

TALLINNA TEHNIKAÜLIKOOL  
Infotehnoloogia teaduskond

Konstantin Burdakov 164846IAPB

# **Sportlaste arengu monitoorimise ja hindamise mobiilirakenduse arendamine Android baasil**

Bakalaureusetöö

Juhendaja: Jekaterina Tšukrejeva  
Magistrikraad

Tallinn 2021

## **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: Konstantin Burdakov

20.05.2021

## **Annotatsioon**

Bakalaureusetöö eesmärgiks on arendada vabavaraline sportlase arengu juhtimise Android mobiilirakendus nimega Sporty, mille abil lahendatakse mitmeid treeninguprotsessiga seotud probleeme ja kitsaskohti. Rakenduse analoogid on tasulised ning avatud lähtekoodiga täisfunktsionaalset lahendust ei ole tuvastatud. Sellest tulenevalt iga kasutaja võib rakenduse vabalt kätte saada ning iga arendaja saab panustada lahenduse edasiarengusse.

Mobiilirakendus on mõeldud spordiasutustele, treeneritele, sportlastele ja lapsevanematele. Igal rollil on oma funktsionaalsus, mida toetab mitmekesine kasutajaliides. Programmi abil on võimalik luua (rakenduses) sportlasi, treenereid, meeskondi ja treeninguid. Lisaks on arendatud sportlase hinnangu andmise funktsionaalsus, mis tulevikus võimaldab vaadata kõikide soorituste koondtulemusi ja teha parandusi treeninguprotsessis.

Programm on realiseeritud arhitektuuril, mis koosneb mitmest elemendist ning hõlmab endas versioneerimise ja mastabeerimise võimekust. Rakendus on arendatud kaasaegse tehnoloogia abil ning mobiilirakenduse töö on testitud simulatsiooni, ühik- ja kasutaja testidega.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 33 leheküljel, 6 peatükki, 12 joonist, 2 tabelit.

## **Abstract**

### **Android Mobile Application Development for Monitoring and Evaluating the Development of Athletes**

The aim of the bachelor's thesis is to develop a freeware athlete development management Android mobile application called Sporty, which helps to solve many problems and bottlenecks related to the training process. Analogs of the application are paid and no full-featured open source solution has been identified. As a result, each user is free to access the application and each developer can contribute to the further development of the solution.

The mobile application is intended for sports institutions, coaches, athletes and parents. Each role has its own functionality, supported by a diverse user interface. With the help of the program it is possible to create athletes, coaches, teams and trainings. In addition, the Athlete Assessment Functionality has been developed, which in the future will allow you to view the overall results of all performances and make improvements in the training process.

The program is implemented in an architecture that consists of several elements and includes versioning and scaling capabilities. The application has been developed using modern technology and the work of the mobile application has been tested with simulation, unit and user acceptance tests.

The thesis is in Estonian and contains 33 pages of text, 6 chapters, 12 figures, 2 tables.

## Lühendite ja mõistete sõnastik

API	<i>Application Programming Interface</i> - rakendusliides
APK	<i>Android Application Package</i> - Android rakenduse pakett
HTTP	<i>Hypertext Transfer Protocol</i> - hüpertexti edastusprotokoll
iOS	<i>iPhone Operating System</i> - Apple ettevõtte mobiilsete seadmete operatsioonisüsteem
JSON	<i>JavaScript Object Notation</i> - andmevahetusvorming
PaaS	<i>Platform as a Service</i> - pilveteenus, mis on võimeline hallata erinevaid elemente, nagu infrastruktuur ja operatsioonisüsteem
PK	<i>Primary key</i> - andmebaasi tabelis olev primaarvõti
REST	<i>Representational State Transfer</i> - veebirakenduste arhitektuur
SQL	<i>Structured Query Language</i> - struktuuripäringukeel
UML	<i>Unified Modeling Language</i> - ühtne modelleerimiskeel

## Sisukord

1 Sissejuhatus .....	9
2 Sportlase arengu juhtimise tarkvara .....	10
2.1 Alternatiivsed lahendused.....	11
2.2 Lahenduse valik.....	13
3 Rakenduse analüüs .....	14
3.1 Metoodika.....	14
3.2 Kasutuslood .....	15
3.3 Mobiilirakendusele esitatavad nõuded .....	19
3.4 Arendustehnoloogia valik.....	21
3.5 Arenduskeskonna valik.....	23
4 Arhitektuur .....	25
4.1 Koodibaasi ülesehitus .....	27
4.2 Andmete esitamine kliendirakenduses .....	28
4.3 Kasutajaliides.....	30
4.4 Testimine .....	34
5 Tulemused ja järeldused .....	38
5.1 Realiseeritud funktsionaalsus .....	38
5.2 Järeldused ja ettepanekud .....	39
5.3 Edasiarendus .....	40
6 Kokkuvõte .....	41
Kasutatud kirjandus .....	42
Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks .....	44
Lisa 2 – Ressursid.....	45

## Jooniste loetelu

Joonis 1. Ekraani voodiagramm. ....	17
Joonis 2. UML diagramm. ....	18
Joonis 3. Maailmaturul kasutatavate mobiiliseadmete operatsioonisüsteemide osakaal.21	
Joonis 4. Rakenduse arhitektuur. ....	25
Joonis 5. Olemi-suhte diagramm. ....	26
Joonis 6. Kohal käimise statistika vaade. ....	28
Joonis 7. Andmete töötlemine statistika vaate genereerimiseks. ....	29
Joonis 8. Statistilise vaate vertikaalandmete genereerimine. ....	30
Joonis 9. Kasutaja registreerimise (vasakul) ja profiili (paremal) vaade. ....	31
Joonis 10. Treeninguplaani (vasakul) ja kasutaja statistika (paremal) vaade. ....	33
Joonis 11. Meeskonna detailide (vasakul) ja klubiliikme hindamise (paremal) vaade. .	34
Joonis 12. Autoriseerimise ja verifitseerimise ühiktest. ....	36

## **Tabelite loetelu**

Tabel 1. Alternatiivsete rakenduste võrdlus. ....	11
Tabel 2. Sisse logimise ja grupi vaatamise funktsiooni kasutuslood. ....	16



# 1 Sissejuhatus

Tänapäeval praktiliselt iga inimene on käinud spordi seksioonides. Olenemata treeningute tasemest, iga sportlase treeningu protsess peab olema planeeritud ja organiseeritud ning selle eest vastutab tavaliselt treener. Nimetatud isiku ülesanne on jaotada iga sportlase koormus ühtlaselt lähtuvalt sportlase võimetest, vanusest ja personaalsest arengukavast. Treeninguprotsessi üldine eesmärk on omandada põhioskusi ja teadmisi sporditehnika põhialustest, treeningu strateegiast ja spordiala võistlusreeglitest.

Iga treeneri ülesanne on täita treeningualane dokumentatsioon, mille sisu koosneb õppe- ja treeningukavadest, ohutusalse instruktaaži andmetest ja korrast, spordiala normatiivide täitmise statistikast ja paljust muust. Kuna treeneritöö tehakse enamasti kontori tingimustest väljaspool, siis sellega kaasnevad probleemid kontoritarkvara või arvutitehnika kasutamisega. Sellest tulenevalt paljud treenerid täidavad dokumentatsiooni paber kandjal või hiljem kannavad sisse andmed kontoritarkvara abil.

Eeltoodud lähenemisel esineb mitu puudust. Esiteks, treeneril peab olema oskus kasutada kontoritarkvara. Teiseks, treeneril peab olema alati kaasas sülearvuti (väiksematel mobiilsetel seadmetel kontoritarkvara ei ole piisavalt mugav). Kolmandaks, nimetatud andmete sisestamise viis ei anna võimalust sportlasel või tema vanematel igal hetkel jälgida treeningutulemusi.

Lõputöö eesmärk on luua avatud lähtekoodiga Android mobiilirakendus, mille abil on võimalik juhtida sportlase treeninguprotsessi. Rakenduse arendamiseks analüüsitakse olemasolevaid lahendusi, määratakse nõuded rakendusele, valitakse programmeerimiskeel ja arenduskeskkonnad, viiakse läbi ühik- ja kasutajatestid.

Lõputöö metoodika hõlmab endas kvalitatiivset sisuanalüüsi, mille abil analüüsitakse olemasolevad lahendused, tehnilist dokumentatsiooni ja arendusalased käsiraamatud. Lõpus teostatakse arendatud rakenduse funktsionaalsuse kvantitatiivne võrdlus olemasolevate lahendustega ja tehakse järeldused.

## 2 Sportlase arengu juhtimise tarkvara

Iga sportlase arengu eelduseks on korralikult juhitud treeninguprotsess. Selleks on tarvis täita erinevaid planeerimisdokumente ja pidevalt jälgida statistilisi andmeid. Tänapäeval dokumentatsiooni haldamiseks kasutatakse kontoritarkvara ja üheks tuntud näiteks on Microsoft Office komplekslahendus [1]. Kontoritarkvara kasutamine treeneritöös ei ole kõige efektiivsem lahendus. Põhjusi on selleks mitu:

- Treeneri põhitegevus toimub treeningu toimumise alas (nt jõusaal või staadion), kus pole võimalik kasutada tavapäraselt kontoritarkvara mobiilsuse puudumise tõttu. Täiendavate mobiilsete seadmete hankimine (nt tahvelarvuti) tekitab lisakulud ja ei ole alati mugav kasutamiseks. Lisaks eelnevale, kontoritarkvara eest tuleb ka perioodiliselt maksta.
- Kontoritarkvara nõuab kasutajalt teatud tasemel oskusi, mille saamiseks on tarvis läbida kursus või koolitus. See omakorda tähendab lisakulu. Kuna kontoritarkvara versioone on palju ja need perioodiliselt uuenevad, siis tihti on vaja kombineeritud oskusi.
- Paberile kantud ja hiljem arvutisse sisestatud andmetes võivad esineda erinevusi ja sisestusvigu, mis on seotud inimfaktoriga (st toimub andmete adekvaatsuse kadu). Lisaks on antud meetod ajamahukas.
- Kontoritarkvara kasutamine ei anna asjaosalistel võimalust jälgida statistilised andmed jooksvalt ja automaatselt.

Eeltoodud puuduste likvideerimiseks pakuvad mitmed arendusettevõtted sportlase arengu juhtimise programme. Nende ühine omadus on mobiilsus, st treener võib kasutada rakendust oma telefonis olenemata asukohast ja situatsioonist. Vaatamata sellele, on tarvis hoolega analüüsida turul pakutud lahendusi, kuna igal lahendusel on omad eripärad ja puudused, mis kokkuvõttes ei pruugi olla iga treeneritöö jaoks sobivad.

## 2.1 Alternatiivsed lahendused

Käesolevas alapeatükis analüüsitakse turul pakutud lahendused sportlaste tulemuste juhtimise valdkonnas. Autor otsustas analüüsida neli populaarset lahendust: Журнал тренера (Android ja iOS versioonid), Тренерская Online ja Sportlyzer [2] - [3]. Analüüsi eesmärk on defineerida lahenduste eelised ja tugevused (vt tabel 1).

Tabel 1. Alternatiivsete rakenduste võrdlus.

Parameeter	Журнал тренера	Журнал тренера	Тренерская Online	Sportlyzer
Platvorm	iOS	Android	Web	Web, Android, iOS
Hind/EUR (26-50 sportlast)	1,84	0	20	20
Grupid	Jah	Jah	Jah	Jah
Hinnang treeningu kohta	Jah	Jah	Jah	Jah
Võlgnevuste jälgimine	Jah	Jah	Jah	Jah
Edukuse hindamine	Jah	Ei	Jah	Jah
Sportlase ülesannete salvestamine	Ei	Jah	Jah	Jah
Suhtlus sportlasega	Ei	Ei	Jah	Jah
Suhtlus lapsevanematega	Ei	Ei	Jah	Jah
Plaanide koostamine	Jah	Jah	Jah	Jah
Andmete eksportimine	Jah (osaliselt)	Ei	Jah	Jah
Personaalsete andmete hoiustamine kliendi seadmes	Jah	Jah	Ei	Ei

Tabelis 1 esitatud (vasakult paremale) esimest kaks lahendust on monoliitsed, st mobiilirakendust installeeritakse otse seadmesse vastavalt kasutatavale platvormile. Sellest tulenevalt andmete salvestamiseks kasutatakse failipõhist (seadme) andmebaas. See tähendab, et ka kasutajate isikuandmed jäävad kasutaja seadme mällu ja rakenduse pakkuja ei pea muretsema nimetatud andmete turvalise hoiustamise eest oma serveris. Siinkohal on oluline mainida arendaja kulud, mida vähem kliendiandmeid (ja ka muid andmeid) on tarvis hoida serveris, seda väiksemad on infosüsteemi ülevalpidamise kulud.

Esimese lahenduse tähtsaks eeliseks teise lahenduse ees on andmete eksportimise võimalus, st on võimalik koostada andmestik kasutamiseks kontoritarkvaras (nt

statistiliste mudelite väljatöötamiseks ja koondraportite tegemiseks). Eksportida saab aga oluliselt piiratud infot, mistõttu tulevikus teostatav andmete analüüsi kvaliteet ja täiuslikkus ei pruugi rahuldada. Täiuslikum andmete eksportimise funktsionaalsus on kolmandal ja neljandal rakendusel.

Rakenduste analüüsi oluline aspekt on kulu, mis tekib rakenduse kasutamisel. Kuna iga organisatsiooni protsess peab olema kuluefektiivne ja optimeeritud, siis rakenduse maksumus peab olema kaalutud. Rakenduse kasutamisega seotud kulu on kõige kõrgem kolmandal ja neljandal rakendusel, kusjuures nimetatud kulu on oluliselt suurem võrreldes kahe esimesega. Fakt on ka see, et kolmandal ja neljandal rakendusel on ka rohkem funktsioone, millest on tingitud ka kõrgem tasu.

Kolmanda ja neljanda lahenduse kliendivaate analüüsides võib mainida, et tegemist on mitmekesise arendusega. Iga roll omab eraldi kasutajaliidest: sportlase, treeneri, lapsevanema ja spordi sektsiooni juhi vaated. Kasutajaliides on arendatud vastavalt rolli vajadustele ja omab vastavat funktsionaalsust. See annab võimaluse mugavalt arendada igat liidest eraldi, mis oluliselt tõstab efektiivsust tarkvara edasiarenduse faasisid või vigade parandamisel ja testimisel.

Oluliseks eeliseks on suhtlusvõimalus ja infovahetus spordiasutuse ja kliendi vahel, eriti juhul, kui tegemist on laste sektsiooniga – vanemad saavad täiendavat kontrolli laste igapäeva tegevuste üle (treeningukavad, plaanid, trenniajad, edukus jne). Antud funktsionaalsus tõstab infovahetuse kiirust ja suunab kliente kasutama spordiasutuse infokanalit tsentraliseerides selleks infovoogu. See aitab ära hoida mitmeid organisatoorseid probleeme, mis võivad tekkida kliendi ja spordiasutuse vahel. Tabelis 1 esitatud võrdluse kohaselt realiseeriti suhtlusvõimekust ainult rakendustes kolm ja neli.

Selleks, et sportlase areng oleks edukas, on vajalik fikseerida tema tulemused. Seda aga ei võimalda teine rakendus, kusjuures sama rakenduse iOS versioon seda aga võimaldab. Rakenduse Android versioon võimaldab aga sportlase ülesannete salvestamist, mida iOS versioonis ei ole realiseeritud. Selles kategoorias liidrid on kolmas ja neljas lahendus – mõlemal realiseeriti nimetatud funktsionaalsus.

Lisaks olemasolevate lahenduste analüüsile uuris autor avatud lähtekoodiga (tasuta) rakendusi, mis lahendavad lõputöös esitatud probleemi. Antud juhul leiti ainult üks lahendus, millel puudub dokumentatsioon ja seoses sellega käivitada seda ei õnnestunud

[4]. Nimetatud rakendus hõlmab endas tagarakendust (*back end*) ja SQL tabelite valmistamise skripti. Kuna rakendusel puudub mistahes dokumentatsioon, siis on suhteliselt keeruline antud rakendust kasutada ja edasi arendada.

## **2.2 Lahenduse valik**

Eeltoodud analüüsi kokku võttes selgus, et eksisteerib kaks rakendust, mille funktsionaalsus on mitmekesine ja mille kasutamine aitab lahendada lõputöös esitatud probleemi. Probleem on aga rakenduste maksumuses – mida rohkem meeskondi ja sektiioone soovib spordiasutus programmi sisestada, seda suurem tuleb arve. Antud momendil ei ole ka teada teenuseosutaja hinnakujunduspoliitika ja ei ole garanteeritud, et hinnad ei tõuse tulevikus. Puuduseks on ka see, et rakendust saab alla laadida standardkujul ja lisafunktsionaalsuse arendamine ei pruugi olla võimalik isegi lisatasu eest.

Nimetatud puuduste tõttu otsustas autor arendada vabavaralist tarkvara, mille abil kavatsetakse lahendada sportlase tulemuslikkuse juhtimise probleemi. Arendamisel orienteerutakse teises peatükis läbiviidud analüüsile, täpsemalt aga kahe viimasele lahendusele.

Autori arvates on mainimist väärt kaks viimast lahendust ka sellepärast, et nendes on realiseeritud lisafunktsioonid, mis on kasulikud mitte ainult treenerile. Sisuliselt tegemist on rakendustega, mille abil on võimalik juhtida spordiasutuse tegevust ja mis on suhteliselt suureks abiks asutuse juhtkonnale.

## 3 Rakenduse analüüs

Käesolevas peatükis teostatakse arendatava rakenduse analüüs. Selleks pannakse paika analüüsi metodoloogia, tulenevalt kasutuslugudest määratakse tarkvarale nõuded ja valitakse arendustehnoloogia (programmeerimiskeeled ja arenduskeskkonnad).

### 3.1 Metoodika

Lõputöö metodoloogiline lähenemine hõlmab endas andmekogumise ja andmeanalüüsi meetodeid. Peamiseks andmekogumismeetodiks on kvalitatiivne sisuanalüüs, mille abil analüüsitakse erinevaid allikaid (peamiselt käsiraamatud ja artiklid veebikeskkonnas) [5]. Kuna antud temaatika kohta praktiliselt kogu dokumentatsioon asub veebis, siis enamus materjalidest on elektroonilisel kujul. Kvalitatiivset andmekogumismeetodit kasutatakse järgmisteks ülesanneteks:

- Olemasolevate lahenduste uurimine ja analüüs.
- Olemasolevate alternatiivide võtmefunktsionaalsuse välja selgitamine.
- Funktsionaalsete nõuete määramine kogutud informatsiooni põhjal.
- Mittefunktsionaalsete nõuete seadmine analüüsitud andmete alusel.
- Programmeerimiskeelte uurimine ja valik.
- Arenduskeskkondade uurimine ja valik.
- Rakenduse arhitektuuri valimine ja analüüs.
- Testimistehnoloogia valik.

Andmeanalüüsi meetodi näol on tegemist kvalitatiivse ja kvantitatiivse võrdluse tegemisega, mille raames analüüsitakse kogutud andmed ja tehakse otsus parima või sobivaima kasuks [5]. Andmeanalüüsi tegemisel kasutatakse ainult antud lõputöö raames kogutud andmeid, tagades uurimuse kindlat struktuuri.

Metoodikat rikastavad kasutuslood, mida autor on koostanud jalgpalliklubi Narva Trans (edaspidi tellija) treeninguprotsessi stsenaariumi alusel [6]. Autor on antud klubi tegevjuht ja oskab detailselt kirjeldada organisatsioonivajadusi lähtuvalt klubi äriprotsessist. Kasutuslood on põhilised funktsionaalsust kirjeldavad elemendid.

## 3.2 Kasutuslood

Iga tarkvara arendamine algab selle funktsionaalsuse kirjeldamisega. Üheks meetodiks kirjeldada tarkvara kasutamist ja sellele esitatavad nõuded on formuleerida kasutuslood. Nimetatud lähenemine aitab minimeerida erinevusi kasutaja ootuste ja arendaja tulemuste vahel. Lisaks eelnevale aitavad kasutuslood säästa aega ja ressursi, kuna klient saab ise defineerida protsesse koos sammudega, mida ta soovib tulemuseks saada. [7]

Arendaja taustast lähtuvalt on oluline analüüsida kasutuslood, kuna viimased annavad võimaluse aru saada, millist arhitektuurilist realisatsiooni oleks mõistlik antud rakenduse juures kasutada. Mõned kasutuslood viitavad tehnilistele detailidele, mida on tähtis analüüsi faasis tähele panna. [8]

Lõputöö raames analüüsitud kasutuslugude koguarv on 21. Kasutuslood on tehtud viie rolli lõikes: tavakasutaja, juhataja, sportlane, treener ja lapsevanem. Iga registreeritud kasutaja saab alguses tavakasutaja rolli, kes peab ennast verifitseerima emaili või telefoni teel (kontaktandmed ilmuvad peale vastava nupu vajutamist registreerimise protsessi lõpus).

Kõige suuremate võimalustega roll on juhataja, kes saab määrata tavakasutajat erinevatesse rollidesse vastavalt oma rollile organisatsioonis. Nimetatud roll ei hõlma absoluutselt kõiki funktsioone, nt juhataja ei saa määrata sportlasele hinnangut soorituste või käitumise kohta, kuid juhataja omab kõik ülejäänud treeneri funktsioone.

Rollid on arendatud põhimõttel, et neid on võimalik tulevikus modifitseerida. Iga uue funktsionaalsusega saab redigeerida ka rolle, määrates neile erinevaid õigusi. Nimetatud tegevus annab võimaluse mobiilirakenduse väärtuse tõstmiseks.

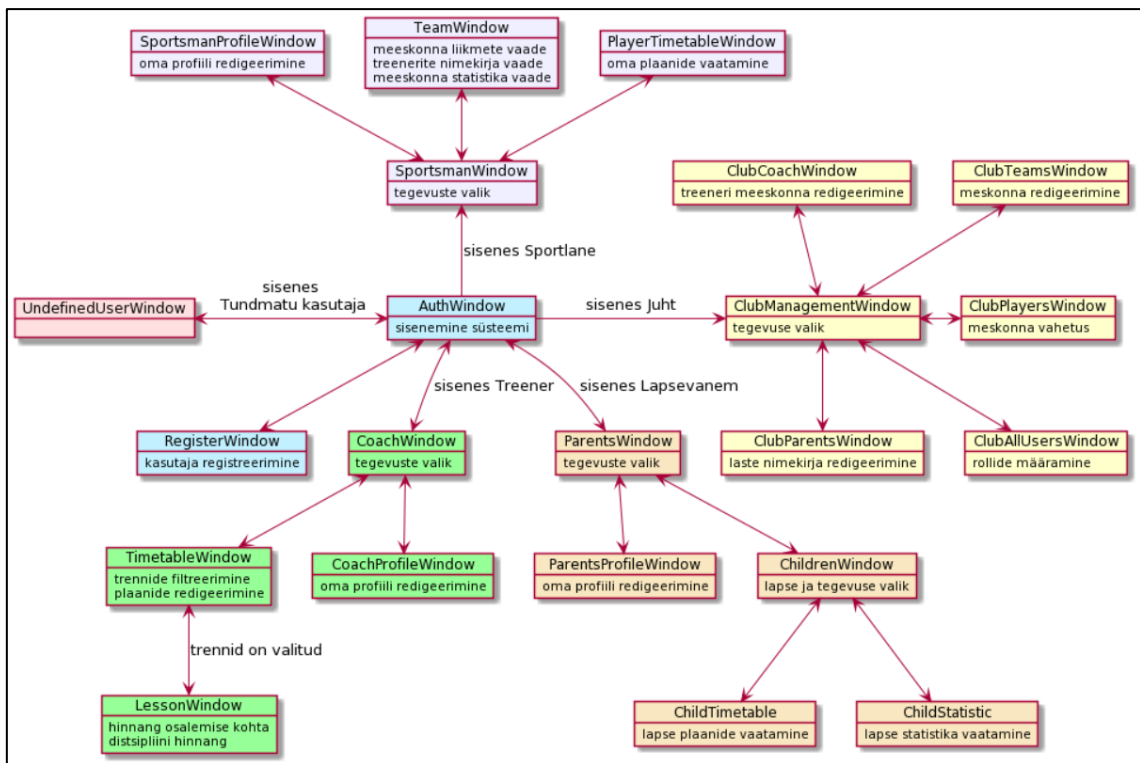
Iga kasutuslugu on eraldi lahti kirjutatud ja iga kasutuslugu hõlmab endas instruksioone, mida ja mis järjekorras tehakse. Selleks valmistas autor eraldi formaadi, et iga kasutuslugu oleks ühtemoodi struktureeritud ja seda oleks mugav lugeda. Formaat on koostatud vastavalt organisatsiooni nõuetele ja vajadustele. Osa kasutuslugudest on esitatud tabelis 2.

Tabel 2. Sisse logimise ja grupi vaatamise funktsiooni kasutuslood.

<b><i>Kasutuslugu 1</i></b>	
Nimetus	Sisse logimine
Eesmärk	Logida sisse
Eeldus	Avatud aken „Sisse logimise aken“
Peamine stsenaarium	1. Sisestada kasutaja andmed (kasutajatunnus ja parool)
	2. Vajutada nupule <i>Log in</i> (inglise keeles logi sisse)
Tulemused	Avatud aken, mis vastab kasutaja rollile
Alternatiivne stsenaarium	1. Sisestada kasutaja andmed (kasutajatunnus ja parool)
	2. Vajutada nupule <i>Log in</i> (inglise keeles logi sisse)
Alternatiivse stsenaariumi tulemused	Ekraanile on ilmunud veateada <i>Wrong username or password</i> (inglise keeles vale kasutajatunnus või parool)
<b><i>Kasutuslugu 2</i></b>	
Nimetus	Oma grupi andmete vaatamine
Roll	Sportlane
Eesmärk	Tutvuda oma grupi liikmete andmetega
Eeldus	Sportlase peaaken on avatud
Peamine stsenaarium	1. Vajutada nupule <i>Teams</i> (meeskond inglise keeles)
	2. Avatud aknas valida vastav nupp <i>Schedule/Players/Coaches</i> (inglise keeles plaan/mängijad/treenerid)
	3. Vastavalt valitud nupule ilmub kas plaani, mängijate või treenerite vaade
Tulemused	Plaan, mängijate või treenerite nimekiri on avatud
Alternatiivne stsenaarium	Puudub
Alternatiivse stsenaariumi tulemused	Puuduvad

Kasutuslood on tõhus tarkvara arendusprotsessi juhtimise vahend. Selle abil on võimalik üles ehitada, järjestada ja struktureerida arendaja ülesandeid. Autoril oli kasutuslugudest suhteliselt palju abi. Arendusprotsessi kirjeldamiseks ja tõhustamiseks otsustas autor koostada ekraani voodiagrammi (vt joonis 1) ja UML diagrammi (vt joonis 2). Nimetatud diagrammi valmistati lähtudes UML metodoloogiast [9].

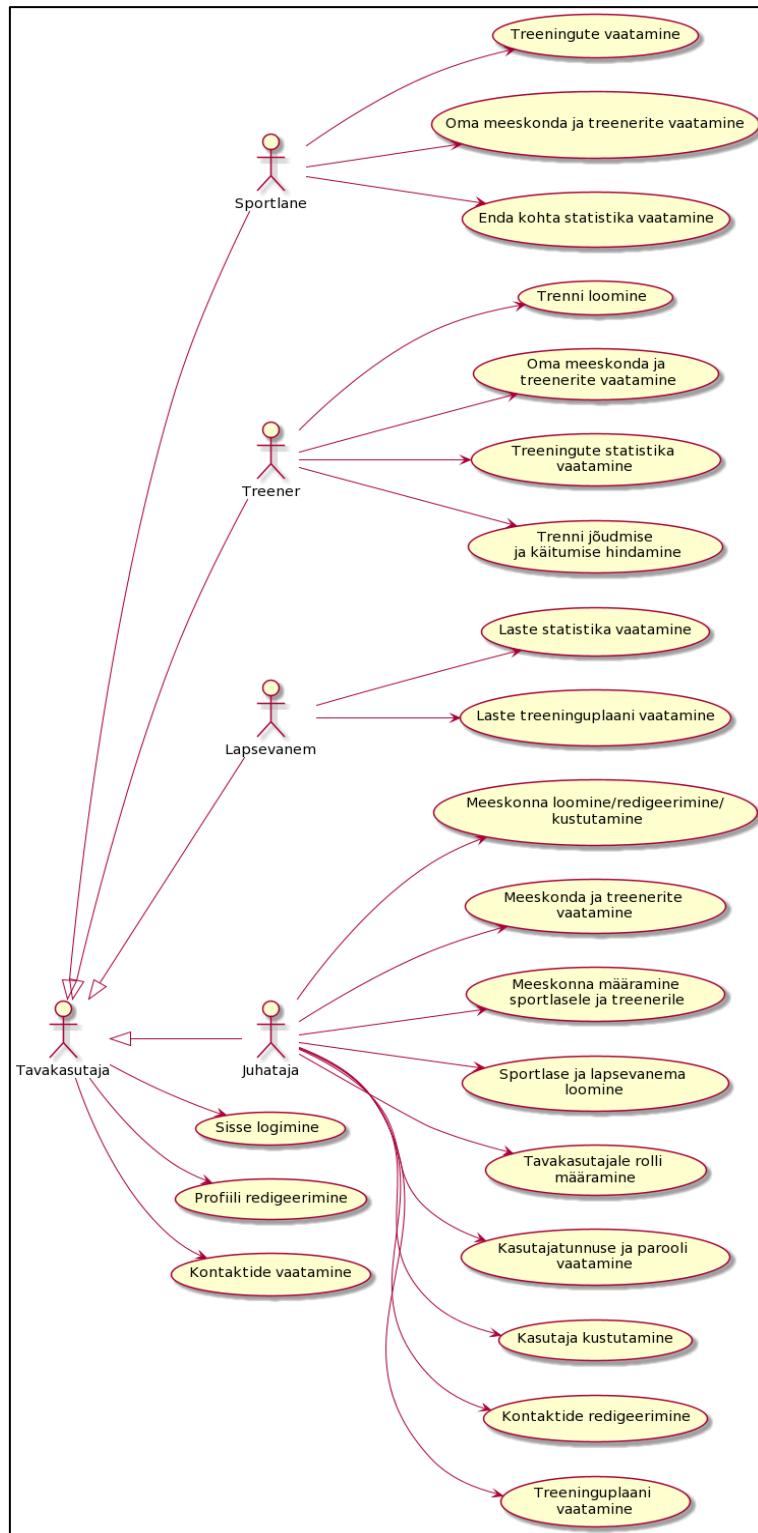




Joonis 1. Ekraani voodiagramm.

Käesoleval juhul kasutas autor UML metoodikat selleks, et kaardistada erinevaid kasutajavaateid. Diagrammi paremaks arusaamiseks kasutati erinevaid värvitoone, millega eraldati erinevaid rolle:

- Treener - roheline.
- Sportlane - roosa.
- Lapsevanem - oranž.
- Juht - kollane.
- Tundmatu kasutaja - sinine või punane.



Joonis 2. UML diagramm.

Diagrammi valmistamiseks kasutati PlantUml veebipõhist tööriista [10]. Lisaks rollide eristamisele määrati vaadete vahel sõltuvused. Antud juhul toetab UML metoodika objektorienteeritud programmeerimist [9].

### 3.3 Mobiilirakendusele esitatavad nõuded

Antud peatükis defineerib autor nõuded arendatavale rakendusele. Nõuded on jaotatud kahte kategooriasse – funktsionaalsed ja mittefunktsionaalsed nõuded. Iga nõuete grupp kirjeldatakse lahti kliendi (kasutaja) ja arendaja (administraatori) tasemetest lähtuvalt. Käesolev jaotus on tehtud selleks, et süstematiseerida arendusprotsess. See tagab tulevikus paremat arendusprotsessi juhtimist ja dokumenteerimist.

Nõuded on määratud vastavalt tellija reaalsele treeninguprotsessile ja äritegevusele. Arvesse võeti klubi erinevaid rolle ja nende vajadusi mobiilirakenduse suhtes. Tegemist on esialgse, lõputöö raames, esitatud funktsionaalsuse kirjeldamisega. Tulevikus kavatses autor viia parandused sisse vastavalt kasutaja tagasisidele.

Funktsionaalsed nõuded kliendi (juhataja) vaatest lähtuvalt:

- Meeskonna loomise, redigeerimise ja kustutamise funktsioon.
- Treenerikoosseisu ja meeskondade ülevaatamise funktsioon.
- Sportlase meeskonda määramise funktsioon.
- Treeneri meeskonda määramise funktsioon.
- Sportlase ja lapsevanema loomise funktsioon.
- Rolli määramise funktsioon.
- Kasutajate konto andmete vaatamise ja redigeerimise funktsioon.
- Kasutaja kustutamise funktsioon.
- Kontaktide muutmise funktsioon.
- Treeningute jälgimise funktsioon.

Funktsionaalsed nõuded kliendi (sportlase) vaatest lähtuvalt:

- Treeningute vaatamise funktsioon.
- Oma tulemuste statistika vaatamise funktsioon.
- Oma grupi ja treenerite vaatamise funktsioon.

Funktsionaalsed nõuded kliendi (treeneri) vaatest lähtuvalt:

- Trenni loomise funktsioon.
- Sportlaste kohal käimise ja käitumise fikseerimise funktsioon.
- Oma meeskonda kuuluvate sportlaste statistika vaatamise funktsioon.

- Oma meeskondade vaatamise funktsioon.
- Oma grupi ja treenerite vaatamise funktsioon.

Funktsionaalsed nõuded kliendi (lapsevanema) vaatest lähtuvalt:

- Oma lapse statistika vaatamise funktsioon.
- Oma lapse treeningute vaatamise funktsioon.

Üldised funktsionaalsed nõuded kliendi vaatest lähtuvalt:

- Sisse ja välja logimise funktsioon.
- Profiilandmete muutmise funktsioon.
- Kontaktide lehekülje vaatamise funktsioon.

Mittefunktsionaalsed nõuded kliendi vaatest lähtuvalt:

- Kliendi andmete turvalisus peab olema tagatud.
- Kõik kliendivaated peavad olema inglise keeles.
- Mobiilirakenduse menüü peab olema kasutajasõbralik ja võimalikult lihtne.
- Kliendirakendus arendatakse Android operatsioonisüsteemi jaoks. Põhjuseks on olukord, kus enamus tellija töötajatest (peaasjalikult treeneritest) kasutab mainitud operatsioonisüsteemi.

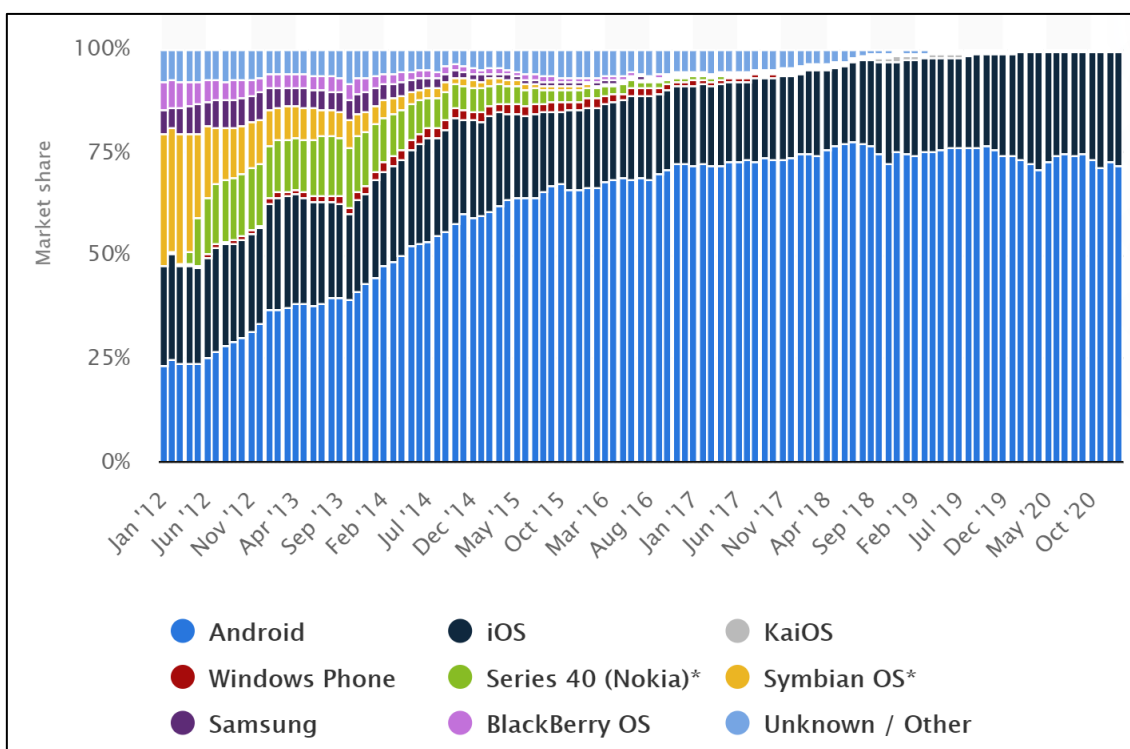
Mittefunktsionaalsed nõuded arendaja või administraatori vaatest lähtuvalt:

- Koodibaas, kommentaarid ja dokumentatsioon peab olema kirjutatud inglise keeles.
- Andmebaas asub tsentraalses serveris. See tagab andmete salvestamist ja taasesitamist juhul kui treeneril (või muul kasutajal) peaks seade rikki minema.
- Koodibaas peab asuma repositooriumis.
- Koodibaas on loogilise ja arusaadava ülesehitusega.
- Lõputöö raames arendatakse ainult defineeritud funktsionaalsused, kuid tulevikus peab jääma võimalus süsteemi võimalikult mugavalt ja kiirelt edasi arendada ja mastabeerida. Sellest tulenevalt defineeritakse kaks alamnõuet:
  - Kliendirakendus peab salvestama ja sisaldama minimaalse koguse andmeid, st kogu informatsioon saadetakse serverist.

- Server tagab REST API teenust ja kasutab HTTP protokollit [11]. Iga kliendi tegevusega kaasneb serveri päring ja seadmemällu kirjutatakse andmeid minimaalselt.
- Tagarakendus peab olema paigaldatav tellija poolt kasutatavas serveris. Hetkel kasutatakse Heroku PaaS teenust [12].

### 3.4 Arendustehnoloogia valik

Rakenduse arendustehnoloogia valikul tuleb lähtuda rakendusele esitatavatest nõuetest. Antud juhul määravaks nõudeks on Android operatsioonisüsteemi valik, mis on tingitud organisatsioonist, mille näitel teeb autor uurimuse. Siinkohal on oluline mainida, et Android operatsioonisüsteemi valik on arukas ka sellepärast, et seda on maailmas kasutuses kõige rohkem. Statista 2021 veebiallika põhjal on näha, et Androidi kasutajaid on maailmas kuni 75% kõikide teiste mobiiliplatvormidega võrreldes (vt joonis 3) [13].



Joonis 3. Maailmaturul kasutatavate mobiiliseadmete operatsioonisüsteemide osakaal. (*ibid.*)

Eeltoodud joonisel on näha vertikaalteljel turuosakaal protsentides *Market share*, horisontaalteljel kuu ja aasta. Joonisel esineb ka legend, kus on värviindeksiga välja toodud erinevad operatsioonisüsteemid: Android, Windows Phone, Samsung, iOS, Series 40 (Nokia), Black Berry OS, KaiOS, Symbian OS ja tundmatu *Unknown/Other*. Symbian

OS ja Series 40 (Nokia) on märgistatud tärniga sellepärast, et neid kas ei eksisteeri enam turul või nende osakaalu on võimatu määrata oluliselt väiksema kasutusosakaalu tõttu.

Seoses sellega, et mobiilirakenduse üks nõue on arendada rakendus Android operatsioonisüsteemi jaoks, otsustas autor kasutada käesoleva rakenduse arendamiseks Kotlin programmeerimiskeelt. Nimetatud keel on tugevalt seotud Android rakenduste arendamisega. Viimane omab suurt populaarsust tänu Java süntaksi sarnasusele. [13]

Kotlin programmeerimiskeele valiku põhjuseid on mitu. Üks peamistest on see, et autoril on varasem arendamiskogemus Kotlin keeles ja uue keele õppimine lõputöö raames ei ole antud töömahu juures aktuaalne. Sellele vaatamata on arukas välja tuua Kotlini põhilised eelised (*ibid.*):

- Suhteliselt suur tootlikus tänu keele eripäradele ja abstraktsioonitasemele, mis aitab vähendada arendaja kirjutatud koodi mahtu ja toetab kiiret ja kvaliteetset arendusprotsessi.
- Heal tasemel realiseeritud turvalisuse funktsioonid aitavad vältida programmeerimisvigude nagu null-kursori erand (inglise keeles *null pointer exception*). Vastavalt Google poolt tehtud uuringule Kotlinit sisaldavad rakendused annavad 20% vähem vigu võrreldes Kotlini alternatiividega.
- Ühilduvus Java keelega annab ühendusvõimaluse Kotlinil ja Javal arendatud rakendustel. St, et olemasolevad Javal arendatud funktsionaalsused saab lisada Kotlini rakendusse ilma kadudeta.
- Kotlin on tõhus keel asünkroonse koodi arendamiseks. Nimelt Kotlin realiseerib taga taustal toimuvate protsesside juhtimist efektiivselt ja kiirelt, mistõttu kulub minimaalselt aega lokaalse mälu kasutamisele ja võrgupõhiste protsessidele.

Lisaks Kotlinile kasutatakse JavaScript keelt. Täpsemalt aga tegemist on Express raamistikuga, mis aitab arendada kiireid ja minimalistlikke veebiteenuseid [14]. Tulenevalt sellest, et üheks nõudeks on andmete hoiustamine serveris, siis on aktuaalne kasutada veebiteenuseid nagu nt rakendusliides. Express võimaldab kiiret ja mugavat API arendust, mille tulemusel saab kliendirakendus suhelda serveris paikneva andmebaasiga. (*Ibid.*)

On oluline välja tuua ka SQL struktuuripäringukeelt. Tegemist on keele tüpoloogiaga, mida saab kasutada rakenduse eriosades andmebaasiga suhtlemiseks. Selle abil realiseeritakse andmete kirjutamine, kustutamine ja pärimine andmebaasist. [15]

### 3.5 Arenduskeskkonna valik

Antud lõputöö raames kasutatakse mitut erinevat keskkonda. Kuna arendusprotsess on suhteliselt mahukas, siis keskkondade valiku paremaks lahti seletamiseks kategoriseeritakse keskkondi kolme erinevasse gruppi: prototüübi, kliendirakenduse ja infrastruktuuripõhised arenduskeskkonnad.

Prototüübi arenduskeskkonna näol on tegemist Figma keskkonnaga. Nimetatud lahenduse põhifookuseks on tarkvara prototüübi arendus klienditasemest lähtuvalt. Figma kasutab võimsat graafilist redaktorit, mille abil saab disainida tuleviku rakenduse erinevaid vaateid (nt menüü, plaanid ja erinevad sündmused). Eriti oluline on see kasutaja psühholoogiast ja kasutajasõbralikkuse aspektist lähtuvalt, kuna Figma võimaldab näha kavandatavat lahendust enne kui see valmis on. [16] St paljud projekteerimisega seotud vead võivad olla avastatud ja ära lahendatud juba algusfaasis. Autor kasutas Figma lahendust mobiilirakenduse vaadete ja üldise rakenduse funktsioneerimise loogika määramiseks ning rakenduse töö simuleerimiseks. Käesoleva rakenduse prototüübi link on esitatud lisa 2.

Kliendirakenduse arenduskeskkonna näol on tegemist populaarse Android Studio keskkonnaga. Nimetatud keskkond on valitud mitmel põhjusel [17]:

- Autoril on varasem kogemus Android Studio kasutamisel.
- Android Studio tagab programmeerimisvõimalust Kotlin keeles.
- Automatiseeritud graafiline redaktor, mille abil on võimalik kujundada kliendirakendust vastavalt määratud nõuetele. Redaktori abil on võimalik hinnata rakenduse vaateid kasutades erinevaid resolutsioone. Peale disaini tegemist genereerib keskkond ise vajalikku koodibaasi graafilise liidese jaoks.
- APK analüsaatori abil on võimalik tuvastada liigset või ebaefektiivset koodi rakendust koodibaasis. Sellest tulenevalt tekib võimalus rakenduse mälu kasutamise (ja ka muude ressursside kasutamise) optimeerimiseks.
- Kasutajasõbralik ja mugav koodi redaktor.

- Android Studio võimaldab arvuti ressursside kasutamise monitooringut.
- Rakenduse emulaator aitab kiirendada arendusprotsessi. Nimelt Android Studio abil on võimalik käivitada rakendust palju kiiremalt kui seda võimaldab füüsiline seade. See annab võimaluse kiiremini läbi proovida erinevaid funktsioone ja hinnata nende mõju rakendusele.

Nagu on eelnevalt mainitud, kasutas autor lisaks kliendirakenduse ja prototüübi arenduskeskkondadele ka infrastruktuuripõhiseid keskkondi. Nimetatud keskkondade kasutamist ja funktsionaalsust antud projekti raames saab defineerida järgnevalt:

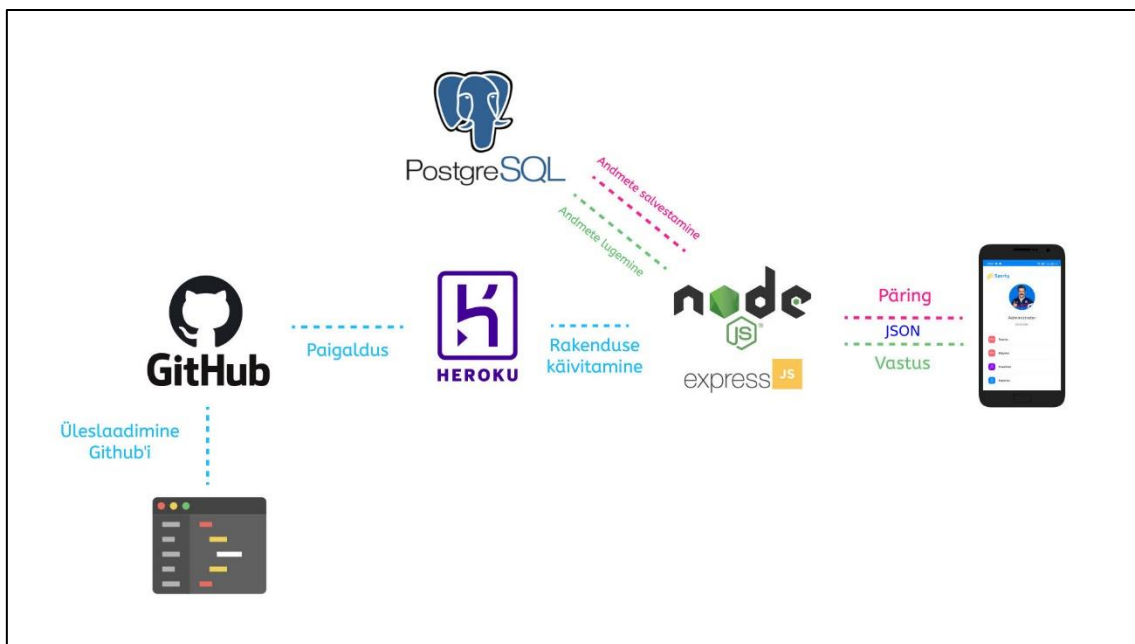
- Node.js keskkonnas käivitatakse JavaScripti Express raamistiku abil genereeritud koodi. Sündmusjuhitav arhitektuur annab võimaluse asünkroonse koodi arendamiseks. St, et andmete sisestamine ja lugemine on kiire ja tõhus [18].
- PostgreSQL keskkond on vahend, mille abil on võimalik juhtida ja hallata relatsioonilisi andmebaase. Kuna enim on viidatud objektorienteeritud programmeerimise kasutamisele, siis relatsioonilise andmebaasi kasutuselevõtt on arukas, kuna andmebaasi tabelid võivad esindada objektide struktuuri [19].
- GitHub ja Gitlab keskkondades on võimalik hoida koodibaasi, säilitades arendamise jaoks vajalikku versioneerimise võimaluse [20]- [21].
- Heroku platvorm on mugav rakenduse paigaldamise vahend. Antud projekti raames on võimalik mugavalt ja kiirelt ühendada GitHub koodibaasi Heroku serveriga [12].
- Firebase on mugav Google poolt pakutud teenus, mille abil on võimalik hoiustada erinevaid graafika elemente. Antud juhul on võimalik Firebase keskkonnas hoiustada rakenduses kasutatavaid pilte ja jooniseid. [22]



## 4 Arhitektuur

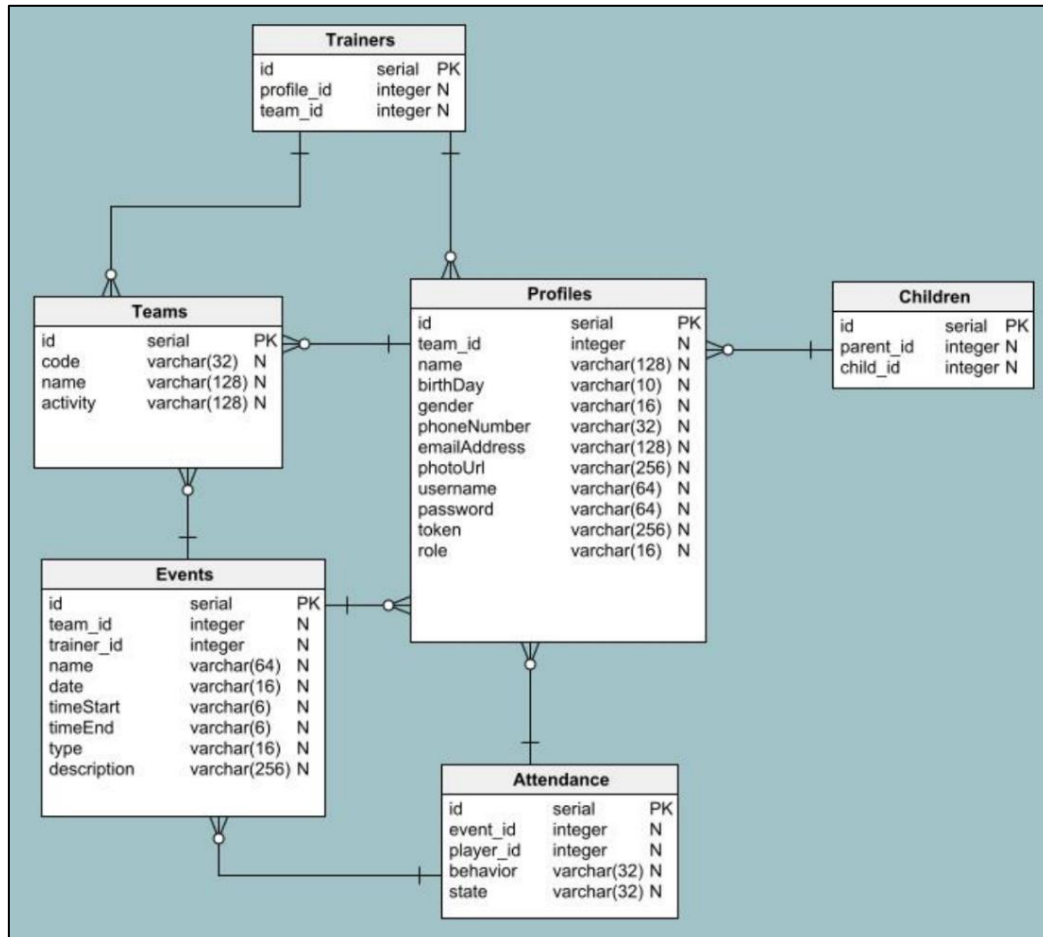
Käesoleva lõputöö raames arendatud mobiilirakendus hõlmab endas mitmeid arhitektuurseid elemente. Selleks, et lahti seletada rakenduse funktsioneerimiseks vajalikke elemente koostas autor joonist 4. Iga elemendi töö ja funktsioon on seotud kogu süsteemi ahelaga ning ühe elemendi eemaldamisega kaasnevad tõrked süsteemis ja funktsionaalsuse kadu. Ahelas on kokku viis põhilist elementi:

- GitHub keskkonnas toimub koodibaasi salvestamine ja versioneerimine.
- Heroku automaatse teenuse abil ühendatakse GitHub repositoorium serveriga. Teenus pakub automaatse paigaldusteenuse *auto deploy*, mis tagab rakenduse taas käivitamise serveris iga kord kui toimuvad muudatused repositooriumis.
- Node.js keskkonnas asub JavaScripti kood, mille põhiliseks ülesandeks on rakendusliidese funktsiooni realiseerimine. Sellest tulenevalt Node.js omab ühendust andmebaasiga, kuhu vajadusel salvestatakse ja kust päritakse rakendusega seotud kasutajaandmeid.
- PostgreSQL keskkonnas asub andmebaas.
- Kasutaja seadmes on installeritakse mobiilirakendus APK faili kasutades.



Joonis 4. Rakenduse arhitektuur.

Kasutajalugude ja kasutajadisaini analüüsidest kujunes välja andmebaasi disain, mis on oluline osa kogu rakenduse arhitektuurist, kuna tagab vajalikke andmete säilitamist ja representeerimist. Olemite lahti kirjutamisel valmis olemi-suhte diagramm (vt joonis 5).



Joonis 5. Olemi-suhte diagramm.

Olemi-suhte diagrammil esineb kuus tabelit: profiili *Profile*, kohal käimise *Attendance*, sündmuse *Events*, meeskonna *Teams*, treenerite *Trainers* ja laste *Children* tabelid. Diagrammil on näha iga tabeli suhet teise tabeliga ja esitatud on ka primaarvõtmed. Igas tabelis on olemas ka andmerekad koos andme tüübiga. Kuna tegemist on relatsioonilise andmebaasi olemi-suhte diagrammiga, siis viimane toetab objektorienteeritud programmeerimise kasutamist programmikoodis ja on aluseks domeenimudelitel koostamisel.

## 4.1 Koodibaasi ülesehitus

Projekti kliendirakenduse arendamiseks kasutati Kotlin programmeerimiskeelt ja kogu rakenduse paketi paigaldatakse APK faili vormingusse. Kliendirakenduse arhitektuur koosneb järgmistest võtmelementidest (kliendirakenduse link on leitav lisas 2):

- APK paketi kokku panemise skriptid.
- Rakendusliidesega töötamiseks vajalikud funktsioonid. Põhielemendiks on HTTP protokoll kasutatav andmevahetuse funktsionaalsus.
- Rakenduse funktsioneerimiseks ja relatsioonilise andmebaasi kasutamiseks loodud domeeni mudelid (inglise keeles *domain models*) kõikide objektide kohta.
- Mobiilirakenduse ärioloogikat sisaldavad teenused (inglise keeles *services*).
- Android Studio keskkonnas kiiremaks programmi testimiseks ja protsesside simuleerimiseks genereeritud *mock* failid.
- Kasutajaliidese failid (kõikide vaadete genereerimiseks).
- Muud konfiguratsiooni failid.

Serveripoolse rakenduse arendamisel koostati JavaScript keeles (Express raamistiku kasutades) skriptid, mis asuvad Node.js keskkonnas. Nimetatud koodibaasi ülesanne on API teenuste tagamine kliendirakenduse (Android mobiilirakenduse) ja PostgreSQL teenuse vahel. Android rakenduse ja Node.js serveri rakenduse vahel toimub andmevahetus Json vormingu alusel, mis lihtsustab ärimudelite kasutamist mõlemal poolel. Skriptide põhielemendid on (täpne koodibaas on leitav lisas 2):

- *Rout* failid, kus on realiseeritud API päringuid töötlev funktsionaalsus. Failid on koostatud kindla struktuuri alusel, st iga objekt kirjeldatakse eraldi failis koos tema suhtes rakendatava API päringuga (nt andmebaasi kirjutamise funktsioon *put* või andmebaasi lugemise funktsioon *get*). Antud lähenemine aitab hoida igat objekti eraldi ja vastavalt vajadusele redigeerida temaga seotud API päringud.
- Domeeni mudelid, mis kirjeldavad objekti struktuuri.
- *Route* ressursi initsieeriv fail.
- PostgreSQL teenuse konfiguratsiooni failid.
- SQL skripte sisaldavad failid.
- Muud konfiguratsiooni failid.

## 4.2 Andmete esitamine kliendirakenduses

Seoses lõputöö mahu piiranguga ei ole võimalik esitada kirjeldused kõikide tehniliste lahenduste kohta. Sellest tulenevalt otsustas autor lahti kirjutada koodibaasis esinevaid kohti, mis on tema arvates kõige keerulisemad. Antud juhul tegemist on kliendirakenduse realiseerimisega, täpsemalt aga dünaamilise statistika vaate genereerimise funktsionaalsusega (vt joonis 6).



The screenshot shows a mobile application interface for 'Attendance' at 'JK Narva Trans A'. The top bar is blue with the time '16:38' and status icons. Below the title, there are filters for '2021' and 'MAY', and a search icon. The main content shows '23 Players' with a percentage indicator and a green checkmark. A list of players follows, each with their name, attendance percentage, and two counts.

Player Name	Attendance %	Count 1	Count 2
Aleksandr Zaka	90%	6	0
Aleksei Stepanov	50%	2	1
Arseni Kovaltšuk	90%	6	1
Artjom Škinjov	50%	2	0
Bi Sehi Elysee Irie	90%	6	0
Daaniel Maanas	100%	7	0
Denis Poliakov	80%	5	1
Eduard Golovljov	100%	7	0
Gleb Pevtsov	90%	6	0

Joonis 6. Kohal käimise statistika vaade.

Eeltoodud joonise abil on võimalik vaadata sportlaste statistikat kohal käimise kohta. Nimetatud statistikat genereeritakse kahes etapis. Kõigepealt päritakse andmebaasis struktureeritud andmed kujul: sportlaste arv, kohal käimise osakaal protsentides, kohal käimiste, mitte ilmumiste ja hilinemiste arv. Nimetatud andmeid sisestatakse funktsiooni *buildInfoLayout()* (vt joonis 7).

```
private fun buildInfoLayout(dates: List<Int>, count: Int) {
    binding.playersCount.text = "$count " + getString(R.string.players)
    binding.infoLayout.removeAllViewsInLayout()

    for (date in dates) {
        val dateView = inflater.inflate(R.layout.view_date,
            binding.infoLayout, false) as TextView
        dateView.text = date.toString()
        binding.infoLayout.addView(dateView)
    }
}
```

Joonis 7. Andmete töötlemine statistika vaate genereerimiseks.

Järgmiselt toimub vaate vertikaalandemete genereerimine kasutades *RecyclerView* ja *ViewHolder* abil (vt joonis 8). Sellest tulenevalt tekib vaade, kus on esitatud vajalikud statistika koondandmed sportlase kohta.

```

class AttendanceHolder(view: View) : Holder<PlayerAttendance, Empty>(view) {
    override fun bind(value: PlayerAttendance, callback: Empty) {
        val name = itemView.findViewById<TextView>(R.id.name)
        val attendedCount = itemView.findViewById<TextView>
            (R.id.attendedCount)
        val lateCount = itemView.findViewById<TextView>(R.id.lateCount)
        val excusedCount =
            itemView.findViewById<TextView>(R.id.excusedCount)
        val percent = itemView.findViewById<TextView>(R.id.percent)
        val layout = itemView.findViewById<LinearLayout>
            (R.id.attendanceLayout)
        name.text = value.name
        attendedCount.text = value.attendedCount.toString()
        lateCount.text = value.lateCount.toString()
        excusedCount.text = value.excusedCount.toString()
        percent.text = value.percent.toString() + "%"
        layout.removeAllViewsInLayout()

        for (state in value.states) {
            val imageView = LayoutInflater.from(itemView.context)
                .inflate(R.layout.view_attendance, layout, false)
                as ImageView
            if (state == AttendanceState.NONE) {
                imageView.setImageResource(R.drawable.ic_none)
            }
            else {
                imageView.setImageResource(Attendance
                    .fromState(state).icon)
            }
            layout.addView(imageView)
        }
    }
}

```

Joonis 8. Statistilise vaate vertikaalandmete genereerimine.

### 4.3 Kasutajaliides

Mobiilirakenduses realiseeritud kasutajadisain ja -liides põhineb varem kirjeldatud ekraani voodiagrammil ja kasutuslugudel. Lisaks kasutati Figma keskkonnas valmistatud prototüübi tulemusi. Kuna autor kasutas arenduseks Android Studio keskkonda, siis liidese disaini realiseerimine programmi koodis oli suhteliselt mugav.

Käesolevas alapeatükis kirjeldatakse kasutajaliidese lahenduse põhimõtteid ning tuuakse näited kasutajaliidese vaadetest. Alapeatüki raames kasutatakse vaid mõned (kõige aktuaalsemad) vaated, illustreerimaks erinevate rollide kasutajadisaini ja arendatud funktsionaalsust. Kõikide vaadete töösse lisamine ei ole arukas, kuna neid on piisavalt

palju ja vajadusel on alati võimalik neid vaadelda otse vabavaralises Android rakenduses (link APK pakatile asub lisas 2).

Esimene tähtis vaade on kasutaja registreerimise vaade (vt joonis 9). Nimetatud vaade tuleb ette igal kasutajal. Peale seda on võimalik kasutajal määrata roll või muuta andmeid.

The image shows two side-by-side screenshots of an Android application. The left screenshot is the 'Sign up' screen, featuring a blue header with the text 'Sign up'. Below the header are four text input fields: 'Username', 'Password', 'Confirm password', and 'Your name'. At the bottom of the screen are two buttons: a blue 'Sign up' button and a grey 'Back' button. The right screenshot is the 'Profile' screen, featuring a blue header with the text 'Profile'. Below the header is a circular profile picture of a man with a beard and glasses. To the right of the profile picture are several text input fields: 'Name' (containing 'Administrator'), 'Birthday' (containing '20.06.1979'), 'Gender' (containing 'Male'), 'Phone Number' (containing '+5555 5555'), and 'Email Address' (containing 'admin@admin.ee'). At the bottom of the screen is a blue 'Save Changes' button.

Joonis 9. Kasutaja registreerimise (vasakul) ja profiili (paremal) vaade.

Selleks tuleb registreerimise vaates *Sign up* ära täita kasutaja tunnus *Username*, parool *Password*, kinnitada parool *Confirm Password*, sisestada nimi *Your name* ja vajutada registreerimisnupule *Sign up*. Vaates on võimalik navigeerida tagasi vajutades nupule *Back*. Andmete muutmine toimub aknas *Profile*.

Profiili muutmise vaates on võimalik redigeerida järgmised konto andmed: nimi *Name*, sünniaeg *Birthday*, sugu *Gender*, telefoninumber *Phone Number* ja emaili aadress *Email Address*. Muudetud andmeid saab salvestada vajutades nupule *Save Changes*.

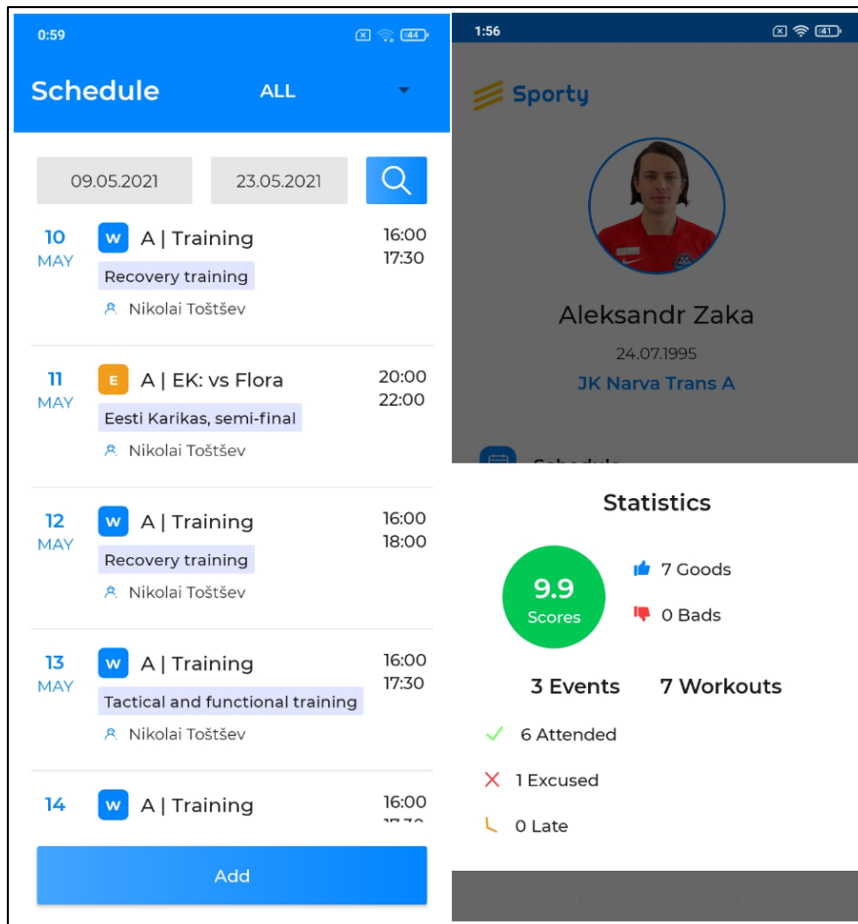
Andmete muutmise funktsionaalsus on samuti lisatud igale kasutajale olenemata rollist. Oluline on mainida, et sama vaade on ka administraatoril iga kasutaja andmete kohta.

Hetkeseisuga valikus on viis elementi kasutajaandmetes: nimi, sugu, telefoni number, meiliaadress ja sünniaeg.

Järgmine oluline funktsionaalsus on trenniplaani jälgimine (*Schedule*, vt joonis 10). Selline vaade on tagatud sportlasel, lapsevanemal, juhatajal ja treeneril. Treeninguplaani uurimisel on võimalik näha erinevaid andmeid: treeningu või sündmuse kuupäev (nt *10 May* ehk kümnes mai), treeneri nimi (nt *Nikolai Toštšev*), kellaaeg, meeskonna indikaator (nt *A*), sündmuse *E (Event)* või trenni *W (Workout)* lühend, trenni puhul on näha kommentaar *Training* ja sündmuse korral osalejate meeskonnanimed (nt *EK vs Flora*). Lisaks on võimalik otsida trenne ja sündmusi kasutades luubiga nupu, kasutada sorteerimist (nt *All* ehk kuvada kõik trennid ja sündmused) ja lisada trenn või sündmus vajutades *Add* nupu.

Lisaks eelnevale, igale sportlasele, lapsevanemale ja treenerile on kättesaadav sportlase statistika vaade *Statistics* (vt joonis 10). Sellest tulenevalt on võimalik saada koondandmeid soorituste kohta ja teha järeldusi trennis käimise edukuse kohta. Nimetatud andmed hõlmavad endas koondtulemust *Scores*, koondnumbrit õnnestunud *Goods* ja ebaõnnestunud *Bads* treeningute kohta ning üldise sektsiooni külastamise statistikat: kohal käimised *Attended*, vabastused *Excused*, hilinemised *Late*, sündmuste arvu *Events* ja treeningute koguarv *Workouts*).

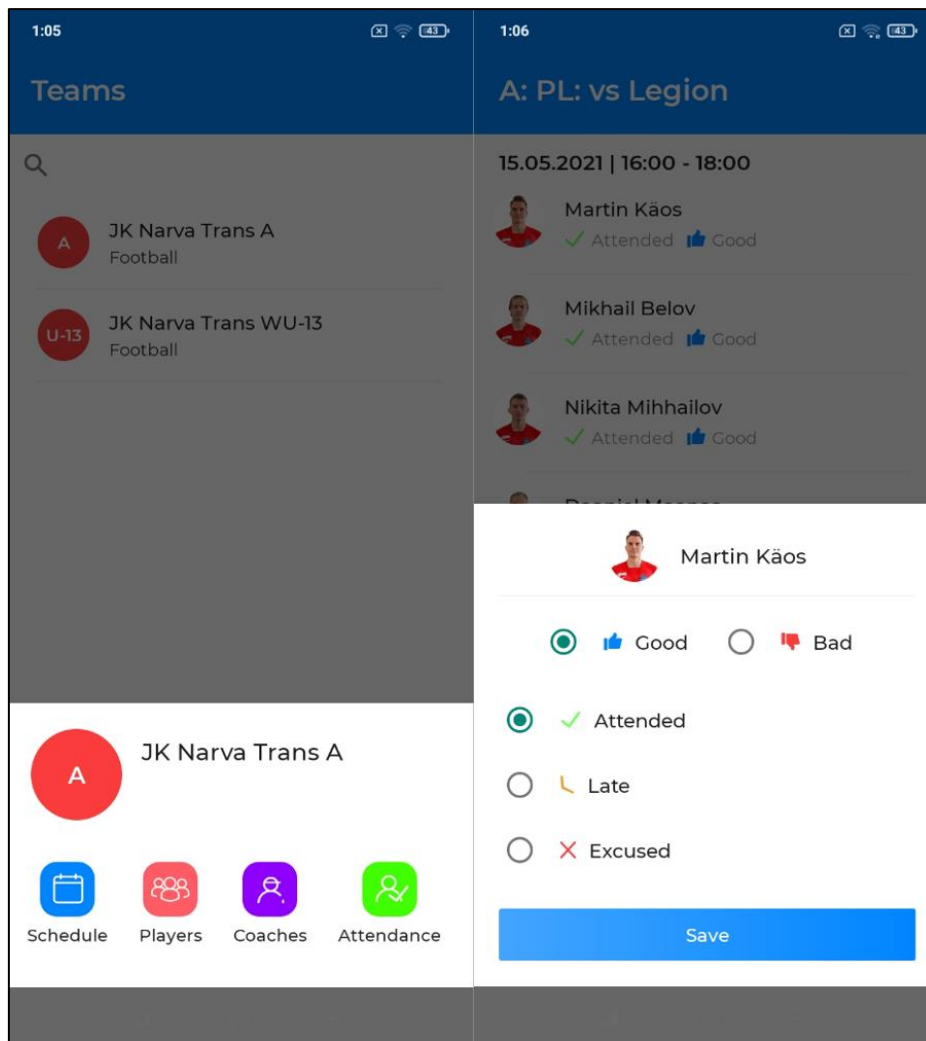




Joonis 10. Treeninguplaani (vasakul) ja kasutaja statistika (paremal) vaade.

Joonisel 11 on esitatud treeneri spetsiifilised vaated, nimelt tegemist on kõige sagedamini kasutatavate treeneri vaadetega. Treenerile on nähtav meeskonna detailne vaade *Teams*, kus on võimalik uurida meeskonna treeninguplaani *Scchedule*, koosseisu *Players*, treenereid *Coaches*, spordiala (nt *Football*), meeskonna indikaator (nt *A* või *WU-13*) koos täisnimega (nt *JK Narva Trans*) ja kohal käimise hinnangut *Attendance*. Treener saab vajadusel redigeerida oma meeskonna andmeid.

Nimetatud joonisel on näha ka sportlase hindamise funktsionaalsust. Selline vaade on kättesaadav ainult treenerile ja on aluseks sportlase tulemuslikkuse hindamiseks. Selleks on vaja nime järgi välja valida sportlane. Hinnata saab üldist sooritust *Good* või *Bad* ja treeningusse ilmumise iseloomu - kas sportlane on hilinenud *Late*, trennist vabastatud *Excused* või õigel ajal ilmunud *Attended*. Lõpus on vaja salvestada andmed vajutades nupule *Save*.



Joonis 11. Meeskonna detailide (vasakul) ja klubiliikme hindamise (paremal) vaade.

Kasutajaliidese juures on võimalik tähele panna, et kogu rakenduse disain on tehtud ühes stiilis, kasutades ühtset vormingut, fonti ja värve. Disaini projekteerimisel soovis autor teha loogilist ja võimalikult intuitiivse kasutajadisaini.

#### 4.4 Testimine

Käesoleva lõputöö raames valminud mobiilirakendust testiti kolme testi tüübiga. Esiteks, viidi läbi simulatsiooni testid. Selleks kasutas autor arenduskeskkonna vahendeid. Simulatsiooni testiks olid kirjutatud eraldi *mock* failid, mille abil süsteem käivitas rakendust ilma serveri abita. Sellel testimisel on mitu eelist:

- Autoril on võimalik vaadata, kui suurt koormust avaldab rakendus süsteemile (nt protsessori- ja mälu kasutus erinevates rakenduse protsessides). See on aluseks

programmi spetsifikatsiooni kirjeldamiseks ja kasutaja seadme võimete määramisel.

- Simulatsioon aitab säästa arendaja aega, nimelt autor ei pea tegelema serveri seadistamise või mobiiliseadme haldamisega. Kogu funktsionaalsus testitakse ühes kohas ja minimaalse ressursiga.
- Graafiline protsessor käivitab rakendust ja on võimalik käsitsi läbi proovida kõik rakenduse funktsioonid. See on eriti tõhus vahend analüüsima süsteemi osade funktsionaalsust ja selle toimimist.
- Analüsaatori abil on võimalik tuvastada nn üleliigset ja ebatõhusat koodi, mida on võimalik muuta või üldse eemaldada. See tagab mälu kasutuse optimeerimist ja lihtsustab koodibaasi loetavust. Pärast simulatsiooni on võimalik kiiresti koodibaasi muuta ja koheselt käivitada uus simulatsioonitest.

Peale simulatsioonitesti viis autor läbi kasutaja testid (UAT) Narva Trans meeskonna tegevuste tingimustes. Kokku kaasati kaks gruppi: täiskasvanute meeskond ja laste sektsioon, kuhu on kaasatud ka lapsevanemad.

Kasutajatesti eesmärk on testida programmi tööd reaalses tingimustes. Nimelt kasutati täisfunktsionaalsust igapäevase treeningukoormuse juures. UAT tagajärjel saadud tagasiside analüüsi ja tehti süsteemis muudatused. Kokku on tehtud 17 programmi versiooni koos uuendustega. Kasutaja testi ajal sisestatud kasutajate andmeid on võimalik näha rakenduse alla laadides, sisenedes administraatorina (kasutajatunnus *admin*, parool *admin*) ja nupule *all users* vajutades.

Selleks, et kirjeldada UAT raames avaldatud koormust programmile, on arukas defineerida testikeskkonna stsenaariumeid. Nagu on eelnevalt mainitud, üks tegevuste stsenaarium on täiskasvanute jalgpalli meeskonna igapäevategevus:

- Meeskonnas on 23 sportlast.
- Kaks treenerit.
- Treeningute protsess on uuritud ajavahemikul 01.05.2021 - 15.05.2021.
- Iga trenn on rakenduse poolt fikseeritud, kirjeldatud ja hinnatud.
- Iga sportlase tegevus on hinnatud (resultatiivsus ja kohale ilmumine), millest hiljem oli võimalik näha statistilisi koondandmeid.

Lisaks täiskasvanute grupile testiti läbi ka laste meeskonda. Nimetatud testi tingimused olid järgmised:

- Tegemist on segameeskonnaga, kuhu kuuluvad nii poisid kui tüdrukud.
- Meeskonnas on üks treener.
- Grupis on 12 sportlast.
- Treeningute toimumise aeg 01.05.2021 - 15.05.2021.
- Neljal lapsel on sisestatud lapsevanem.
- Ühel lapsevanemal käis trennis korraga kaks last, teistel vanematel on üks laps.

Kokkuvõtteks võib mainida, et testimine läks edukalt ja ilma tõrgeteta. Iga rakenduse kasutaja sai programmi funktsionaalsusest aru ja oskas seda kasutada. Programm töötas kõikides mobiiliseadmetes ja iga kasutaja sai kasutada täisfunktsionaalsust. UAT ajal välja tulnud kitsaskohad ja kasutajapoolsed parandusettepanekud fikseeriti ning hiljem viidi koodis parandused sisse.

Lisaks kasutaja- ja simulatsioonitestidele kirjutati ka ühiktestid. Nimetatud testide abil kontrolliti peamiselt rakenduse arhitektuursete elementide omavahelist funktsioneerimist. Ühiktestide põhirõhk seati serveriga seotud ühenduste ja funktsioonide testimisele. Näiteks võib tuua autoriseerimise ja verifitseerimise testi (vt joonis 12).

```
class AuthServiceImplTest : TestCase() {
    private val client = HttpClient()
    private val authService = AuthServiceImpl(client)
    private val account = Account("admin", "admin")
    private val signUp = SignUp("neolab", "neopass", "NeoLab",
        SimpleDate(2001, 4, 1).toString())

    fun testLogin() {
        val token = authService.login(account)
        println(token)
    }

    fun testSignUp() {
        val accounts = (1..3).map { signUp.copy(username =
            signUp.username + it, name = signUp.name + it)}
        accounts.forEach {
            val token = authService.signUp(it)
            println(token)
        }
    }
}
```

Joonis 12. Autoriseerimise ja verifitseerimise ühiktest.

Eeltoodud koodis realiseeritakse HTTP klient ja luuakse autoriseerimisteenus. Mõlemad objektid arendatakse keeletehnoloogia sisseehitatud funktsioonide abil - *HttpClient()* ja *AuthServiceImpl(client)*. Peale seda luuakse kaks objekti. Esimene neist (*Account*) hoiab sees kasutaja andmeid (kasutajatunnuse ja parooliga *admin*) ja teine (*SignUp*) sisaldab autoriseerimisteenusele vajaliku konfiguratsiooni (nt *neopass* parooli valideerimise säte ja kuupäev sõne kujul).

Edaspidi luuakse funktsioon *testLogin()*, kus luuakse *authService.login(account)* sisseehitatud klassi abil identsustõend *token*. Meetod saadab välja serverile kasutaja andmed (*account*) ja genereerib vastavalt identsustõendit. Kui nimetatud kasutaja eksisteerib andmebaasis, siis printitakse välja identsustõend. Kui aga kasutajat ei eksisteeri, siis printitakse välja tühi rida.

Analoogselt toimib ka *testSignUp()* funktsioon, kuid selle erinevusega, et testitakse läbi mitme kasutajaga registreerimist. Kasutajaandmed genereeritakse automaatselt iteratsiooni abil. Kahe viimase funktsiooni tulemusi hinnatakse visuaalselt ja selle alusel tehakse järeldusi funktsioonide töö kohta.

## 5 Tulemused ja järeldused

### 5.1 Realiseeritud funktsionaalsus

Käesoleva uurimistöö lõpp-produktina saadi avatud lähtekoodiga mobiilirakendus, mis omab piisava funktsionaalsuse sportlase arengu juhtimiseks. Kõik projekti osad, k.a erinevate arhitektuuri osade lähtekood, on avalik ja tasuta kasutamiseks. Arendatud rakendus on kasulik spordiasutusele, sportlasele, treenerile ja lapsevanemale, kelle laps käib spordisektsioonis.

Mobiilirakendus töötab Android operatsioonisüsteemil ja selle kasutajaliides arendati koostöös reaalse spordiasutusega. Lahendus annab treenerile võimaluse salvestada sportlase treeningutulemusi omas mobiiliseadmes. Sellega vähendatakse paberikasutust, st antud lahendus aitab säästa keskkonda. Treener, kes varem kasutas sülearvutit selleks, et sisestada sportlaste sooritused, ei pea enam sülearvutit kaasas kandma.

Lõputöö projekti raames täideti 23 funktsionaalset ja 12 mittefunktsionaalset nõuet (kokku 35), viie erineva rolli lõikes. Funktsionaalsust on võimalik edasi arendada, kuna kasutati aktuaalseid arendustehnoloogiaid ja mastabeerimiseks sobivat arhitektuuri. Koodibaasis kasutatud meetodite, funktsioonide, parameetrite ja muude elementide nimetusi kirjutati inglise keeles, mis tagab koodist paremat arusaamist ja loetavust.

Kaasaegne tehnoloogia ja andmete salvestamise funktsionaalsus aitavad tagada programmi turvalisust ja veakindlust. Lisaks eelnevale suureneb lahenduse vastu huvi mitte ainult kasutajate, vaid ka arendajate poolt. Viimane võib olla nii kommertsettevõtte kui ka hobiarendaja.

Tarkvara on testitud kasutades arenduskeskkonna simulaatoreid ja koormusjälgimise vahendeid. Lisaks testiti mobiilirakendust reaalse kliendi juures, sisestades terve jalgpalli meeskonna andmeid ning katsetades kõiki rakenduses olevaid funktsioone. Mobiilirakendus on funktsionaalne ja sobiv trenniprotsessi juhtimiseks.

## 5.2 Järeldused ja ettepanekud

Lõputöö raames arendatud projekti võib lugeda õnnestunuks, kuna töö eesmärk ja uurimuses käsitletud probleem on lahendatud. Kõik püstitatud ülesanded ja nõuded on täidetud. Rakendus on testitud ja asub avalikus repositooriumis, kuhu pääseb ligi iga soovija. Mobiilirakenduse kasutajal on tarvis ainult internetiühendust ja Android seadet, et installeerida ja uurida rakenduse tööd. Seoses töö õnnestumisega teeb autor järgmised ettepanekud:

- Kuna mobiilirakendus on testitud ainult ühe reaalse spordisektsiooni näitel, siis arukas on viia läbi testimine ka teistes spordiasutustes või sektsioonides. See annab parema ülevaate eksisteerivatest kitsaskohtadest ja suunab arendajat tegema õiged parandused või täiendused. Programmi kasutusvaldkond peab olema maksimaalselt lai ja sobima erinevatele spordisektsioonidele, see on ka kindel põhjus, miks on mõistlik testida seda teiste asutuste näitel. Kusjuures viimased ei pea tegutsema tingimata Eestis, vastupidi, mida rohkem kasutatakse tarkvara erinevates praktikates, seda mitmekesisem tuleb tagasiside.
- Antud töö kirjutatakse eelkõige akadeemilisel eesmärgil. Sellest tulenevalt teeb autor ettepaneku reklaamida töö teistele, spordiga tegelevatele, haridusasutustele. Tegemist on aktuaalse võimalusega arendada koolile õpilaste soorituste juhtimise tarkvara vastavalt enda soovidele ja vajadustele.
- Koostada ühtne ja avalik tagasisidevorm, mille abil saavad anda tagasisidet mobiilirakenduse kohta kõik asutused ja erakasutajad. Kõigile avalik tagasiside formuleerib aktuaalseid probleeme ja aitab teistel kasutajatel paremini aru saada rakenduse sihtgrupist.

### 5.3 Edasiarendus

Lõputöös saavutatud tulemusi analüüsidest tekkis autoril mitu ideed, kuidas saaks mobiilirakendust veelgi paremaks ja atraktiivsemaks teha. Kusjuures edasiarendus ei ole seotud tingimata ainult funktsionaalsuse arendamisega. Ka mitmed mittefunktsionaalsed aspektid on olulised tarkvara edasiarendamisel. Käesolevaga esitab autor võimalikud edasiarendused projekti osas:

- Arendada rohkem ühikteste, mille abil võib tekitada erinevaid väärtusi. Esiteks, ühiktestide olemasolu võimaldavad oluliselt kiiremini tuvastada vigu ja neid parandada. Teiseks, ühiktestide abil saab uus arendaja paremini aru meetodite ja funktsioonide tööst.
- Teiste keelte kasutamine programmi vaadetes potentsiaalselt laiendab selle kasutajaskonda. Antud lõputöö raames keskenduti Narva Trans jalgpalliklubi vajadustele, kuid tulevikus võib tekkida soov kasutada programmi ka teiste klubide ja organisatsioonide poolt. Rakenduse mitmekeelsus aitab kaasa populaarsuse kiiremale saavutamisele.
- Kuna mobiilirakenduses realiseeriti API liidest, siis antud tehnoloogia abil saab realiseerida ka veebirakenduse versioon. Võib juhtuda, et tulevikus tekkivad kliendid, kes soovivad näha rakenduse veebilehitsejas. Antud funktsioon toob kaasa uusi kliente.
- Kliendibaasi suurenemisel on oluline tagada andmete kopeerimine ettearvamatute olukordade vältimiseks, nagu nt rike serveritöös või teenuseosutaja soov peatada teenuse osutamist.
- Võib kaaluda täiendavate turvalisuse tehnoloogiate kasutamist. Lisaks eelnevale on arukas realiseerida juba olemasolevas koodibaasis erinevaid turvaelemente, nagu nt kasutaja parooli keerukuse hindamine.
- Lisada juturuumide kasutamise funktsionaalsus. Tegemist on juturuumiga, kus treenerid saaksid vahetada sõnumeid meeskondade, sportlaste ja lapsevanemate vahel. Tsentraalse infokanali olemasolu oluliselt parandab informatsiooni adekvaatsust, kättesaadavust ja õigeaegsust.



## 6 Kokkuvõte

Antud uurimistöö eesmärk on arendada vabavaralist Android mobiilirakendust, mille abil on võimalik juhtida sportlase treeninguprotsessi ja edukust. Rakenduse põhifookuseks on keskkonna loomine, kus iga sportlane (lapse puhul ka tema vanem) saaks jälgida treeningute statistikat ja sooritusi. Lahendus aitab treeneritel ja spordiasutustel säästa ajalist ja tehnilist ressursi, tagades mugavat ja kiiret kasutajaliidest spordiasutuse sündmuste planeerimisel (nt trennide organiseerimine, meeskondade kokku panemine, treenerite või sportlaste määramine meeskondadesse).

Lõputöö eesmärgi saavutamiseks analüüsiti olemasolevaid lahendusi ja sellest lähtuvalt määrati süsteemile 35 nõuet. Lisaks eelnevale uuriti analüüsi käigus erinevaid arendustehnoloogiaid ja keskkondi. Analüüsi skoopi kuulus ka kasutajadisaini puudutava dokumentatsiooni uurimine. Uurimuse raames tehtud analüüsi tulemused andsid ülevaate mobiilirakenduse arhitektuurist, selle elementidest ja turvalisusest.

Autor saavutas lõputöös defineeritud eesmärgi, kuna kõik eeltoodud nõuded olid mobiilirakenduses realiseeritud ja rakenduse kasutatavus on kinnitatud reaalse spordiasutuse poolt. Arenduseks kasutati kaasaegset ja vabavaralist tehnoloogiat, mille abil disainiti versioneerimist ja mastabeerimist võimaldavat rakenduse arhitektuuri. Programmitöö paremaks analüüsiks ja paranduste tegemiseks viis autor läbi simulatsiooni, ühik- ja kasutaja testid 37 kasutajaga. Projekti koodibaas on kättesaadav avaliku domeeni kaudu ja mobiilirakendust on võimalik igal Androidi kasutajal tasuta alla laadida.

## Kasutatud kirjandus

- [1] Microsoft, „Microsoft 365,“ [Võrgumaterjal]. Available: <https://www.microsoft.com/et-ee/microsoft-365>. [Kasutatud 14.04.2021].
- [2] Ihor's Universal Apps, "Журнал тренера," [Online]. Available: <https://play.google.com/store/apps/details?id=com.phoenixria.coachjournal&hl=r u&gl=US>. [Accessed 12.04.2021].
- [3] Sportlyzer LLC, „Team management software that helps you assign and track athletes' homework during the COVID-19 outbreak,“ [Võrgumaterjal]. Available: <https://www.sportlyzer.com/ru>. [Kasutatud 12.04.2021].
- [4] B. Veit, „Coaching Journal Server,“ [Võrgumaterjal]. Available: <https://github.com/bjoernveit/Coaching-Journal-Server>. [Kasutatud 12.04.2021].
- [5] L. Õunapuu, Kvalitatiivne ja kvantitatiivne uurimisviis sotsiaalteaduses, Tartu: Tartu Ülikooli Kirjastus, 2014.
- [6] JK Narva Trans, „Narva Trans 1979,“ [Võrgumaterjal]. Available: <https://www.fctrans.ee>. [Kasutatud 12.04.2021].
- [7] I. Jacobson, Object-oriented software engineering : a use case driven approach, Boston: Addison-Wesley Publishing Company, Inc, 1992.
- [8] J. Rumbaugh, Object-oriented modeling and design. Prentice-Hall, Hoboken: Prentice Hall, 1991.
- [9] D. Rosenberg, Use Case Driven Object Modeling with UML: A Practical Approach, Boston: Addison-Wesley Professional, 1999.
- [10] PlantUML team, „PlantUML,“ [Võrgumaterjal]. Available: <http://www.plantuml.com/plantuml/uml>. [Kasutatud 15.04.2021].
- [11] R. T. Fielding, „Representational State Transfer (REST),“ [Võrgumaterjal]. Available: [https://www.ics.uci.edu/~fielding/pubs/dissertation/rest\\_arch\\_style.htm](https://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm). [Kasutatud 13.04.2021].
- [12] Salesforce, "What is Heroku," [Online]. Available: <https://www.heroku.com/what>. [Accessed 15.04.2021].
- [13] Statista, „Mobile operating systems' market share worldwide from January 2012 to January 2021,“ [Võrgumaterjal]. Available: <https://www.statista.com/statistics/272698/global-market-share-held-by-mobile-operating-systems-since-2009/>. [Kasutatud 18.05.2021].
- [14] Google, „Android's Kotlin-first approach,“ [Võrgumaterjal]. Available: <https://developer.android.com/kotlin/first>. [Kasutatud 15.04.2021].
- [15] OpenJS Foundation, „Express,“ [Võrgumaterjal]. Available: <https://expressjs.com/>. [Kasutatud 15.04.2021].
- [16] InfoWorld, "What is SQL? The lingua franca of data analysis," [Online]. Available: <https://www.infoworld.com/article/3219795/what-is-sql-the-lingua-franca-of-data-analysis.html>. [Accessed 16.04.2021].

- [17] Figma, „Minds meeting minds is how great ideas meet the world,“ [Võrgumaterjal]. Available: <https://www.figma.com/>. [Kasutatud 16.04.2021].
- [18] Google, „Android Studio,“ [Võrgumaterjal]. Available: <https://developer.android.com/studio>. [Kasutatud 16.04.2021].
- [19] OpenJS Foundation, „Node.js,“ [Võrgumaterjal]. Available: <https://nodejs.org/en/>. [Kasutatud 15.04.2021].
- [20] The PostgreSQL Global Development Group, „PostgreSQL,“ [Võrgumaterjal]. Available: <https://www.postgresql.org/>. [Kasutatud 16.04.2021].
- [21] GitHub Inc, „GitHub,“ [Võrgumaterjal]. Available: <https://github.com/>. [Kasutatud 15.04.2021].
- [22] Gitlab Inc, „GitLab is the open DevOps platform,“ [Võrgumaterjal]. Available: <https://about.gitlab.com/>. [Kasutatud 10.05.2021].
- [23] Google, „Firebase helps you build and run successful apps,“ [Võrgumaterjal]. Available: <https://firebase.google.com/>. [Kasutatud 15.05.2021].
- [24] Troline, "Система учёта и автоматизации для спортивных организаций," [Online]. Available: <https://tronline.ru>. [Accessed 12.04.2021].
- [25] V. Romanenko, „Журнал тренера. Учет и анализ работы тренера,“ [Võrgumaterjal]. Available: <https://apps.apple.com/ru/app/журнал-тренера/id1415326996>. [Kasutatud 11.04.2021].
- [26] Salesforce, „Heroku,“ [Võrgumaterjal]. Available: <https://www.heroku.com/>. [Kasutatud 15.04.2021].

## **Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks<sup>1</sup>**

Mina, Konstantin Burdakov

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose "Sportlaste arengu monitoorimise ja hindamise mobiilirakenduse arendamine Android baasil", mille juhendaja on Jekaterina Tšukrejeva.
  - 1.1. reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
  - 1.2. üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. Olen teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.
3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

20.05.2021

---

<sup>1</sup> Lihtlitsents ei kehti juurdepääsupiirangu kehtivuse ajal vastavalt üliõpilase taotlusele lõputööle juurdepääsupiirangu kehtestamiseks, mis on allkirjastatud teaduskonna dekaani poolt, välja arvatud ülikooli õigus lõputööd reprodutseerida üksnes säilitamise eesmärgil. Kui lõputöö on loonud kaks või enam isikut oma ühise loomingulise tegevusega ning lõputöö kaas- või ühisautor(id) ei ole andnud lõputööd kaitsvale üliõpilasele kindlaksmääratud tähtjaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktidele 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.

## Lisa 2 – Ressursid

- Prototüübi arenduskeskkond Figma (viide projekteerimisfailile):  
<https://www.figma.com/file/jzleWDMxe9BTc76PjwX1zM/Sporty?node-id=0%3A1>
- Prototüübi arenduskeskkond Figma (viide käivitatud prototüübile):  
<https://www.figma.com/proto/jzleWDMxe9BTc76PjwX1zM/Sporty?node-id=1%3A2&scaling=scale-down&page-id=0%3A1>
- Kliendirakenduse koodibaas:  
<https://gitlab.cs.ttu.ee/koburd/sporty>
- Node.js JavaScripti koodibaas:  
[https://github.com/mrdreamyland/sporty\\_api/](https://github.com/mrdreamyland/sporty_api/)
- Heroku teenus:  
<http://sporty-api.herokuapp.com>
- APK paketti asukoht:  
<https://drive.google.com/drive/folders/1ZXUNq4BooIUG9mPAYqzYKVtJXJ9sFWX2?usp=sharing>