaja saab valida kategooria, milles arengut näha
2. Kasutajale näidatakse arengut valitud kategoorias

3.2 Kasutusmugavus

Töö üheks peamiseks eesmärgiks oli luua hea kasutusmugavusega rakendus, sest autor leidis, et paljud olemasolevad on ühes või teises kategoorias puudulikud. Kuigi rakenduse oleks võinud ka teha veebirakendusena, tundus mobiilirakendus parem valik.

Mobiilirakenduse eelised veebirakendusega võrreldes:

- Mobiilirakendust saab paremini kohandada vastavalt seadmele, kuna on mobiilile mõeldud, veebirakendus peab toetama ka muid resolutsioone ning seadmeid, olles samal ajal sõltuv brauserite toimimisest
- Mobiilirakendust saab kasutada ilma internetiühenduseta
- Mobiilirakendusel on jõudlus parem[19]
- Mobiilirakenduses saab kasutada sisseehitatud žeste

Kuna jõusaali külastajad suurema osa ajast ei võta kaasa sülearvuteid, vaid kasutavad mobiile, tundub mobiilirakendus loogiline valik, kuna pole vaja toetada mitte mobiilseid seadmeid.

Rakendust luues autor proovis mitut erinevat värviskeemi ning lõpuks otsustas Ebtihaj Khan loodud värviskeemi [6]. See oli sobiv stiili ja kontrasti poolest, et oleks sobilik kasutada ka inimestele kelle nägemine on kehvem tavaliselt. Lisaks sellele aktiivse treeningu vaadetes peeti silmas, et rakendust oleks mugav kasutada ka ühe käega. Selle tõttu on raskuste ning korduste sisestamise ning harjutuste ning *set*-ide salvestamiseks mõeldud elemendid paigutatud ekraani alumisele poolele, et ka suuremate ekraanidega seadmete kasutajad saaksid ühe käega tähtsamaid funktsionaalsusi kasutada. Ekraani ülemisele poolele, mida on ühe käega rakendust kasutades raskem ulatada, on pandud elemendid, mille funktsionaalsust kasutatakse harvemini.

Lisaks jälgiti arenduse käigus, et rakendus oleks seadme omadustele reageeriv. Veendumiseks, et rakendus töötab korrektselt erinevate ekraani suuruste puhul, testis autor rakendust mitme erineva ekraani suurusega seadmega.

3.2.1 Olemasolevate rakenduste analüüs

Olemasolevaid rakendusi, mis pakuvad treeningplaane ning võimaldavad treeningssessioone logida ning analüüsida, on palju, kuid neid esineb mõningaid probleeme. Tihti on ühte rakendusse kombineeritud väga suur hulk funktsionaalsust - kõikvõimalikud eri treeningliigid, alustades kardiost ja jõusaalist, lõpetades joogaga. Sinna juurde pakutakse veel toitumisplaane ja muud. Tihti kasutaja tahab vaid ühte spetsiifilist funktsionaalsust, ning kui rakenduses on suur osa kasutaja jaoks mitteolulist funktsionaalsust, tekitab see kasutaja jaoks liigset kompleksust ja müra, rääkimata sellest, et fokuseerides laiale funktsionaalsusele, väheneb ühele kindlale funktsionaalsusele keskendumise osakaal. See võib põhjustada kvaliteedi languse.

FitnessAI

Üldiselt kasutajate rahulolu rakendusega on hea. Samuti saavutas rakendus App of the Day 2020 tiitli.

Välja toodud probleemid FitnessAI kasutajate poolt:

- Pole võimalik korduste arvu suurendada
- Harjutuste kogu on limiteeritud - näiteks 23.01.2020 seisuga polnud *incline barbell press*'i (kangiga rinnalt surumine kaldus pingil), mis on vägagi levinud harjutus.
- Treeningu ajal pole võimalik harjutust asendada mõne muuga. Kuna jõusaalis juhtub tihti, et keegi teine kasutab mõnda masinat või jõusaalis teatud masinat üldse pole, on kasutajatele vajalik, et saaks asendada harjutusi teiste samadele lihasgruppidele mõeldud harjutustega.
- Pole võimalik valida harjutusi, mida treeningutes vältida. See on vajalik kui kasutaja ei saa vigastuse või terviseprobleemide tõttu teatud harjutust sooritada.
- Pole võimalik harjutust vahele jätta, mille tõttu peab rakendusele "valetama", et sooritasid harjutuse, et jõuda järgmise harjutuseni.
- Rakendus sisaldab palju eri seksioone ja funktsionaalsust, mis võib olla segavaks faktoriks kasutajale, kes soovib ainult jõutrenni teha.
- Ilma valimata tasulist paketti, on funktsionaalsus väga limiteeritud, prooviperiood on vaid 7 päeva, mis on lühike periood, et näha, kas rakendus on kasutajale sobiv ning pakub soovitud tulemusi

Tugevused:

- Hea UI, lihtsasti arusaadav
- Vastab enamus kasutajate vajadustele
- Sobilik algajatele

My Workout Plan

My Workout Plan puudused:

- Kasutaja peab ise treeningkava looma või kasutama teiste kasutajate loodud kavasid. See on ühest küljest hea, sest kasutaja saab luua kava täpselt vastavalt oma soovidele, kuid nõuab rohkem aega ja vaeva kasutaja poolt.
- Treeningu ajal korduste ja raskuste sisestamine on ebamugav. Näiteks korduste puhul tuleb vajutada korduste väljale, siis avaneb dialoog korduste sisestamiseks, tuleb fookuseerida tekstiväli dialoogis ning avaneb klaviatuur ja saab sisestada uue korduste arvu ning vajutades *ok* salvestatakse uus väärtus.
- Kava koostades ei saa jaotada treeninguid päevade peale

My Workout Plan head omadused:

- Lai treeningute valik
- Harjutused on lihasgruppide järgi grupeeritud

3.2.2 Funktsionaalsuse võrdlus teiste rakendustega

Tabel 1. Loodud rakenduse, FitnessAI ja My Workout Plan võrdlus

Funktsionaalsus	Antud rakendus	FitnessAI	My Workout Plan
E-postiga sisselogimine	+		
Google-ga sisselogimine	+		+
Facebook-iga sisselogimine	+		
Parooli taastamine	+		
Ülevaade eelseisvatest treeningutest	+	+	
Ülevaade treeningute ajaloost	+	+	+
Harjutuste lisamine treeningu ajal	+		
Harjutuste eemaldamine treeningu ajal	+		
Harjutuste muutmine treeningu ajal	+		
Treeningplaani pakkumine kasutajale	+	+	
Aktiivse treeningsessiooni jälgimine	+	+	+
Kasutaja saab märkida sooritatud harjutusi koos raskuste ja kordustega	+	+	+
Kasutaja saab märkida sooritatud kordusi aktiivse <i>set</i> -i kaupa	+	+	+
Valikuliselt näitab rakendus puhkeperioodide taimerit	+	+	

Treeningu lõppedes annab rakendus ülevaate sessioonist	+	+	+
Treeningu ajal on ülevaade sooritatud/sooritamata harjutustest	+	+	
Profiili loomine/muutmine	+	+	
Rakenduse seadete muutmine	+	+	+
Ülevaade arengust	+	+	+
Ülevaade treeningkavadest	+	+	+
Võimalik treeningplaane lisada/muuta	+	+	+

Tabelist 1 on näha, et loodud rakendusel on palju funktsionaalsuseid, mida konkurentidel ei ole. See ei tähenda, et loodud rakendus on tingimata parem võrreldavatest rakendustest, võrreldavatel rakendustel on kindlasti ka funktsionaalsusi, mida loodud rakendusel pole. Kuid autor ei pidanud neid funktsionaalsusi piisavalt tähtsaks, et esialgses versioonis neid implementeerida. Üks esmane erinevus, mis on tabelist näha on see, et loodud rakendusel on rohkem sisselogimisviise. FitnessAI-l saab rakendust kasutada vaid telefoninumbri abil ning My Workout Plan puhul vaid Google kasutajaga. My Workout Plan puhul võib rakendus kaotada potentsiaalseid kasutajaid kui kasutajal pole Google kontot. FitnessAI puhul võib juhtuda sama kui kasutaja ei soovi kasutada või avaldada oma telefoninumbrit. Samuti pole telefoninumbri sisselogimist võimalik kasutada mobiilside puudumise korral. Lisaks sellele Üheks peamiseks erinevuseks võrreldes teiste rakendustega on see, et teistes rakendustes ei saa aktiivse treeningssessiooni ajal muuta harjutusi, mis on vaadates Play Store'i ja App Store'i kommentaare teiste rakenduste kohta, vägagi vajalik funktsionaalsus. Ning samuti on võrreldavatel rakendustel mõned teised puudused aktiivse sessiooni ajal.

3.3.4 Kasutusmugavuse saavutamine

Et saavutada head kasutusmugavust, on antud rakenduse raames analüüsitud teisi sarnaseid rakendusi ning neis kasutatud funktsionaalsust ja disaini aspekte, mille kohta kasutajad on andnud head tagasisidet, ning välditud või modifitseeritud osasid, mille juures kasutajad on leidnud probleeme. Samuti järgiti peatükis 2 mainitud disainiprintsiipe.

Ühed peamised probleemid, mis on plaanis lahendada, et saavutada head kasutusmugavust, on järgnevad:

- 1) Treeningu ajal oleks rakenduse kasutamine kiire ja peab tegema võimalikult vähe “klikke” ning samuti on andmete sisestamine kiire. Sellest tulenevalt saab kasutaja keskenduda treeningule ning rakendus on abivahend, mitte lisatöö kasutajale.
- 2) Rakendus on lihtsasti kasutatav ning arusaadav, vältida funktsionaalsuste kuhjumist vaadetes

Et aktiivse sessiooni ajal oleks raskuste ja korduste lisamine mugav ja kiire, on algselt automaatselt valitud soovitatav raskus ning korduste arv, millega harjutust sooritada. See on valitud vastavalt eelnevalt sooritatud treeningutele. Samuti on raskuste või korduste arv muudetav ühe nupuvajutusega ning *set*-i salvestamine toimub ühe nupuvajutusega. UI poolest on korduste salvestamine sarnane FitnessAI-ga. FitnessAI miinuseks on see, et kordusi ei saa suurendada ja vähendada saab vaid ühe korduse kaupa, juhul kui sooritatud korduste arv on väga erinev ettenähtud korduste arvust, on kasutajal vaja teha palju üleliigseid klikke. Lõputöö raames loodud rakenduses saab kordusi nii suurendada kui vähendada, ning seda mitme korduse kaupa. Lisaks FitnessAI-s ei saa aktiivse harjutuse ajal raskusi muuta. My Workout Plan puhul saab küll kordusi ning raskusi muuta, kuid see on tülikas. Nii korduste kui raskuse puhul on muutmiseks vaja kõigepealt fookuseerida väärtuse väli, siis avaneb dialoog, kus saab väärtust muuta. Dialoogis tuleb uuesti fookuseerida väärtuse väli, siis avaneb klaviatuur ning saab sisestada soovitud väärtuse. Kõige lõpuks tuleb dialoogis väärtus kinnitada. Võrreldes antud rakenduse ja FitnessAI 1 sammuga tuleb My Workout Plan-i puhul teha 4 sammu korduste või raskuse väärtuse muutmiseks.

Treeningu ajal harjutuste eemaldamine toimub ühe nupuvajutusega. Et harjutust lisada, saab nupust avada dialoogi, kust saab valida sobiva harjutuse. Samuti on võimalik kasutada otsingut harjutuse leidmiseks. Samuti on koguaeg nähtav aktiivne harjutus ja ka eelseisvad ning lõpetatud harjutused. Nii My Workout Plan kui FitnessAI puhul ei saa aktiivse treeningu ajal treeninguid asendada mõne teisega või lisada harjutusi. Samuti ei saa harjutusi eemaldada. See on halb kuna jõusaalis juhtub tihti, eriti kui on tihe ajavahemik nagu näiteks õhtul peale tööaega, et masin, mida kasutaja soovib kasutada, on hõivatud. Et säiliks treeningtempo, saab kasutaja asendada harjutuse mõne muu harjutusega, mis hõlmab samasid lihasgruppe ja pole jõusaalis kellegi teise poole kasutusel. Sama kehtib sellisel juhul kui teatud masinat ei ole jõusaalis, mida rakenduse kasutaja külastab.

3.3 Kasutatud tehnoloogiad

Antud rakendus on tehtud kasutades Flutter raamistikku. On ka mitmeid teisi häid raamistikke, milles mobiilirakendust arendada nagu näiteks React Native[17] ja Ionic[18]. React Native põhineb Javascriptil ning Ionic AngularJS-il. Kõik mainitud raamistikud on *cross-platform* (nii Android kui iOS kasutavad sama koodibaasi) ja võimaldavad kiiret arendust. React Native'i ja Ionicu eeliseks on see, et mõlemad kasutavad JavaScripti, mis teeb arendusprotsessiga alustamise kiiremaks kuna ei pea selgeks õppima uut keelt nagu Dart Flutteri puhul. Lisaks sellele on Flutter suhteliselt uus raamistik, mille tõttu on dokumentatsioon ning liidestamine teiste pistikprogrammide ning süsteemidega keerulisem. Teisest küljest Flutter on jõudluse poolest parem tänu *native* komponentidele, samas kui Ionic ja React Native sõltuvad Javascript sildadest, et moodulitega suhelda. Kokkuvõtlikult kõik antud raamistikud oleks olnud hea valik, ning React Native'i või Ionicu valimine oleks tõenäoliselt arendusprotsessi kiirendanud, kuid autor valis Flutteri, sest selle tulevik on paljulubav, jõudlus hea ning arendusprotsess mugav.

Sisemiselt kasutab Flutter Dart programmeerimiskeelt[7]. Rakenduse kasutamiseks on vaja sisse logida. Autentimiseks on kasutatud Firebase Authentication-it. Andmete haldamiseks on kasutatud Cloud Firestore'i. Kuna Flutter on suhteliselt uus raamistik, on paljud selle platvormi jaoks loodud pistikprogrammid algjärgus ning mitmete API-de

integreerimine keeruline ning vähe dokumenteeritud. Selle tõttu valitigi Firebase andmete haldamiseks, sest mõlemad, nii Flutter kui Firebase on loodud Google'i poolt ning API-de integreerimise kohta on dokumentatsioon parem.

Nii Flutter, Dart kui Firebase on autori jaoks uued tehnoloogiad, millega pole varem kokku puutunud. Töö käigus õpiti antud tehnoloogiaid kasutama.

3.3.1 Flutter

Flutter[9] on avatud lähtekoodiga kasutajaliideste loomiseks mõeldud tarkvara raamistik, mis on arendatud Google'i poolt. Peamiselt kasutatakse seda Androidi, iOS-i, veebi-, Windowsi, Maci ja Linuxi rakenduste loomiseks. See loodi 2017. Aastal. Flutterit kasutavad juba palju tuntud ülemaailmsed ettevõtted nagu Ebay, BMW, Square ja Alibaba.

Peamised põhjused, miks Flutterit on mobiilirakenduste arendamiseks hea kasutada:

1. Arendusprotsess on kiire tänu *hot reload*-ile.
2. Kasutajaliideste loomine on paindlik tänu kihilisele arhitektuurile, raamistik sisaldab laiab valikut olemasolevaid *UI* komponente. Ise uusi komponente luua on vaja väga harva. (Joonis 13)
3. *Native* (platvormidele omane) jõudlus - Flutteri kood kompileeritakse Dart *compiler*-eid (kompilaatoreid) kasutades ARM masinakoodi.

Antud raamistikus arendamiseks kasutatakse programmeerimiskeelt Dart[7], mis on savuti Google-i poolt loodud. Dart on spetsiaalselt arendatud kasutajaliideste loomist silmas pidades. Olemuselt on Dart väga sarnane teistele programmeerimiskeeltele nagu Java ja C#, millega autor on eelnevalt kokku puutunud. Saab kasutada *forEach*, *map*, *reduce* meetodeid ning *lambda* süntaksit (Joonis 12), mis on sarnane näiteks JavaScripti ja C#-ga. Dart võimaldab *hot reload*-i, mis tähendab, et rakenduse arenduse käigus tehes koodimuudatusi, laetakse uuesti vaid muudetud osa koodist, mitte kogu rakendus. See teeb arendusprotsessi mitu korda kiiremaks, sest koodimuudatused on *debug*-itavas seadmes koheselt nähtavad ning ei pea ootama, et rakendus uuesti laeks.

```

static double weightMaxForExercise(String exerciseId) {
  List<SetModel> allSets = [];
  userWorkouts
    .map((exercisesInWorkouts) => exercisesInWorkouts.exercises)
    .forEach((allExercises) {
      allExercises.where((x) => x.exerciseID == exerciseId).forEach((exercise) {
        allSets.addAll(exercise.sets);
      });
    });

  return allSets
    .map((x) => x.weight)
    .reduce((current, next) => current > next ? current : next);
}

```

Joonis 12. Dart koodinäide harjutuse jaoks maksimaalse raskuse leidmiseks

```

Container(
  child: Column(
    children: <Widget>[
      GestureDetector(
        onTap: () {
          Navigator.pushNamed(context, Pages.profile);
        },
        child: Icon(
          Icons.account_circle,
          color: Styles.primaryTextColor,
          size: 30,
        ), // Icon
      ), // GestureDetector
      Container(
        child: Text("Profile",
          style: Styles.textH4()
            .copyWith(fontSize: 12, fontWeight: FontWeight.normal)), // Text
        ) // Container
    ]
  )

```

Joonis 13. Flutter koodinäide Dartis põhivaate profiilinupust kastuaades olemasolevaid UI komponente

Arendamiseks kasutas autor Visual Studio Code[15] koodiredaktorit kuna selle jaoks on olemas ametlikult toetatud Flutteri ja Darti *extensionid* (laiendused). Alternatiivselt saaks kasutada ka Android Studiot, IntelliJ-d või Emacsi. Kuna Visual Studio Code on *lightweight* (kergekaaluline, ei nõua palju jõudlust) ning autor oli seda varem kasutanud, otsustas ta selle kasuks. Rakenduse debugimiseks kasutati erinevaid Android emulaatoreid ning füüsilise seadmeid.

3.3.2 Firebase

Firebase[8] on mobiili- ja veebirakenduste arendamiseks mõeldud platvorm. See pakub mitmeid erinevaid teenuseid ja tooteid, nagu näiteks pilveteenus, autentimine, analüütika, testimine, reklaamiteenus, mis võimaldavad rakendusi kiiresti arendada ning integreerida ilma et arendaja peaks muretsema suure infrastruktuuri haldamise üle.

3.3.3 Cloud Firestore

Rakenduse andmete haldamiseks valis autor Cloud Firestore-i. Esialgu autor kaalus AWS[11] kasutamist, kuid töö käigus leidis, et AWS-i Flutteriga liidestamise kohta on dokumentatsioon kehv ning otsustas Firestore-i kasuks. Cloud Firestore[10] on kergesti skaleeritav NoSQL andmebaas pilves. Andmebaas koosneb *collection*-itest, mis omakorda sisaldavad dokumente. Dokumendid sisaldavad andmeid ning võivad omakorda sisaldada uusi *collection*-e. Firestore on mobiilirakenduste jaoks hea, sest pakub *offline support*-i (rakendust saab kasutada ka internetiühenduse puudumise või kehvasti interneti korral).

3.3.4 Firebase Authentication

Firebase Authentication[12] pakub serveriteenuseid, mis võimaldavad lihtsasti kasutajaid autentida erinevate meetodite kaudu. Näiteks on võimalik autentida emaili, Google-i, Facebook-i ja Twitter-i kaudu. Samuti järgib Firebase Authentication OAuth 2.0 ja OpenID Connect standardeid. OAuth 2.0 on *industry-standard* (standardne ning tunnustatud viis antud sektoris) protokoll autoriseerimiseks pidades silmas lihtsust kliendi arendajale. OpenID Connect on lihtne identiteedikiht OAuth 2.0 protokolliga kohal. Kõik see tähendab, et rakendus on turvaline, ilma et arendaja peaks liigselt muretsema turvariskide pärast.

4 Tulemused

Esialgu oli autoril plaanis rakendust valideerida A/B testimise ning küsitluse läbi, kuid hiljem leidis, et sellest ei oleks eriti suurt kasu. Üks argument kasutajatestimist mitte teha, oli see, et praeguse ühiskonnakorralduse tõttu on jõusaalid suletud ning ainuke võimalus rakendust testida oleks imiteerides jõusaalis käiku, mis võib anda ebakorrektsid tulemusi. Lisaks sellele võiks kasutajad olla kasutanud teisi sarnaseid rakendusi, et nad oskaks anda tagasisidet, mis on perspektiivis teiste sarnaste rakendustega. Viimane põhjus on see, et rakenduse põhifunktsionaalsus, mis on aktiivse treeningu järgimine ja *set*-ide salvestamine, võtab maksimaalselt aega vaid mõne sekundi ning seal pole suurt kasu testimisel.

Kogu plaanitud funktsionaalsus saadi töö käigus valmis. Nagu eelpool on töös mainitud, oli peamiseks eesmärgiks luua kasutusmugavuse poolest parem rakendus kui on konkurendid samas valdkonnas. Kasutusmugavuse saavutamine algas sellest, et mõeldi läbi, kuidas jõusaali küllastajad jõusaalis käituvad, millised ebamugavused võivad jõusaalis ette tulla ja mis on hetkel olemasolevates rakenduses aspektid, mis teevad kasutamise ebamugavaks ning häirivad treeningu ajal harjutuste sooritamist. Analüüsiti ka sarnaseid rakendusi ning loodavas rakenduses prooviti võimalikult palju ebameeldivaid faktoreid rakenduse kasutamise protsessist eemaldada ning jätta alles osad, millega teiste rakenduste kasutajad olid rahul. Põhilised punktid läbi mille kasutusmugavust parendati olid järgnevad:

- Treeningu ajal rakenduse kasutamine on kiire ning võtab vähe aega, jättes fookuse harjutusele, mida kasutaja sooritab
- Rakendust on lihtne kasutada
- Ühe käega saab kasutada põhilisi funktsionaalsusi – treeningu alustamine, raskuste ja korduste muutmine, harjutuste ja *set*-ide salvestamine, treeningu lõpetamine
- Aktiivse treeningu käigus saab päeva kava kohandada vastavalt jõusaali oludele
- On minimiseeritud töö, mida kasutaja peab tegema kavade ning treeningute ettevalmistamiseks ning läbimiseks

Tabel 2. Tähtsamate funktsionaalsuste kasutusmugavuse võrdlus teiste rakendustega

	Antud rakendus	FitnessAI	My Workout Plan
Treeningu ajal raskuse vähendamine	1 nupuvajutus	Pole võimalik	3 nupuvajutust + väärtuse sisestamine klaviatuuril
Treeningu ajal raskuse suurendamine	1 nupuvajutus	Pole võimalik	3 nupuvajutust + väärtuse sisestamine klaviatuuril
Korduste vähendamine treeningu ajal	1 nupuvajutus	1 nupuvajutus	3 nupuvajutust + väärtuse sisestamine klaviatuuril
Korduste suurendamine treeningu ajal	1 nupuvajutus	Pole võimalik	3 nupuvajutust + väärtuse sisestamine klaviatuuril
Treeningu ajal harjutuse asendamine alternatiiviga	3 nupuvajutust	Pole võimalik	Pole võimalik

5 Kokkuvõte

Lõputöö eesmärk oli luua mobiilirakendus, mis töötab kasutaja personaaltreenerina ning on kasutusmugavuse poolest parem teistest sarnastest rakendustest. Kuigi kasutusmugavuse mõõtmine on suhteline ning võib erineda olenevalt kasutajast, sest kõik kasutajad ei kasuta rakendust samamoodi, saavutati autori arvates lõputöö eesmärk.

Töö käigus käigus implementeeriti funktsionaalsed nõuded ning koostati põhimõtted, kuidas saavutada parem kasutusmugavus. Seda võrreldi ka teiste sarnaste rakendustega. Lisaks autor põhjendab kasutatud tehnoloogiate valikut ning teeb lühikokkuvõtte tulemustest.

Kuna lõputöö maht pole piisavalt suur kogu soovitava funktsionaalsuse realiseerimiseks, arendatakse rakendust edasi ka peale lõputööd. Lõputöö raames valmis ainult rakenduse esialgne versioon ning kindlasti on funktsionaalsuse ning disaini osasid, mida saaks paremini teha ning ka tulevikus muudetakse.

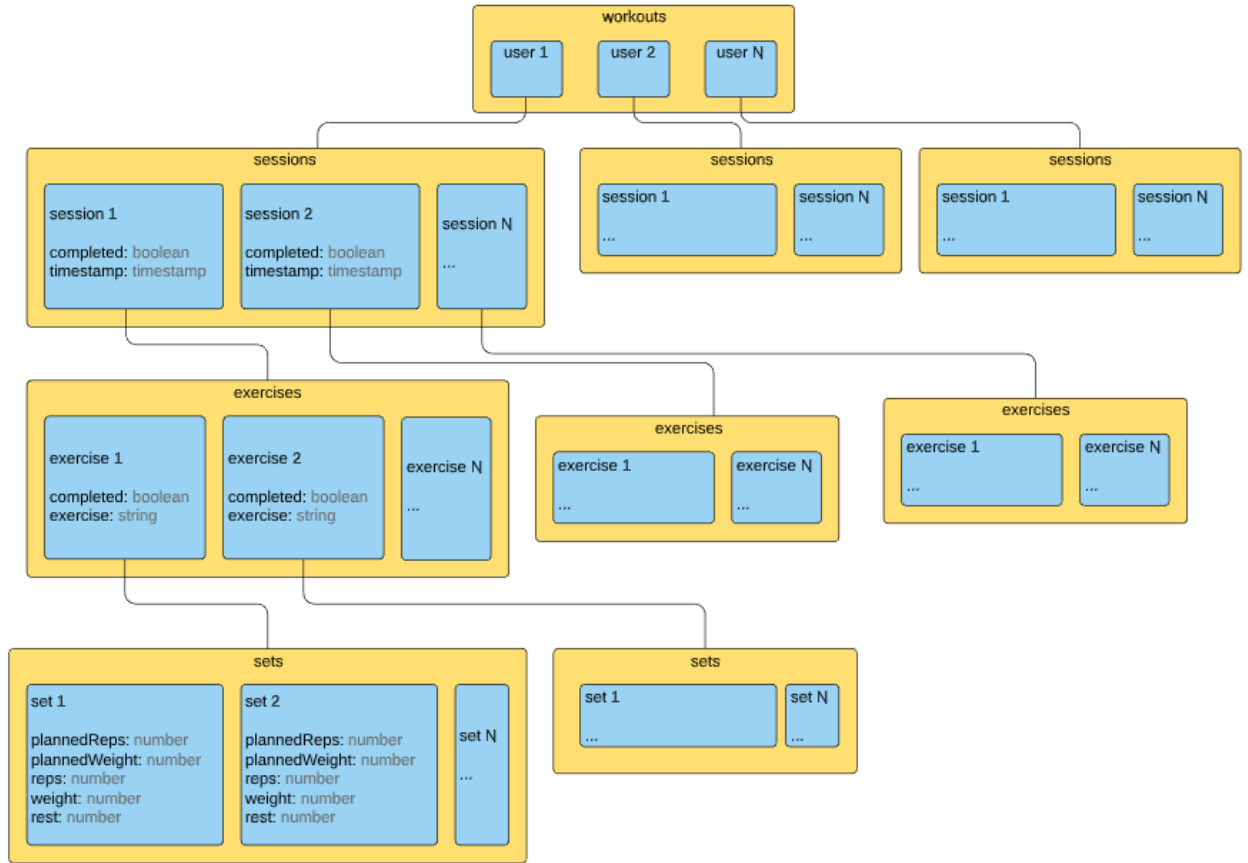
Kasutatud kirjandus

- [1] “FitnessAI,” [Võrgumaterjal]. Available: <https://www.fitnessai.com/> . [Kasutatud 4 mai 2020].
- [2] “My Workout Plan,” [Võrgumaterjal]. Available: <https://play.google.com/store/apps/details?id=com.myworkoutplan.myworkoutplan&hl=en> . [Kasutatud 4 mai 2020].
- [3] “Material Design,” [Võrgumaterjal]. Available: <https://material.io/design> . [Kasutatud 4 mai 2020].
- [4] “JEFIT,” [Võrgumaterjal]. Available: <https://www.jefit.com/> . [Kasutatud 14 mai 2020].
- [5] “How to Start Up (Accidentally),” [Võrgumaterjal]. Available: <https://medium.com/@jakemor/how-to-start-up-accidentally-abf7f05f602e> . [Kasutatud 12 mai 2020].
- [6] “8 Beautiful Flat Color Palettes For Your Next Design Project ,” [Võrgumaterjal]. Available: <https://digitalsynopsis.com/design/flat-color-palettes/> . [Kasutatud 12 mai 2020].
- [7] “Dart,” [Võrgumaterjal]. Available: <https://dart.dev/> . [Kasutatud 13 mai 2020].
- [8] “Firebase,” [Võrgumaterjal]. Available: <https://firebase.google.com/> . [Kasutatud 13 mai 2020].
- [9] “Flutter,” [Võrgumaterjal]. Available: <https://flutter.dev/> . [Kasutatud 13 mai 2020].
- [10] “Cloud Firestore,” [Võrgumaterjal]. Available: <https://firebase.google.com/docs/firestore> . [Kasutatud 13 mai 2020].
- [11] “AWS,” [Võrgumaterjal]. Available: <https://aws.amazon.com/> . [Kasutatud 13 mai 2020].
- [12] “Firebase Authentication,” [Võrgumaterjal]. Available: <https://firebase.google.com/docs/auth> . [Kasutatud 13 mai 2020].
- [13] “OAuth 2.0,” [Võrgumaterjal]. Available: <https://oauth.net/2/> . [Kasutatud 13 mai 2020].
- [14] “OpenID Connect,” [Võrgumaterjal]. Available: <https://openid.net/connect/> . [Kasutatud 13 mai 2020].
- [15] “Visual Studio Code,” [Võrgumaterjal]. Available: <https://code.visualstudio.com/> . [Kasutatud 13 mai 2020].

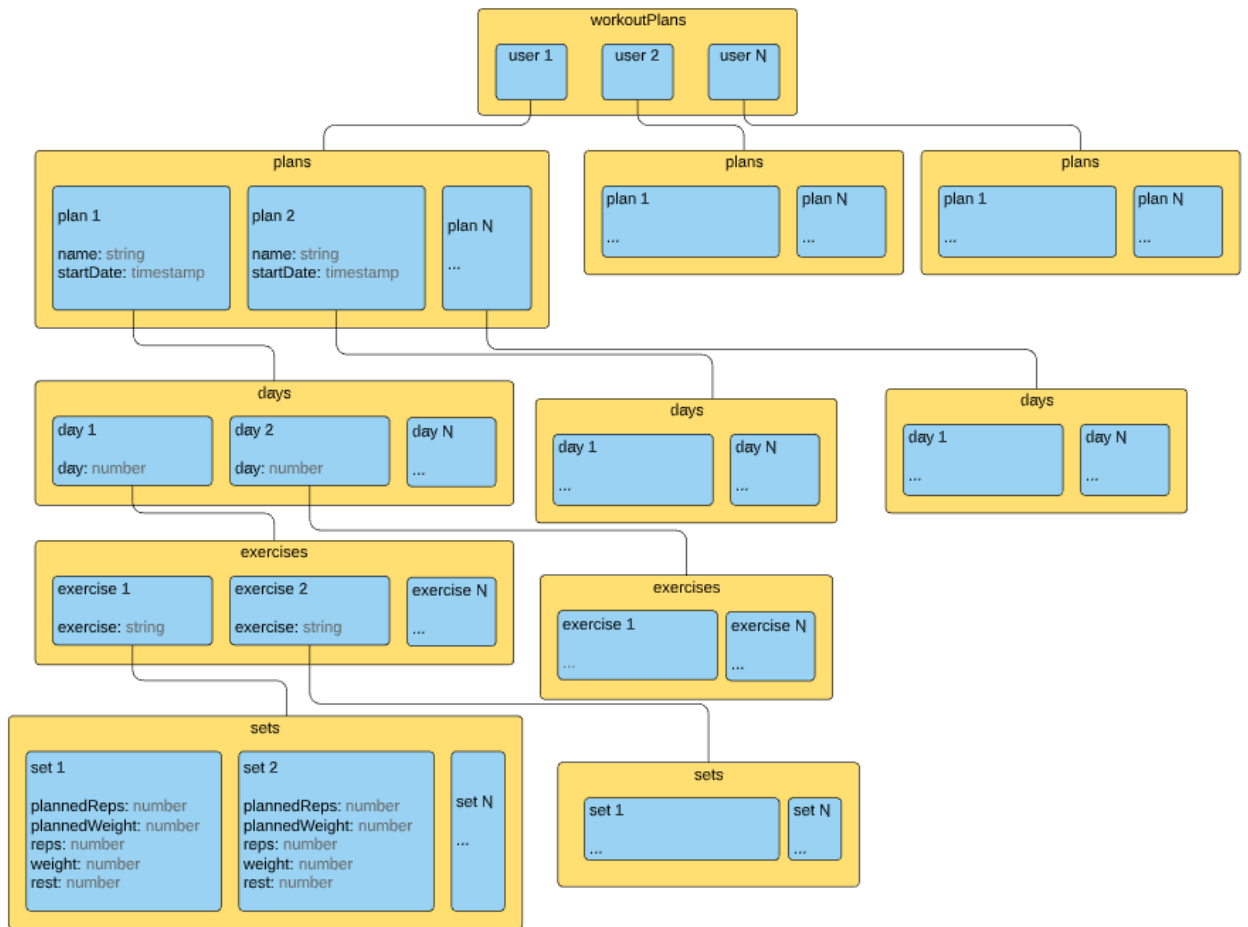
- [16] "Simple Workout Log," [Võrgumaterjal]. Available: <https://play.google.com/store/apps/details?id=com.selahsoft.workoutlog&hl=en> [Kasutatud 14 mai 2020].
- [17] "React Native," [Võrgumaterjal]. Available: <https://reactnative.dev/> . [Kasutatud 14 mai 2020].
- [18] "Ionic," [Võrgumaterjal]. Available: <https://ionicframework.com/> . [Kasutatud 14 mai 2020].
- [19] Florian Rösler, André Nitze, and Andreas Schmietendorf. "Towards a mobile application performance benchmark." International Conference on Internet and Web Applications and Services. Vol. 9. 2014 [Kasutatud 17 mai 2020].

Lisad

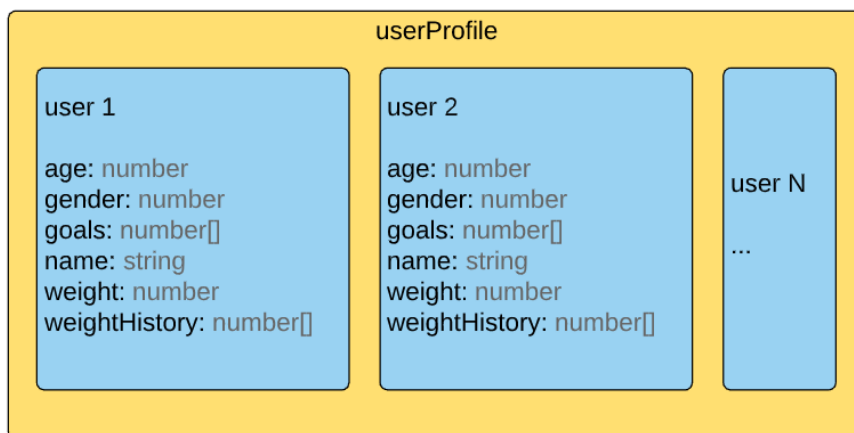
Lisa 1. Andmebaasi struktuur



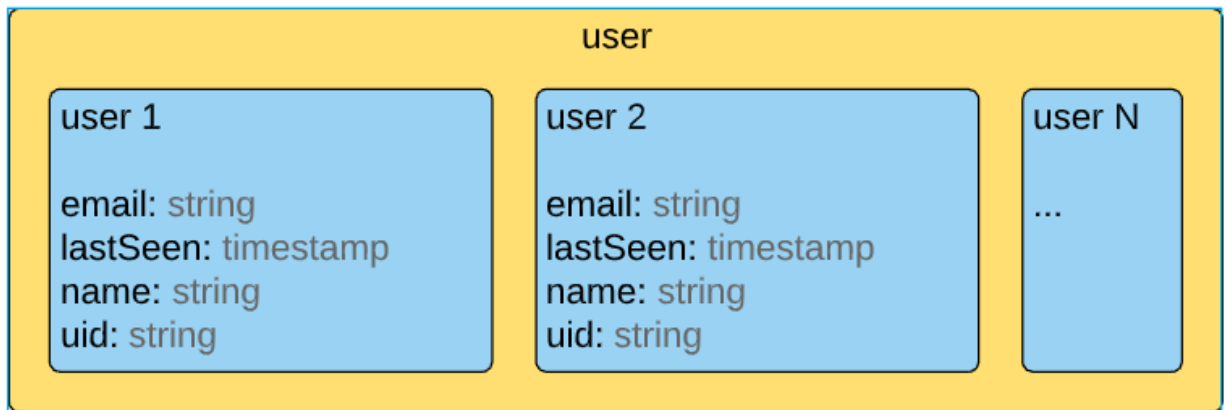
Joonis 14. Struktuur andmebaasi treeningutest



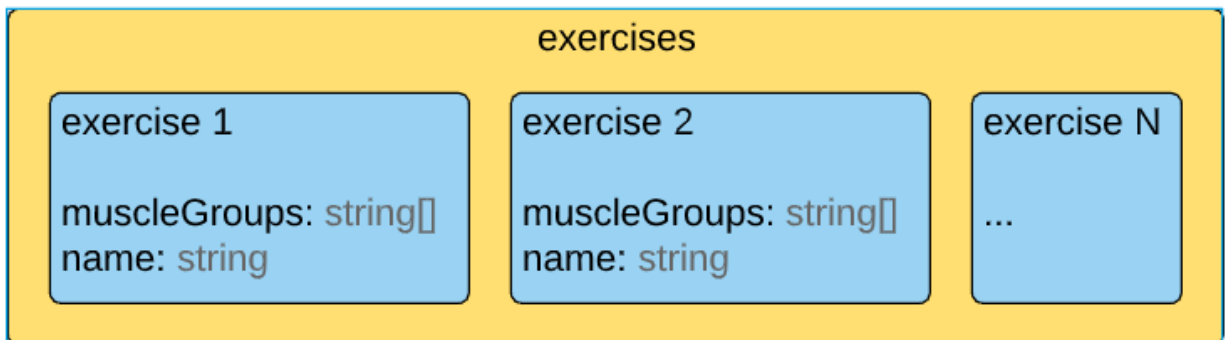
Joonis 15. Andmebaasi struktuur treeningplaanidest



Joonis 16. Andmebaasi struktuur kasutajate profiilidest



Joonis 17. Andmebaasi struktuur autentitud kasutajatest



Joonis 18. Andmebaasi struktuur harjutustest

Lisa 2. Koodinäited Firestore'ist andmete pärimiseks ja andmete sisestamiseks.

```
static Future<List<WorkoutCargo>> getUpcomingWorkout() async {
  DateTime _now = DateTime.now();
  DateTime _start = DateTime(_now.year, _now.month, _now.day, 0, 0);

  var result = Firestore.instance
    .collection('workouts')
    .document(authService.user.value.uid)
    .collection('sessions')
    .orderBy('timestamp')
    .limit(1)
    .where('completed', isEqualTo: false)
    .where('timestamp', isGreaterThanOrEqualTo: _start)
    .getDocuments()
    .then((sessionsSnapshot) async {
      List<WorkoutCargo> workoutsOnDay = new List<WorkoutCargo>();

      for (var document in sessionsSnapshot.documents) {
        var exercises = await getExercises(document);

        WorkoutCargo workout = WorkoutCargo.fromSnapshot(document);
        workout.exercises = exercises;
        workoutsOnDay.add(workout);
      }

      return workoutsOnDay;
    });
  return result;
}
```

Joonis 19. Koodinäide eelseisva treeningu pärimiseks Firestore'ist

```

static void saveSession(WorkoutCargo workout) {
    WriteBatch batch = Firestore.instance.batch();
    batch.updateData(
        Firestore.instance.document(
            'workouts/${authService.user.value.uid}/sessions/${workout.sessionId}',
            {"timestamp": Timestamp.now(), "completed": true});

    for (ExerciseInWorkout exercise in workout.exercises) {
        for (SetModel s in exercise.sets) {
            batch.updateData(
                Firestore.instance.document(
                    'workouts/${authService.user.value.uid}/sessions/${workout.sessionId}' +
                    '/exercises/${exercise.id}/sets/${s.id}',
                    {"reps": s.repCount, "weight": s.weight});
        }
        batch.updateData(
            Firestore.instance.document(
                'workouts/${authService.user.value.uid}/sessions/${workout.sessionId}' +
                '/exercises/${exercise.id}',
                {"completed": true});
    }
    batch.updateData(
        Firestore.instance
            .document('userProfile/${authService.user.value.uid}'),
        {"completedExercises": FieldValue.increment(1)});
    batch.commit();
}

```

Joonis 20. Koodinäide treeningssessiooni salvestamisest Firestore'i