

TALLINNA TEHNIKAÜLIKOOL  
Infotehnoloogia teaduskond

Henry Veetamm 185486IADB

**Reaalajaline punktiseisu ja statistika  
veebirakenduse prototüüp  
võrkpallivõistkondadele**

Bakalaureusetöö

Juhendaja: Meelis Antoi  
Magistrikraad

Tallinn 2023

## **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: Henry Veetamm

15.05.2023

## **Annotatsioon**

Käesoleva bakalaureusetöö eesmärk on luua veebirakenduse prototüüp võrkpallivõistkondadele reaalajas punktiseisu jälgimiseks ja statistikatabelite koostamiseks.

Töö teoreetilises osas analüüsitakse samalaadseid olemasolevaid rakendusi. Seejärel pannakse kirja rakenduse nõuded ning analüüsitakse tehnoloogiaid, mida kasutatakse rakenduse loomisel.

Praktilises osas valmib veebirakenduse prototüüp, mida autor testib, et veenduda, kas seatud nõuded on täidetud. Seejärel testitakse prototüüpi võrkpallivõistkonna näitel.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 32 leheküljel, 8 peatükki, 5 joonist.

## **Abstract**

### **Live Scoreboard and Statistics Web Application Prototype for Volleyball Teams**

The aim of this Bachelor's thesis is to create a web application prototype for volleyball teams to input and track real-time score and statistics.

The theoretical part of the thesis analyzes existing similar applications. Afterwards, the requirements for the application are outlined and the technologies used in development are analyzed.

In the practical part, a prototype of the web application is created and tested by the author to ensure that the set requirements are met. The prototype is then tested by a volleyball team.

The thesis is in Estonian and contains 32 pages of text, 8 chapters, 5 figures.

## Lühendite ja mõistete sõnastik

API	<i>Application Programming Interface</i> , rakendusliides
BLL	<i>Business Logic Layer</i> , äriloogikakiht
CSS	<i>Cascading Style Sheets</i> , stiilide kirjelduskeel
DAL	<i>Data Access Layer</i> , andmekiht
DOM	<i>Document Object Model</i> , dokumendi objektimudel
DTO	<i>Data Transfer Object</i> , andme-edastusobjekt
GUID	<i>Globally Unique Identifier</i> , globaalselt unikaalne identifikaator
HTML	<i>HyperText Markup Language</i> , hüperteksti märgistuskeel
HTTP	<i>HyperText Transfer Protocol</i> , hüperteksti edastusprotokoll
JSON	<i>JavaScript Object Notation</i> , andmevahetusvorming
JWT	<i>JSON Web Token</i> , turvaline viis informatsiooni vahetamiseks JSON kujul
LINQ	<i>Language-Integrated Query</i> , tehnoloogiate komplekt päringute tegemiseks C# programmeerimiskeeles
ORM	<i>Object-Relationl Mapping</i> , tehnika, mis võimaldab objektorienteeritud programmeerimiskeelt kasutades suhelda andmebaasiga
REST	<i>Representational State Transfer</i> , tarkvaraarhitektuuri muster, mis seab piirid veebirakenduse loomiseks
SQL	<i>Structured Query Language</i> , päringukeel
URL	<i>Uniform Resource Locator</i> , internetiaadress

## Sisukord

1 Sissejuhatus .....	9
2 Ülevaade probleemist .....	10
2.1 Probleemi taust .....	10
2.2 Probleemi püstitus .....	11
2.3 Olemasolevate lahenduste analüüs .....	11
2.3.1 Data Volley 4.....	11
2.3.2 Click and Scout.....	12
2.3.3 Skoor.....	13
2.4 Uue lahenduse skoop .....	13
3 Loodava veebirakenduse prototüübi analüüs .....	15
3.1 Rakenduse nõuete määramine .....	15
3.1.1 Võrkpallimäng.....	15
3.1.2 Võistkond .....	15
3.1.3 Statistika .....	16
3.2 Kasutajate õiguste määramine .....	16
3.2.1 Anonüümne kasutaja .....	17
3.2.2 Tavakasutaja .....	17
3.2.3 Administraator .....	17
3.3 Kasutatavus.....	17
3.4 Võrkpall.....	20
3.4.1 Võrkpallimängu kestus .....	20
3.4.2 Statistika .....	20
4 Kasutatavate tehnoloogiate analüüs.....	22
4.1 Klientrakendus.....	22
4.1.1 Klientrakenduse programmeerimiskeeled .....	22
4.1.2 React .....	22
4.1.3 Vue .....	23
4.1.4 Angular ja AngularJS .....	23
4.1.5 Klientrakenduse raamistiku valik .....	24

4.2	Serverrakendus .....	24
4.2.1	C# .....	24
4.2.2	Java .....	25
4.2.3	PHP.....	25
4.2.4	Serverrakenduse tehnoloogia valik.....	26
4.3	Andmebaas .....	26
4.3.1	PostgreSQL.....	27
4.3.2	MySQL .....	27
4.3.3	SQLite.....	28
4.3.4	Andmebaasi valik .....	28
5	Rakenduse arendus .....	29
5.1	Klientrakendus.....	29
5.1.1	Struktuur .....	29
5.1.2	Kasutatud teegid .....	30
5.1.3	Reaalajas informatsiooni uuendamine.....	31
5.2	Serverrakendus .....	32
5.2.1	Suhtlemine andmebaasiga .....	32
5.2.2	Arhitektuur .....	32
5.2.3	REST API kontrollid .....	33
5.2.4	Reaalajas edastus .....	34
5.2.5	Turvalisus .....	35
5.3	Andmebaas .....	35
6	Testimine .....	37
6.1	Autori testimine .....	37
6.2	Võrkpallivõistkonna testimine.....	38
7	Tulemused .....	39
8	Kokkuvõte .....	40
	Kasutatud kirjandus .....	41
	Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks .....	45
	Lisa 2 – Klientrakenduse vaated.....	46
	Lisa 3 – Klientrakenduse ja serverrakenduse lähtekood .....	51

## Jooniste loetelu

Joonis 1. Mängu loomise voodiagramm.....	18
Joonis 2. Mängu voodiagramm. ....	19
Joonis 3. Ühenduse loomine ja sulgemine.....	31
Joonis 4. Vaatajate arvu uuendamine klientrakenduses. ....	31
Joonis 5. Olemisuhte diagramm. ....	36



## 1 Sissejuhatus

Eesti Võrkpalli Liit täidab reaajas punktiseisu ja statistikat kõrg- ja esiliiga võistkondade puhul, kuid erinevad Rahvaliiga tasemed on jäänud välja. Üheks põhjuseks võib pidada suurt võistkondade arvu, mistõttu rahvaliiga võistkonnad ei saa piisavalt kajastust. Samuti tehakse mängude kohta endiselt statistikat paberil või ei tehta seda üldse, kuigi see on mängijate arengut silmas pidades väga oluline.

Lõputöö eesmärk on luua veebirakenduse prototüüp reaajas punktiseisu ja statistika sisestamiseks võrkpallivõistkondadele, kus mängude informatsioon on kõigile vabalt kättesaadav.

Lõputöö teoreetilises osas analüüsitakse olemasolevaid rakendusi, millel on vastav funktsionaalsus. Määratakse lahenduse skoobis vajalikud nõuded ning seejärel analüüsitakse kasutatavaid tehnoloogiaid. Seejärel kirjeldatakse arendusega tehtud valikuid.

Töö praktilises osas valmib nõuetele vastav prototüüp, mida testivad autor ja võrkpallivõistkond.

## 2 Ülevaade probleemist

Selles peatükis antakse ülevaade probleemist ning püstitatakse eesmärk, mida lõputöös lahendatakse. Samuti tuuakse näiteid olemasolevate rakenduste kohta, kus on võimalik sisestada statistikat ning täita reaalajas punktiseisu. Seejärel pannakse paika rakenduse skoop.

### 2.1 Probleemi taust

Eesti Võrkpalli Liit ei täida Rahvaliiga võistkondade puhul reaalajas punktiseise ega tee statistikat, kuid kõrg- ja esiliiga võistkondade kohta seda tehakse. Põhjuseks on see, et tugevusgruppide ja võistkondade arv on suur. Hooajal 2022/2023 on rahvaliigas kolm meeste, kaks naiste ja kaks segavõistkondade tugevusgruppi ning kokku osaleb 120 võistkonda [1]. Eesti Võrkpalli Liidus kasutusel olev programm statistika ja punktiseisu sisestamiseks on litsentsipõhine ning informatsiooni kogumine Rahvaliiga kohta tähendaks lisaväljaminekuid.

Võistkond, kus autor mängib, kasutab endiselt statistika tegemiseks paberit, kuid enamik võistkondi ei tee statistikat üldse. Põhjuseks on see, et võistkonnad pole endi jaoks sobivat rakendust leidnud, kuna need võivad olla liiga keerulised, ebavajaliku või puuduliku funktsionaalsustega või liiga kulukad.

Võistkondade poolehoidjatel ja huvilistel on keeruline mängu vaadata minna, sest mängude toimumisest ei pruugita teadlikud olla, ka toimub autori kogemuse põhjal enamik Rahvaliiga mängu argipäeviti ja õhtustel kellaegadel. Rahvaliiga kõikide võistkondade mängude toimumiste kohta täidetakse rakenduses Google Sheets ajatabelit, mille jälgimine on ebamugav [2]. Samuti on mängude toimumisaegades kokkuleppimine võistkondade ülesanne ning kui kokkuleppele ei suudeta jõuda, sekkub Eesti Võrkpalli Liit [3].

## 2.2 Probleemi püstitus

Rahvaliiga võistkondade mängude kohta ei täideta reaalajas punktiseisu ega tehta statistikat. Lahenduseks on kõikidele võrkpallivõistkondadele see võimalus luua:

- Populariseerimine – pakkuda poolehoidjatele ja kõigile huvilistele mängude kohta reaalajas uuenevat punktiseisu ja statistikat, kui neil pole võimalik mängu vaatama minna. Reaalajas uuendamine tähendab seda, et vaataja ei pea leheküljelt lahkuma ja selleks tagasi tulema, et mängu kohta viimast infot saada.
- Statistika – anda võistkondadele võimalus koguda mängude kohta lihtsustatud kujul statistikat.

## 2.3 Olemasolevate lahenduste analüüs

Selles peatükis analüüsitakse ja tutvustatakse olemasolevaid lahendusi statistika tegemise ja reaalajas uueneva punktiseisu kohta. Peatükis tuuakse välja rakenduste positiivsed ja negatiivsed küljed.

### 2.3.1 Data Volley 4

Data Volley 4 on statistika ja analüüsi programm, millel on väga palju funktsionaalsusi peale punktiseisu ja statistika täitmise [4]. Ettevõtte Data Project loodud esimene Data Volley versioon valmis 2007. aastal ning Data Volley 4 Lite versioon 2017. aastal [5], [6]. Selle ettevõtte lahendusi kasutavad paljud tippliigad ja võistkonnad kogu maailmas [7].

Programm võimaldab teha ka videoanalüüsi ning käskluste põhjal suudetakse kuvada kõiki olukordi. Samuti saab manuaalselt sisestada lisaandmeid, et täiendada või parandada videoanalüüsi [8].

Data Volley 4 programmiga tuleb kaasa veel mitmeid rakendusi [8]:

- Veebipõhine rakendus kuvab infot abitreeneritele, peatreeneritele ja kommentaatoritele, mis on sisestatud Data Volley 4 programmis.
- Data Volley 4 Client on mõeldud professionaalseks kasutamiseks ning aitab veel põhjalikumalt mängu analüüsida, andes võimaluse koostada kohandatud analüüse.

- Data Volley 4 Reader on mõeldud kasutamiseks võistkondade peatreeneritele, kes tahavad peale mängu analüüsida enda võistkonda või järgmise mängu vastasvõistkonda. Rakendust saab kasutada litsentsivõtmeta.
- Data Volley 4 Player on mõeldud kasutamiseks mängijatele ja treeneritele, kes soovivad vaadata videot toimunud mängust. Andmed imporditakse Data Volley 4 programmist.

Praegu on programmist saadaval kaks erineva aastatasuga versiooni: Data Volley 4, mille aastane litsentsitasu on 799 eurot ning Data Volley 4 Lite, mille tasu on 299 eurot. Peamine erinevus kahe versiooni vahel on see, et Lite versioonil puudub ligipääs eelnimetatud neljale lisarakendusele [6].

Data Volley 4 programm on mõeldud kasutamiseks ainult Windowsi operatsioonisüsteemiga. Samas on Data Volley 4 Player saadaval Windowsi, Mac OS-i, Androidi ja iOS-i platvormidel [8].

Programmi positiivseks küljeks võib kindlasti pidada väga põhjaliku analüüsi ja statistika tegemise funktsionaalsust. Samas on statistika sisestamise programm saadaval ainult Windowsi operatsioonisüsteemiga seadmetele ja litsentsitasud on suured.

Programmi ja selle lahendused on mõeldud professionaalseks kasutamiseks ning on harrastusvõrkpallurite jaoks liiga keerulised. Kuna lõputöö eesmärk on pakkuda võimalust jälgida mängu reaajas, siis tuleks soetada Data Volley 4 täisversioon, mis oleks iga-aastase litsentsitasuga võistkonna jaoks väga kulukas.

### **2.3.2 Click and Scout**

Click and Scout on järgmine Data Projecti programm, mis loodi aastal 2015 ning on kasutatav iOS-i, Androidi ja Windowsi operatsioonisüsteemiga seadmetel. Rakendus on mõeldud statistika ja analüüsi tegemiseks. Rakenduses puudub täielikult võimalus kanda üle mängu punktiseisu [9].

Rakenduse üheks heaks küljeks Data Volley 4 lahendusega võrreldes on see, et seda saab kasutada nii saalivõrkpalli kui ka rannavõrkpalli puhul. Samuti peab rakenduse eest maksma litsentsitasu, mis on iga erineva operatsioonisüsteemiga seadme kohta 49 eurot aastas [9].

Rakenduse kasutatavus ei ole Data Volley 4 programmiga võrreldes keeruline, sest puuduvad lisafunktsionaalsused, mis pole harrastusvõrkpalli jaoks olulised. Endiselt tuleb maksta aastast litsentsitasu ning selle rakenduse puhul puudub võimalus mängu punktiseisu reaalajas üle kanda.

### **2.3.3 Skoor**

Skoor on veebipõhine rakendus, mis on mõeldud kasutamiseks mobiiltelefonis, tahvelarvutis ja sülearvutis, et sisestada statistikat ja reaalajas punktiseisu [10]. Skoor.ee vastab peamistele lõputöös seatavatele funktsionaalsustele, kuid lahenduse kasutajaliides ei ole vaatajatele kõige mugavam.

Samuti puudub üldine võistkonna haldamise loogika. See tähendab seda, et ei saa vaadata enda võistkonna mängija kombineeritud statistikat, mis on moodustunud läbi erinevate mängude, vaid saab vaadata mängu spetsiifiliselt.

Rakendus on väga hea baasfunktsionaalsusega, võimalik on sisestada punkte ja teha statistikat, mis ei ole suunatud otseselt võistkonnakeskseks kasutamiseks. Lahenduses saad luua võistkonnad, seejärel lisada mängijad ning mõne sammuga saab mänguga alustada. Samas on veebirakenduse loojal tulevikus plaanis lisada võistkonnakeskne loogika, et näha mängijate kohta koondülevaadet [10].

Rakenduse kasutamine on tasuline. Ühe mängu eest tuleb tasuda 2 eurot, 10 mängu eest 18 eurot ning 100 mängu eest 150 eurot. Veebirakenduses saab korraldada ka turniire, mis maksab 1500 eurot [11].

## **2.4 Uue lahenduse skoop**

Analüüsi järel selgus, et käsitletud rakendused on tasulised ning mõningate probleemidega. Data Projecti poolt loodud lahendused on mõeldud professionaalseks kasutamiseks ehk statistika ja analüüsi koostamine on keerukas ning väga süvitsi minev. Veebirakendus Skoor täidab statistika ja punktiseisu sisestamise eesmärgi, kuid puudub üldine võistkonnakeskne loogika ehk mängijate kohta ei saa koondülevaateid.

Eelnimetatud probleemide parandamiseks ja lisade loomiseks luuakse lõputöö käigus veebirakenduse prototüüp, mida saab kasutada mobiiltelefonis, tahvelarvutis ja arvutis.

Prototüübi käigus on kasutuses kolm kasutajate liiki: anonüümne kasutaja, administraator ja tavakasutaja.

Anonüümsetel ehk sisse logimata kasutajatel on võimalus vaadata informatsiooni toimunud mängude kohta ning jälgida reaajas punktiseisu ja statistikat. Samuti saab vaadata platvormil registreeritud võistkondi ning detaile mängijate kohta. Rakenduse prototüübi käigus on administraatori rollis kasutaja peamine ülesanne lisada kasutajaid. Üks eesmärk on see, et platvormile ei tekiks rakenduse prototüübi faasis soovimatuid kasutajaid. Rakenduse lõppversioonis saavad kasutajad ise registreeruda.

Tavakasutaja peab enne mängude sisestamist looma enda võistkonna. Seejärel tuleb võistkonda lisada mängijad. Prototüübi käigus saab tavakasutajal olla ainult üks võistkond. Edaspidi pöörduakse tavakasutaja poole kui registreeritud võistkonna poole, sest mängude loomine on seotud võistkondadega.

Rakenduses saab luua võrkpallimänge nii registreeritud kui ka registreerimata võistkondade puhul. Kuna mängud on kiiretempolised, siis on ühel võistkonna kasutajal keeruline pidada statistikat mõlema võistkonna kohta. Sellepärast saab võistkond täita statistikat ainult enda võistkonna kohta. Kui mängu loomise hetkel valitakse registreeritud võistkondade hulgast võistkond, siis on vastasvõistkonnal võimalus teha statistikat ka enda võistkonna kohta. Tulemusena on mängude jälgijatel ligipääs mõlema võistkonna statistikale.

Prototüübis ei piirata seda, kui mitme punktini kestab üks geim ning kui mitu geimi mängitakse. Võrkpallimängu kestuse kohta kirjutatakse täpsemalt järgmises peatükis.

Statistika loomise puhul keskendutakse lihtsustatud versiooni loomisele. Prototüübi käigus on võimalik hinnata rünnaku ja vastuvõtu sooritusi heaks, neutraalseks või halvaks. Vastase rünnaku blokeerimist ja servimist on võimalik hinnata heaks või halvaks. Statistika loomise puhul kasutatavaid termineid selgitatakse järgmises peatükis.

## **3 Loodava veebirakenduse prototüübi analüüs**

Selles peatükis püstitatakse prototüübi kõik vajalikud nõuded ning kirjeldatakse eri rollide funktsionaalsusi. Tutvustatakse võrkpallimängude kestust ning antakse ülevaade tähtsamatest terminitest, mis on olulised statistika loomisel.

### **3.1 Rakenduse nõuete määramine**

Peatükis kirjeldatakse rakenduse nõudeid, mis on saadud olemasolevate lahenduste analüüsimise järel, ja uue lahenduse skoobi kirjeldusest. Nõuete määramisel on arvestatud nii lahenduste funktsionaalsustega kui ka nende puudujääkidega.

#### **3.1.1 Võrkpallimäng**

Järgnevalt esitatakse võrkpallimänguga seotud nõuded:

- mängu loomine registreeritud ja registreerimata võistkonna vastu;
- mängu toimumisaja ja koha sisestamine;
- mängijate lisamine ja eemaldamine mängult;
- punktiseisu sisestamine;
- mängu alustamine ja lõpetamine;
- uue geimi alustamine;
- võistkondade statistika vaatamine;
- võistkonna statistika sisestamine;
- alustamata mängu kustutamine;

#### **3.1.2 Võistkond**

Järgnevalt esitatakse võistkonnaga seotud nõuded:

- võistkonna loomine ja muutmine;
- võistkonna logo üleslaadimine;

- mängijate lisamine ja muutmine;
- mängijate profiilipildi lisamine ja muutmine;
- võistkonna mängijate kohta informatsiooni vaatamine;

### 3.1.3 Statistika

Järgnevalt esitatakse statistikaga seotud nõuded:

- rünnaku märkimine heaks, neutraalseks või halvaks;
- vastuvõtu märkimine heaks, neutraalseks või halvaks;
- blokeerimise märkimine heaks või halvaks;
- servi märkimine heaks või halvaks;

Rünnaku, blokeerimise ja servi puhul tähendab soorituse heaks märkimine seda, et mängija tõi võistkonnale punkti ning halvaks märkimine seda, et mängija sooritusest sai vastasvõistkond punkti. Rünnaku neutraalseks märkimine tähendab seda, et peale sooritust jäi pall mängu ning kumbki võistkond veel punkti ei saanud.

Vastuvõttu märgitakse siis, kui mängija võtab vastase servi vastu. Heaks märgitakse vastuvõtt siis, kui pall maandub ligikaudselt võrgust 1,5 meetri kaugusele või lähemale. Neutraalseks märgitakse see siis, kui pall jääb mängu, aga seda ei suudeta toimetada vähemalt 1,5 meetri kaugusele võrgust. Halvaks märgitakse vastuvõtt siis, kui mängija eksib ja vastasvõistkond saab kohe punkti.

## 3.2 Kasutajate õiguste määramine

Peatükis kirjeldatud kasutajate õigused on koostatud analüüsi järel. Nõuded vaatas läbi autori võrkpallivõistkond.

Rakenduses vajalik kasutajate rollide valik on tehtud uue lahenduse skoobi järgi. Anonüümne kasutaja on võrkpallihuviline, kes soovib jälgida mängu ja vaadata informatsiooni mängude kohta. Tavakasutaja on rakenduse peamine kasutaja, kes soovib mängude kohta reaajas informatsiooni sisestada. Administraatori ülesanne on kontrollida rakenduse kasutajaskonda.



### **3.2.1 Anonüümne kasutaja**

Järgnevalt esitatakse anonüümse kasutaja õigused:

- võistkondade vaatamine;
- mängude vaatamine;
- mängijate vaatamine;
- statistika põhjal genereeritud blogi vaatamine;

### **3.2.2 Tavakasutaja**

Järgnevalt esitatakse tavakasutaja õigused lisaks anonüümsele kasutajale:

- parooli muutmine;
- mängude loomine, kustutamine ja alustamine;
- mängijate lisamine ja muutmine;
- võistkonna statistika sisestamine;
- kasutaja loodud mängu punktiseisu ja statistika uuendamine;

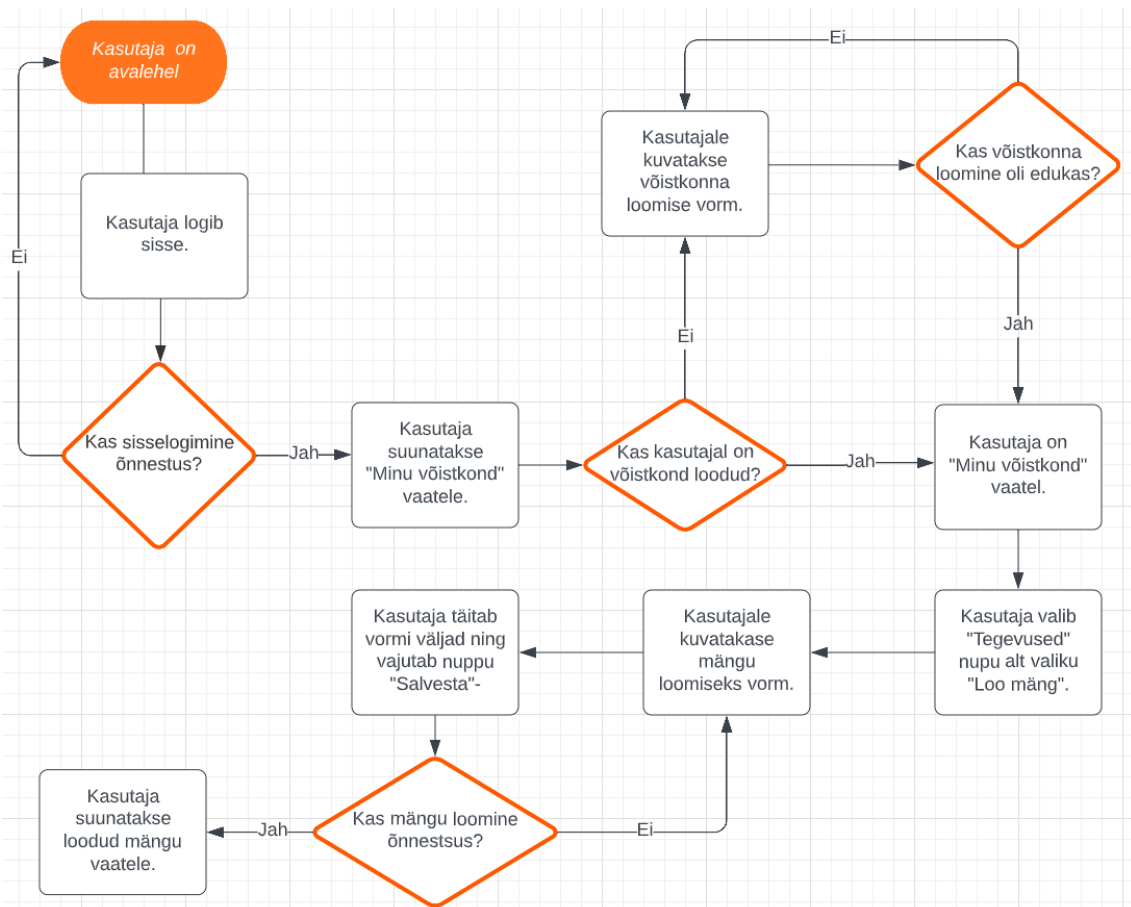
### **3.2.3 Administraator**

Prototüübi faasis on administraatori peamine õigus see, et kasutaja saab lisada veebirakendusse kasutajaid. Eesmärk on anda võimalus kontrollida arendus- ja prototüübifaasis kasutajaskonda ainult rakenduse autoril.

## **3.3 Kasutatavus**

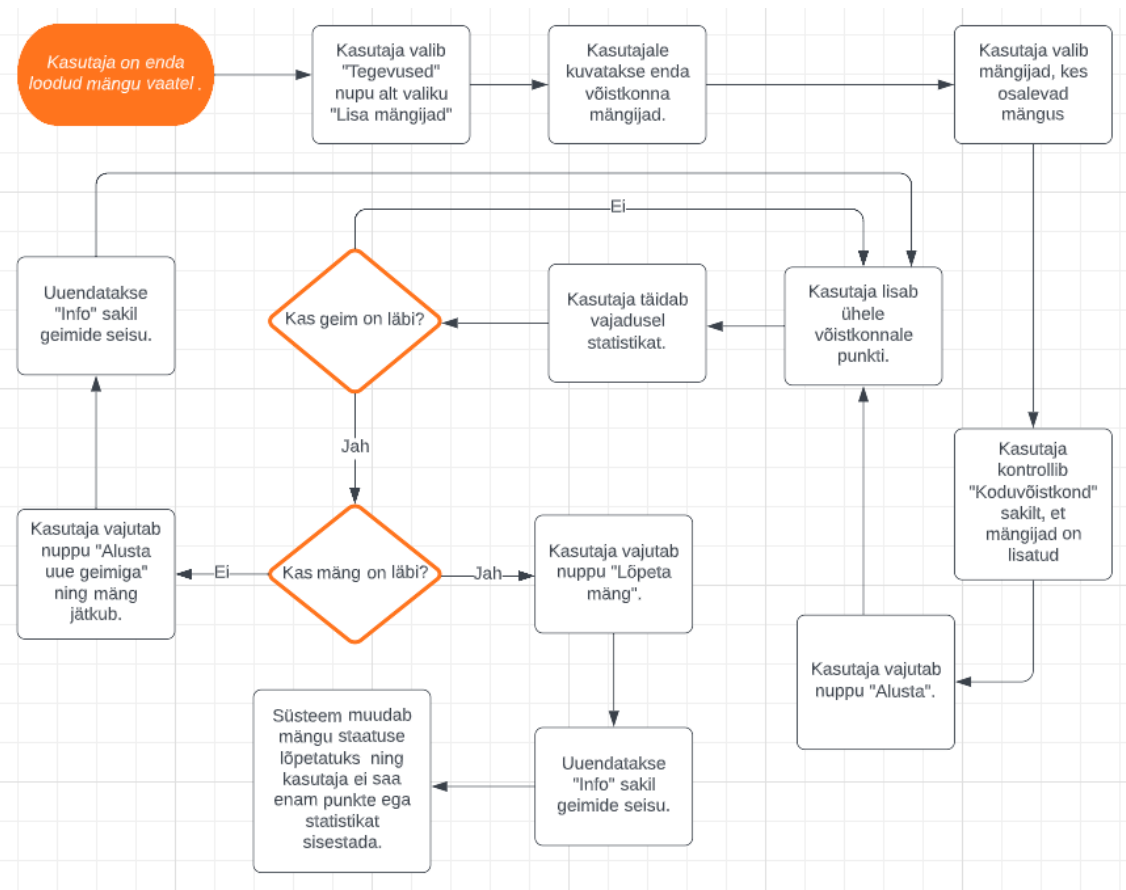
Kasutatavuse analüüsimiseks ning rakenduse vajalike vaadete loomiseks alustas autor voodiagrammi loomisest. Voodiagramm kujutab protsessi ning on skemaatiline esitus probleemi lahendamiseks. Rakendust disainides aitab voodiagrammi loomine määratleda vajalikke samme ning seab need kronoloogilisse järjekorda. Samuti annab voodiagramm suure pildi ning ettekujutuse rakenduse protsessidest [12].

Järgnevalt on kujutatud voodiagrammi, kui kasutaja soovib luua mängu (joonis 1).



Joonis 1. Mängu loomise voodiagramm.

Peale mängu loomist on võimalik mänguga alustada. Järgnev voodiagramm kujutab mängu alustamist ja kulgu, kui mängitakse registreerimata võistkonna vastu (joonis 2). Erinevus registreeritud võistkonnaga on see, et registreeritud võistkond saab ka mängus enda statistikat täita.



Joonis 2. Mängu voodiagramm.

Peale voodiagrammide loomist langetas autor otsuse, et millised vaated ja menüüpunktid on vajalikud rakenduse kasutamiseks.

Rakenduses olevad vaated:

- Vaade „Minu võistkond“ on mõeldud võistkonna haldamiseks ja mängude loomiseks.
- Vaadetel „Mängud“ ja „Minu mängud“ kuvatakse vastavalt kõik süsteemis loodud mängud ja kasutaja mängud.
- Vaates „Võistkonnad“ ja võistkonna detailvaates, kus kuvatakse rakenduses registreeritud võistkonnad ning samuti on võimalik vaadata täpsemat informatsiooni mängijate ja võistkondade kohta.
- Mängu detailvaade kuvab informatsiooni konkreetse mängu kohta. Kasutaja, kes on loonud vastava mängu, saab mängus tegutseda. Sisse logimata kasutaja saab seda jälgida.

## **3.4 Vörkpall**

Selles peatükis kirjeldatakse olulisemaid vörkpallireegleid, mis on vajalikud loodava prototüübi skoobis. Samuti antakse ülevaade vörkpalli statistikast ja täpsustatakse kategooriate tähendust.

### **3.4.1 Vörkpallimängu kestus**

Vörkpalli puhul on tegu kahe peamise stiiliga: rannavörkpall ja saalivörkpall. Nende peamised erinevused on platsi suurus ja mängu kestus. Rannavörkpallis mängitakse kahe geimi võiduni ja geim mängitakse 21 punktini. Saalivörkpallis mängitakse kolme geimi võiduni ning geim lõppeb, kui võistkond kogub 25 punkti. Vajaduse korral mängitakse vastavalt kolmas geim või viies geim, mis mängitakse 15 punktini. Mõlema stiili puhul peab võistkond igas geimis olema vastasvõistkonnast kahe punktiga ees, et geim võita [13].

Rakenduse skoobis mängude kestusi ei piirata. Selle tulemusel saab loodut lahendust kasutada erinevate vörkpallimängude puhul.

### **3.4.2 Statistika**

Statistika on oluline, et peegeldada mängijate efektiivsust ja analüüsida mängu tulemust. Treenerid saavad ülevaate mängijate puudustest ja valmistuda selle põhjal järgmisteks treeninguteks [14]. Statistikat täidetakse peamiselt kuues kategoorias: ründamine, tõstmine, servimine, servi vastuvõtmine, kaitsemäng ja blokeerimine [15]. Autor on loodava rakenduse skoobis valinud statistika täitmiseks neli kategooriat: rünnak, vastuvõtt, servimine ja blokeerimine. Valik otsustati selle põhjal, mida autori võistkond on varem paberil statistika tegemiseks märkinud.

Rünnak on liigutus, kui mängija toimetab palli üle võrgu vastase poolele, sooviga tuua punkt enda võistkonnale. Rünnaku puhul saab määratleda soorituse tulemust järgmiselt: punkti toomine enda võistkonnale, rünnakul eksimine ja vastasvõistkond saab punkti või rünnaku tulemusel jääb pall mängu ning vastasvõistkond saab võimaluse rünnakuks [15].

Servimine on liigutus, kui mängija paneb palli esmakordselt mängu. Servimise tulemusel võib pall jääda mängu, servi sooritanud võistkond saab kohe punkti või eksitakse servil ja vastasvõistkond saab punkti [15].

Vastuvõtuks peetakse mängija esimest kokkupuudet palliga peale servi sooritamist ning tulemuseks võib olla sooritus või viga [15]. Autori võistkonnas hinnatakse vastuvõttu tulemust kolmeks: hea vastuvõtt, kui pall maandub võrgust 1,5 meetri kaugusele võrgust või lähemale, neutraalseks, kui vastuvõtt jääb kaugemale kui 1,5 meetrit, ning halvaks, kui vastuvõtu tulemusel saab vastasvõistkond punkti.

Blokeerimine on tegevus, kui takistatakse vastase rünnakut, viies käed rünnaku hetkel üle võrgu. Blokeerimise tulemust saab hinnata abistavaks, veaks, punkti toovaks või olukorraks, kus pall jäi mängu [15]. Autori võistkond on blokeerimist märkinud punkti toovaks või veaks ning rakenduses kasutatakse viimast lahendit.

## **4 Kasutatavate tehnoloogiate analüüs**

Selles peatükis analüüsitakse klientrakenduse, serverrakenduse ja andmebaasi tehnoloogiaid, et valida enne arenduse alustamist sobivad tehnoloogiad.

### **4.1 Klientrakendus**

Klientrakendus on veebirakenduse osa, mis on loodud kasutajale intuitiivse ja lihtsa kasutajakogemuse tagamiseks. Samuti määrab klientrakendus, kuidas kasutajad saavad tegutseda, vajutades nuppe ning täites välju [16]. Enne arendust valis autor klientrakenduse loomiseks sobivad vahendid. Uuriti võimalikke raamistikke ning programmeerimiskeeli, et teha klientrakenduse loomiseks sobiv valik. Arvesse võeti autori kogemust analüüsitava tehnoloogiatega ja nende funktsionaalsust.

#### **4.1.1 Klientrakenduse programmeerimiskeeled**

JavaScript on üks kõige laialdasemalt kasutatav keel klientrakenduse loomiseks, mis võimaldab arendajatel ehitada interaktiivseid veebilehti [17]. JavaScripti peamiseks kasutusvaldkondadeks on dünaamiliste veebilehtede loomine, animatsioonide ning interaktiivsete funktsioonide lisamine veebilehtedele [18]. Samas saab JavaScripti kasutada ka server- ja mobiilirakenduste arendusteks [19].

TypeScript on tugevalt tüübitud programmeerimiskeel, mis on arendatud Microsofti poolt ning ehitatud JavaScripti peale, andes juurde funktsionaalsusi ning muutes rakenduse skaleeritavamaks. TypeScripti kood on teisendatav JavaScripti koodiks, mis tähendab, et TypeScripti kood töötab ka seal, kus töötab JavaScript. Tüübikindluse lisamine aitab arendajatel varakult koodis vigu leida [20]–[22].

#### **4.1.2 React**

React on avatud lähtekoodiga JavaScripti teek, mis on arendatud Facebooki poolt ning võimaldab arendajatel ehitada kasutajaliideseid. Reacti peamine eelis on selle komponentide struktuur, mis võimaldab kiiret arendust ja modulaarsust [23]. Komponent

on väike ja iseseisev tükk koodi, mis vastutab konkreetse kasutajaliidese osa loomise eest, mis lisatakse eraldi failidesse, et tagada taaskasutatavust [24]. React kasutab virtuaalset DOM-i (*Document Object Model*), mis võrdleb komponentide eelmisi olekuid ja uuendab ainult muutunud elemente, mille tulemusel muutub rakendus kiiremaks [25].

Üheks peamiseks erinevuseks Reacti ja teiste raamistike puhul on see, et tegelikult loetakse seda teegiks, sest tegelikult puuduvad sel peamised funktsionaalsused. Näiteks puudub sisseehitatud navigeerimise võimalus, mille loomiseks tuleb lisada teek [26].

Stack Overflow' küsitluse põhjal on React 2022. aastal 2. kohal kõige kasutatavamate raamistike hulgas – 58 743 vastanu seast kasutab Reacti 42,62% vastanutest [27]. Autoril on Reactiga töötades üheksa kuud kestnud kogemus.

#### **4.1.3 Vue**

Vue on JavaScripti raamistik, millega saab ehitada kasutajaliideseid. See on ehitatud HTML-i (*HyperText Markup Language*), CSS-i (*Cascading Style Sheets*) ja JavaScript-i peale ning pakub deklaratiivset ja komponendipõhist programmeerimist. Vue laiendab HTML-i malli süntaksiga, mis võimaldab deklaratiivselt kirjeldada HTML-i JavaScripti olekust sõltuvalt. Samuti jälgib Vue automaatselt JavaScripti oleku muutusi ning sellele vastavalt uuendab DOM-i [28].

Vue struktuur on loogiline ja läbipaistev, mis muudab selle lihtsasti mõistetavaks ja kasutatavaks. Samuti on raamistiku maht väike, mis on kindlasti selle eeliseks. Samas on Vue arendatud Hiinas ning teekide kirjeldused ja probleemidele vastuste saamine võib olla raskendatud [29].

2022. aasta StackOverflow' küsitluse põhjal on Vue oma populaarsuselt 6. kohal – 58 743 vastanust eelistab seda 18,82% [27]. Autor on loonud seda raamistikku kasutades ühe rakenduse.

#### **4.1.4 Angular ja AngularJS**

Angular on TypeScripti peale ehitatud arendusplatvorm. See sisaldab endas komponendipõhist raamistikku, millega on võimalik ehitada skaleeritavaid kasutajaliideseid. Samuti on Reactiga võrreldes, kus peab sobivad teegid arendaja ise juurde lisama, juba sisse integreeritud mitmed teegid, millega luua navigeerimist ja vorme [30].

2016. aastal jaotati raamistik AngularJS-iks ning Angulariks. AngularJS oli esimene versioon raamistikust ning Angular on selle järglane. Nende peamiseks erinevuseks on see, et Angular kasutab TypeScripti programmeerimiskeelt. Samuti on Angulari jõudlus ja kiirus suuremad. 2021. aastal lõpetas Google toe pakkumise AngularJS raamistikule, seega soovitatakse kasutada Angulari raamistikku [31].

2022. aasta Stack Overflow' küsitluse põhjal on Angular 5. kohal ja edastab Vue raamistikku. AngularJS on 8,99%-ga 13. kohal [27]. Autoril puudub nende raamistikega väljaspool ülikooli kogemus.

#### **4.1.5 Klientrakenduse raamistiku valik**

Autor otsustas analüüsi põhjal valida klientrakenduse tehnoloogiaks Reacti teegi, sest kogemus selle raamistikuga on võrreldes Vue ja Angulariga suurim. Samuti on React analüüsitud raamistikest kõige populaarsem, mille tulemusel on veebist lihtne leida probleemidele lahendusi ja kasutada eri teeke.

Programmeerimiskeeleks valiti TypeScript, sest see on JavaScripti edasiarendus, millesse on lisatud funktsionaalsusi ning selle tüübikindlus aitab koodivead varakult avastada, ka on arendusprotsess kiirem.

## **4.2 Serverrakendus**

Serverrakenduse ülesandeks konkreetse töö raames on andmete töötlemine, salvestamine, suhtlemine andmebaasiga ja informatsiooni edastamine klientrakendusele. Enne serverrakenduse programmeerimiskeele ja raamistiku valimist analüüsiti nende funktsionaalsusi ja populaarsust. Autor langetas otsuse enda kogemuse põhjal.

### **4.2.1 C#**

C# on kaasaegne, objektorienteeritud ja tüübikindel programmeerimiskeel, millega saab luua turvalisi ja võimekaid rakendusi, mis töötavad .NET raamistikus. C# on seotud C programmeerimiskeelte perekonnaga ning keele süntaks on C, C++ ja Java arendajatele tuttav [32].

.NET on Microsofti loodud tarkvararaamistik, mis võimaldab arendajatel luua mitmeplatvormilisi rakendusi. .NET on vabavaraline ja selle arendamist toetab suur



kogukond, mis tähendab, et sellega on seotud palju abistavaid ressursse ja dokumentatsioone, mis võimaldab arendajatel tõhusamalt töötada [33].

Autor tunneb ennast kõige mugavamalt C# programmeerimiskeelega ja .NET raamistikuga, sest seda on ülikoolis kõige rohkem õpetatud. Samuti töötab autor igapäevaselt selle keele ja raamistikuga iga päev.

#### **4.2.2 Java**

Java on laialdaselt kasutatud objektorienteeritud programmeerimiskeel ja tarkavaraplatform, mida kasutatakse sülearvutites, mobiiltelefonides, mängukonsoolides ja teistes seadmetes. Keele süntaks ja reeglid põhinevad C ja C++ keeltele [34].

Lihtsustamaks serverrakenduse loomist on mõistlik kasutada erinevaid raamistikke. Üheks populaarsemaks näiteks on Spring ja sellega seotud raamistikud, millega on võimalik lisada rakendusele turvalisuse aspekt ja luua ühendused andmebaasiga [27], [35], [36].

2022. aasta Stack Overflow' küsimustiku põhjal töötab Java programmeerimiskeelega professionaalselt 53 421 vastanust 33,4%. Seda on ligi neli protsendipunkti rohkem C# keelega võrreldes [37].

Autori kogemus konkreetse programmeerimiskeelega põhineb ülikoolis õpitul. Üheski hobiprojektis ei ole autor seda keelt kasutanud ning sellepärast on tunne ebakindel. Samuti puudub tööalane kogemus.

#### **4.2.3 PHP**

PHP on laialdaselt kasutatud vabavaraline skriptimiskeel, mis sobib hästi veebiarenduseks. PHP-kood manustatakse HTML-i märgistuskeele sisse, kasutades PHP omaseid märgiseid. Peamiseks erinevuseks JavaScriptiga võrreldes on see, et PHP-kood käivitatakse serveris ning genereeritud HTML saadetakse klientrakendusse. Selle tulemusel ei tea klient, mis kood käivitati [38].

Üheks populaarsemaks toetavaks raamistikuks on Laravel [37]. Raamistik võimaldab ehitada nii API (*Application Programming Interface*) kontrollereid kui ka monoliitseid rakendusi. Samuti pakub andmebaasiga ühendamise ja autentimise võimaluse.

Autor on õppinud antud programmeerimiskeelt ühes õppeaines ning kokkupuude sellega on kõige väiksem.

#### **4.2.4 Serverrakenduse tehnoloogia valik**

Analüüsi järel otsustas autor kasutada veebirakenduse prototüübi arendamiseks C# programmeerimiskeelt ja .NET raamistikku. Otsus langetati peamiselt autori senise kogemuse põhjal eelmainitud keelte ja raamistikega. Ka on nende tehnoloogiate populaarsuse tõttu veebist lihtsam abi ja dokumentatsioone leida.

### **4.3 Andmebaas**

Andmebaas on struktureeritud andmete kogum, mis on tavaliselt hoiustatud arvutisüsteemis. Andmebaasi kontrollitakse andmebaasihaldussüsteemi toel. Andmed koos andmebaasihaldussüsteemiga moodustavad ühe terviku, mida kutsutakse andmebaasiks. Enamik andmebaase kasutab andmetele ligipääsemiseks ning kirjutamiseks SQL (*Structured Query Language*) päringukeelt [39].

Andmebaasitüüpe on mitmeid ning sellest sõltub, milliseid rakendusi on võimalik nendega luua. Antud lõputöö raames selgitakse kahte kõige enim kasutatud andmebaasitüüpi [39].

Relatsioonilises andmebaasis on andmed organiseeritud tabelitesse, veergudesse ja ridadesse. See tehnoloogia pakub andmetele ligipääsemiseks kõige efektiivsemat ja paindlikumat viisi [39].

Objekt-orienteeritud andmebaasides on informatsioon esitatud objektide ja klassidena, nagu objekt-orienteeritud programmeerimises [39].

Objekt-relatsiooniline andmebaas on edasiarendatud versioon objekt-orienteeritud andmebaasi tüübist ning põhineb kahel eelnimetatud andmebaasitüübil [40].

Lõputöö raames analüüsitakse kolme kõige populaarsemat andmebaasisüsteemi, mida kasutatakse professionaalselt. Aluseks on võetud Stack Overflow' küsitlus.

### 4.3.1 PostgreSQL

PostgreSQL on võimas, tasuta ja avatud lähtekoodiga objekt-relatsiooniline andmebaasisüsteem, mis kasutab ja laiendab SQL-i. Samuti on PostgreSQL väga kohandatav, mis tähendab, et võimalik on luua kasutaja defineeritud andmetüüpe ja funktsioone [41].

PostgreSQL püüab järgida SQL-i standardit, et ei tekiks vastuolu traditsiooniliste funktsioonidega. Paljud nõutavad funktsioonid on toetatud, kuigi võivad olla erineva süntaksiga. PostgreSQLi versioon 15, mis tuli välja 2022. aasta oktoobris, vastab vähemalt 170 nõutavale funktsionaalsusele 179-st, mis on seatud SQL 2016 standardi kohaselt [41].

Stack Overflows tehtud küsitluse põhjal on PostgreSQL enim professionaalselt kasutatud andmebaasisüsteem ja selle poolt hääletas 48 788 vastanust 46,48% [42].

Autor on kasutanud PostgreSQLit mitmes suuremas ülikooli projektis.

### 4.3.2 MySQL

MySQL on Oracle'i arendatud avatud lähtekoodiga, kiire ja skaleeritav andmebaasisüsteem. MySQLi andmebaasserver on kasutatav arvutis või sülearvutis koos teiste rakendustega, sest see ei koorma ressursi. MySQL oli algselt arendatud käsitlema suuri andmebaase tunduvalt kiiremini, kui eksisteerivad lahendused seda võimaldasid [43].

MySQL on lihtne ja kergesti seadistav. Samas pakub PostgreSQL rohkem funktsionaalsusi ja andmetüüpe. Peamiseks erinevuseks on see, et MySQL on relatsiooniline ja PostgreSQL objekt-relatsiooniline andmebaas [44].

Stack Overflow' küsitluse põhjal eelistab MySQLi andmebaasi 48 788 vastanust 45,68%, mis seab MySQLi teisele kohale [42]. Samuti on MySQL lähemale jõudnud PostgreSQLi funktsionaalsustele ning sellepärast on nad ka populaarsuselt sarnased [44].

Autoril puudub suurem kogemus MySQL andmebaasidega.

### 4.3.3 SQLite

SQLite on tasuta ja avatud lähtekoodiga relatsiooniline andmebaasimootor, mis toetab SQL-i päringuid. SQLite omadusteks on kiirus ja eraldiseisvus. SQLite on sisse ehitatud kõikidesse telefonidesse ja paljudesse arvutitesse [45]. Kui SQLite pakub seadmete ja rakenduste puhul pigem lokaalset andmete hoiustamist, siis eelmainitud andmebaasid rõhuvad skaleeritavusele ja tsentraliseeritusele [46].

SQLite on kui serverivaba andmebaas, kus andmebaasimootor on integreeritud rakendusse. Enamik relatsioonilisi andmebaase on serveripõhised, kus rakendus peab andmete küsimiseks ja sisestamiseks ühenduma serveriga. Selle tulemusel on SQLite seadistamine MySQLi ja PostgreSQL'i andmebaasidega võrreldes kiirem. Samuti toetab SQLite andmebaas märgatavalt vähem andmetüüpe kui eelnevalt analüüsitud andmebaasid [47].

Stack Overflow küsitluse põhjal on SQLite kolmandal kohal – 30,83% vastanutest on seda professionaalselt kasutanud [42].

Autor on kasutanud andmebaasimootorit ülikoolis väiksemates projektides ja ülesannete lahendamiseks. Ülikooliväliselt ei ole autor sellega kokku puutunud, sest ei ole arendanud selliseid lahendusi, mis vajaksid spetsiaalselt lokaalset andmete hoiustamist.

### 4.3.4 Andmebaasi valik

Analüüsi järel valis autor kasutamiseks PostgreSQL'i andmebaasi. Valikuks osutus sellepärast, et populaarsus MySQLiga on peaaegu võrdne ning funktsionaalsuste erinevus on ühtlustumas. Samuti on autoril rohkem kogemust PostgreSQL'i andmebaasisüsteemiga.

## 5 Rakenduse arendus

Peatükis kirjeldatakse veebirakenduse prototüübi arendusega seotud protsessi. Tutvustatakse klientrakenduse, serverrakenduse ja andmebaasi struktuuri ja nende olemust.

### 5.1 Klientrakendus

Klientrakenduse loomiseks kasutati Reacti teeki ja TypeScripti programmeerimiskeelt. Arendusprotsessi alguses genereeriti klientrakenduse struktuur järgneva käsklusega: `npx create-react-app`.

Arendust alustati erinevate teekide installimise ning seadistamisega. Samuti loodi algne struktuur loogika eristamiseks. Esmalt loodi vaated kasutaja sisse logimiseks, et saaks alustada kasutajakeskse arendusega. Järgnevalt võeti arvesse rakenduse kasutatavust (joonis 1 ja 2). Nendest tulenevalt alustati võistkonna loomise ja mängijate lisamise loogika arendamisega ning lisati juurde funktsionaalsused piltide lisamiseks ja andmete muutmiseks. Seejärel oli võimalik luua mängu ning alustada mängu loogika arendamisega.

#### 5.1.1 Struktuur

Autor on struktuuri luues toetunud varasemale kogemusele. Struktuuri loomine aitab eraldada koodi erinevad vastutusalasid ja muudab koodi loetavamaks. Järgnevalt kirjeldatakse struktuuri ülesehitust:

- Kaust `app` sisaldab teenuseid, millega on võimalik serverrakendusega suhelda ning vahevara (*middleware*) ja objektide definitsioone. Samuti sisaldab see rakenduse oleku kohta abistavaid funktsioone.
- Kaustas `components` on üldkasutatavad komponendid, et vähendada koodikordusi ja jagada kood loogilisteks osadeks.

- Kaustas constants sisalduvad numbrilised ja sõnalised väärtused, mida kasutatakse näiteks erinevates vormides.
- Kaust hooks sisaldab autori defineeritud Reacti teegi *hook*-e.
- Kaustas icons on üldkasutatavad ikoonid.
- Kaustas screens on rakenduse peamised vaated. Samuti sisaldab see alamkatalooge funktsionaalsete ja visuaalsete komponentide jaoks.
- Kaustas utils on üldkasutatavad abistavad funktsioonid. Näiteks on loodud funktsioonid kuupäevade vormindamiseks.

### 5.1.2 Kasutatud teegid

Arendusprotsessi kiiruse ja mugavuse huvides kasutati eesrakenduse arenduses abistavaid teeke. Järgnevalt esitatakse ülevaade peamistest klientrakenduses kasutatavatest teekidest:

- Antd Design ja Antd Design Mobile – lõputöö eesmärk oli arendada funktsionaalne veebirakenduse prototüüp ning selle kiiremaks arendamiseks kasutati komponentide kogumeid Ant Design ja Ant Design Mobile. Vastavad kogumid sisaldavad eeldefineeritud blokke kasutajaliidese loomiseks. Näiteks on saadaval eri stiilides nupud ja väga hästi kohandatavad vormid [48], [49].
- Redux Toolkit ja Redux – Redux on olekuhalduse konteiner, mida kasutatakse laialdaselt eesrakendustes. Redux Toolkit on ametlikult soovitatud viis, kuidas kirjutada olekuloogikat. See sisaldab endas mitmeid abistavaid funktsioone ja lihtsustab loogika kirjutamist. Näiteks on võimalik luua ühtlase struktuuriga teenuseid [50]. Autor valis need teegid sellepärast, sest on varem nendega töötanud.
- React Router – React Router teeb rakenduses navigeerimise loomise lihtsaks. Kasutaja defineerib kõik URL-id (*Uniform Resource Locator*) ning neile vastavad vaated [51].
- Lodash – Lodash on kogum abistavaid funktsioone. Näiteks aitab kiiremini ja lihtsamini töötada massiivide ja objektidega [52].

- SignalR – klientrakenduses reaalajalise funktsionaalsuse loomiseks kasutati SignalRi teeki, mis võimaldab suhelda serverrakendusega [53].

### 5.1.3 Reaalajas informatsiooni uuendamine

Reaalajas info uuendamiseks kasutati teekide SignalR ja Redux võimalusi. Nende kahe teegi vaheliseks koostööks on loodud klientrakenduses vahevara. Vahevara kuulab klientrakenduses oleku muutusi ning kui antakse käsklus, et ühenduda soovitud mänguga, siis alustatakse ühenduse loomist. Samuti vastutab vahevara mängust lahkudes selle eest, et loodud ühendus suletakse. Järgnevalt tuuakse konkreetne näide, kuidas toimub ühendus loomine ja sulgemine (joonis 3).

```
export const onlineGameMiddleware : Middleware = (store: any) =>
{
  let connection : HubConnection | null = null;

  return next => action => {

    if (gameActions.setConnect.match(action)) {
      const { gameId } = action.payload;
      connection = connect(gameId);
      subscribe(store, connection);
    }
    if (gameActions.setDisconnect.match(action)) {
      disconnect(connection);
      connection = null;
    }

    return next(action);
  };
};
```

Joonis 3. Ühenduse loomine ja sulgemine.

Peale ühenduse loomist kuulab klientrakendus serverrakenduse saadetud informatsiooni, kuid tuleb defineerida, mida erinevate funktsioonide puhul tehakse. Järgnevalt tuuakse näide, kuidas toimub vaatajate arvu uuendamine (joonis 4).

```
connection.on('personJoined', (data: number) => {
  store.dispatch(gameActions.setViewersCounts(data));
});
```

Joonis 4. Vaatajate arvu uuendamine klientrakenduses.

Mängu ühendudes on klientrakendusele tagastatud kogu mänguga seotud informatsioon, sealhulgas mängu vaatajate arv, mis on salvestatud rakenduse olekusse. Joonisel 4 kuulatakse kutsungit *personJoined*, mille informatsiooni saadab serverrakendus. Selle

tulemusel uuendatakse rakenduse olekut ning kasutajaliideses uuendatakse vajalikud komponendid. Sama lahendust on kasutatud kogu reaajas uueneva informatsiooni kuvamiseks ja muutmiseks.

## 5.2 Serverrakendus

Selles peatükis tutvustatakse serverrakenduse arhitektuuri ja ülesehitust. Samuti antakse ülevaade sellest, kuidas toimub reaajas suhtlus klientrakenduse ja serverrakenduse vahel. Serverrakenduse arendamiseks kasutati .NET versiooni 7.0.

### 5.2.1 Suhtlemine andmebaasiga

Andmebaasiga suhtlemiseks kasutati Entity Frameworki raamistiku, mis võimaldab töötada andmetega, kasutades objekt-relatsioonilist kaardistamist ehk ORM-i (*Object-Relationl Mapping*). Entity Framework võimaldab arendajatel töötada andmetega, kasutades LINQ-i (*Language-Intergrated Query*), mis võimaldab kirjutada SQL-päringuid nagu koodi. Samuti saab kasutada *code-first*-lähendamist, mille tulemusel genereeritakse automaatselt vastavad andmebaasi tabelid [54].

### 5.2.2 Arhitektuur

Serverrakenduse arenduse puhul on teostatud kolmekihiline arhitektuur, mis aitab efektiivsemalt arendada ja kergemini muudatusi sisse viia. Eesmärk on tagada koodi parem loetavus, koodi loogilisteks osadeks jagamine ja testimine [55]. Nendeks kolmeks kihiks on DAL (*Data Access Layer*) ehk andmekiht, BLL (*Business Logic Layer*) ehk ärioloogikakiht ja esitluskiht.

Andmekiht võimaldab rakendusel andmebaasiga suhelda. Andmekiht asetseb andmebaasi või andmehoidla ja ärioloogikakihi vahel ning selle eesmärk on pakkuda ärioloogikakihil ligipääsu andmetele ja teha nendega toiminguid. Andmekihi peamine eesmärk on eraldada andmebaasi juurdepääsu kood ülejäänud rakendusest [56]. Andmekihis on kasutatud hoidlamustrit (*repository pattern*), mis on selle abstraksiooniks. Üldjuhul on iga domeeni objekti jaoks loodud eraldi hoidla [57].

Autor on loonud ühe üldise hoidla, millega on võimalik sisestada, uuendada, küsida kõiki kirjeid ning konkreetseid kirjeid primaarvõtme põhjal. Eesmärk on vähendada eelnimetatud tegevuste koodikorduseid. Samuti on loodud domeeni objektide kohta



hoidlad, et oleks võimalik teha spetsiifilisemaid päringuid ja kaasata seotud informatsiooni.

Äriloogikakiht ehk *business-logic layer* eesmärk on töödelda andmeid ja teha nendega reeglitele vastavaid toiminguid. Samuti võib see kiht täita turvalisuse eesmäärke, mis tähendab, et kontrollitakse, kas kasutajal on ligipääs teatud toimingutele või andmetele ning vajaduse korral koostatakse vastus, et anda kasutajale teada ebaõnnestunud tegevusest [58]. Näiteks kontrollitakse loodud veebirakenduse prototüübis mängijate lisamist võistkonda. Selleks küsitakse päringu tegija võistkonna primaarvõti ning kontrollitakse, kas võistkond, kuhu kasutaja soovib mängijat lisada, kuulub talle. Sellega tagatakse andmete õigsus ning piiratakse võimalust teha tegevusi teiste võistkondadega. Selles rakenduses asetseb äriloogikakiht REST (*Representational State Transfer*) API kontrollierite ja andmekihi vahel.

### 5.2.3 REST API kontrollierid

Eelmises peatükis mainitud kihtidele järgnevad viimasena REST API kontrollierid, mis täidavad esitluskihi ülesandeid. Eesmärk on tagastada andmeid, mille päringud on tehtud klientrakendusest.

REST API on rakenduse programmeerimisliides, mis vastab REST-i arhitektuuri stiili piirangutele ning kasutab HTTP (*HyperText Transfer Protocol*) protokollit. See tähendab et kasutajatel on võimalik teha GET, POST, PUT, DELETE päringuid, et pääseda ligi või muuta andmeid [59].

Klientrakendus kasutab päringute tegemiseks API lõpp-punkte, mis on REST API üheks komponendiks. Lõpp-punkt on kindel URL, millele vastab serverrakenduses kindel ressurss [60].

Lõputöös on vajalikud kontrollierid loodud manuaalselt, mis sisaldavad endas vajalikke lõpp-punkte rakenduse toimimiseks. Kontrollierid on loodud vastavalt vajadusele ja arendusprotsessile.

Vastavalt päringutele tagastatakse andmed DTO (*Data Transfer Object*) objekti kujul. DTO on objekt, mis kannab andmeid erinevate protsesside vahel – näiteks andmete saatmisel klientrakenduse ja serverrakenduse vahel. Peamine eesmärk on tagada informatsiooni turvalisus, et tundlikku informatsiooni ei paljastataks. Samuti on eesmärk

tagastada kasutajale andmed, mida klientrakendusel on konkreetsel hetkel vaja [61]. Loodud lahenduses teisendatakse domeeniobjektid otse DTO objektideks, mida API lõpp-punktid tagastavad JSON-i (*JavaScript Object Notation*) kujul.

#### 5.2.4 Reaalajas edastus

Veebirakenduse üheks nõudeks oli see, et mängu vaatajale tagastatakse infot reaalajas, mis tähendab, et kasutaja ei pea veebilehte värskendama, et saada mängu kohta kõige uuemat infot.

Lahenduse loomiseks on serverrakenduses kasutusele võetud teek SignalR – Microsofti loodud avatud lähtekoodiga tehnoloogia, mis võimaldab veebirakenduste jaoks reaalajas kahepoolset suhtlust. Võimalus on luua püsiv ühendus serverrakenduse ja klientrakenduse vahel ning edastada reaalajas andmeid mõlemas suunas. SignalR kasutab WebSocketi, SSE (*server-sent events*) ja *long-polling*’u tehnikaid [62].

WebSocket on protokoll, mis võimaldab kahesuunalist ja püsivat suhtlust veebipõhiste rakenduste vahel. Traditsioonilises HTTP protokollis käib suhtlus nii, et klient saadab serverrakendusse päringu, millele vastatakse. Kui klient soovib uut infot, peab klient tegema uue päringu. WebSocketi puhul saadab klientrakendus päringu serverrakendusse, et avada ühendus, mille järel saab serverrakendus otsustada, kas avada ühendus või mitte. Kui nõustatakse, siis püsib ühendus püsib nii kaua, kuni klientrakendus või serverrakendus selle katkestab [63].

Järgnevalt tuuakse näide, kuidas ühendatakse mänguga. Kui mängu vaataja avab mängu, siis tehakse päring, et avada ühendus. Ühenduse loomise käigus kontrollitakse mälus olevast sõnastikust, kas vastava mänguga grupp on loodud ning vajaduse korral luuakse see. Gruppi kuuluvad loodud ühenduste identifikaatorid, mis on vajalik sellepärast, et andmete uuendamisel peab saatma uuenenud informatsiooni kõikidele loodud ühendustele, mis on seotud konkreetse mänguga.

Järgnevalt kirjeldatakse, kuidas toimub punktide uuemine. Kui mängu läbiviija ehk sisse logitud kasutaja lisab punkti, siis serverrakenduse andmebaasis uuendatakse vastav mänguseis ning saadetakse info uuenenud seisuga kohta kõikidele selle mängu vaatajatele. Sarnane loogika toimub ka statistika uuendamise puhul.

### 5.2.5 Turvalisus

Andmetele ligipääsu kontrollimiseks on kasutusele võetud JWT (*JSON Web Token*), mida päringuid tehes rakendatakse. JWT on avatud standard, mida kasutatakse turvaliseks infovahetuseks kahe poole vahel. Tavaliselt kasutatakse seda autoriseerimiseks, kuid ka turvaliseks info edastamiseks [64].

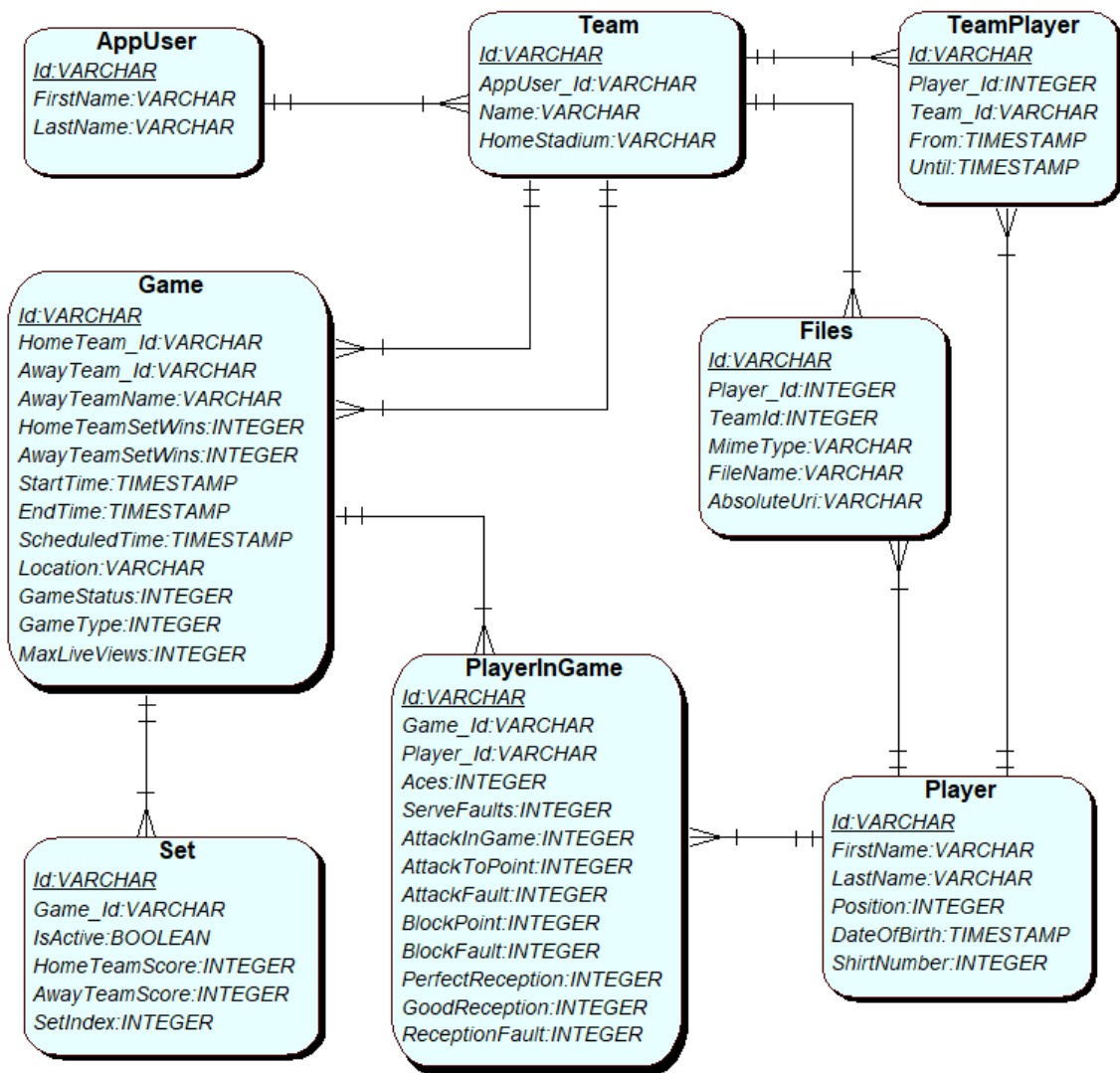
JWT koosneb kolmest osast: päis, sisu ja allkiri. Päis sisaldab informatsiooni märgise ja allkirjastamise algoritmi kohta. Sisu on kirjas „väited“ (*claim*). Loodud rakenduse puhul on sisuks kasutajaandmed. Allkirja genereerimiseks on eelnevad osad kodeeritud Base64URL-skeemi järgi ning lisatud on salajane võti. Seejärel kasutatakse päises täpsustatud krüpteerimisstrateegiat. Allkirja olemasolu on oluline, et tagada loodud märgise õigsus [64].

Loodud kontrollid on märgitud annotatsiooniga: [*Authorize(AuthenticationSchemes = JwtBearerDefault.AuthenticationScheme)*]. Lõpp-punktid, millele on anonüümsetel kasutajatel ligipääs, on märgistatud vastavalt: [*AllowAnonymous*]. Loodud lahenduse puhul on sellisteks lõpp-punktideks näiteks kõikide mängude ja võistkondade päringud. Samuti tasub märkida, et anonüümsetel kasutajatel on ligipääs ainult osale GET-päringuga seotud lõpp-punktidele. See tähendab, et sisse logimata kasutajad ei saa andmeid sisestada ega muuta.

Vajalikes meetodites kontrollitakse päringu tegija võistkonda kuulumist. JWT küll tagab, et kasutaja on autenditud, kuid tuleb kontrollida, et kasutajal on õigus teha mängude või võistkondadega seotud tegevusi.

### 5.3 Andmebaas

Arendusprotsessi alguses koostati olemisuhte diagramm, mis annab ülevaate loodavast andmebaasist. Olemisuhte diagramm on kõige laiemalt levinud meetodika andmemudelite koostamiseks ja kirjelduse esitamiseks [65]. Diagrammis on kirjeldatud tabelid, veerud ja nende andmetüübid (joonis 5).



Joonis 5. Olemisuhte diagramm.

Autor kasutas andmebaasi loomiseks *code-first*-lähendamist. Algselt loodi olemisuhte diagramm ning seejärel loodi C# koodis vajalikud objektid. Primaarvõtmena kasutati GUID (*Globally Unique Identifier*) andmetüüpi. GUID on 128-bitine sõne, mis on unikaalne üle kõigi tabelite [66].

## 6 Testimine

Peatükis kirjeldatakse lõputöö raames valminud veebirakenduse prototüübi testimist. Esmalt testis autor kogu veebirakenduse funktsionaalsust ning seejärel kasutati rakendust võrkpallivõistkonnas.

### 6.1 Autori testimine

Uue lahenduse skoobis seati tingimuseks, et veebirakenduse prototüüp peab olema kasutatav mobiiltelefonis, tahvelarvutis ja sülearvutis. Üheks tähtsamaks tingimuseks oli see, et kõik kasutajaliidese elemendid mahuksid ekraanile ära. Samuti pidi statistika sisestamine olema võimalikult mugav. Autor testis kõikides seadmetes alati kasutajaliidese elementide kuvamist, kui lisati uus vaade või mõni uus funktsionaalsus.

Prototüüpi testiti järgnevates seadmetes ja brauserites:

- Mobiiltelefonis iPhone 12, mis kasutab iOS-versiooni 16.3.1 ning brauserit Google Chrome (versioon 111.0.5563.101).
- Mobiiltelefonis OnePlus 5, mis kasutas OxygenOS (versioon 11.1.2.2) operatsioonisüsteemi ning brauserit Google Chrome (112.0.5615.47).
- Tahvelarvutis Lenovo M10, mis kasutas Android 10 operatsioonisüsteemi ja brauserit Google Chrome (versioon 110.0.5563.115).
- Windows 10 sülearvutit ning brausereid Google Chrome (versioon 111.0.5563.148) ja Mozilla Firefox (versioon 112.0).

Kõikides eelnimetatud seadmetes testiti järgnevaid funktsionaalsusi:

- kasutaja lisamine administraatori poolt;
- tavakasutaja sisselogimine;

- võistkonna lisamine, muutmine ning logo lisamine ja muutmine;
- mängijate lisamine, muutmine ning profiilipildi lisamine ja muutmine;
- mängude loomine ja kustutamine;
- mängu alustamine ja lõpetamine;
- uue geimi alustamine;
- punktide lisamine kodu- ja vastasvõistkonnale;
- statistika sisestamine;
- statistika ja punktiseisu automaatne uuenedmine;

## 6.2 Võrkpallivõistkonna testimine

Võrkpallivõistkond testis rakendust Rahvaliiga finaaltourniiril, mis toimus 07.04.2023. Rakendust kasutati kahe mängu puhul, kus autori võrkpallivõistkond mängis. Esimese mängu puhul kasutati rakendust arvutis ning teise mängu puhul mobiiltelefonil.

Testijaks oli inimene, kes varem on teinud võistkonna kohta statistikat paberil. Üldpildis oldi loodud lahendusega rahul, kuid anti ka näpunäiteid, mis võiks olla paremini. Peamiselt sooviti parandusi, mis on seotud visuaalse poolega, et mängijate särginumbrid oleksid statistikat sisestades suuremad ning statistikat sisestades antaks kasutajale paremini tagasisidet, et nuppu vajutati. Üheks peamiseks funktsionaalsuseks, mida peab arendama, on dokumendi genereerimine mängude kohta.

Esimese mängu puhul oli vaatajate arv viis ning teise mängu puhul kolm.

Võrkpallivõistkond tundis loodud lahenduse vastu huvi ning autor jätkab selle arendamisega. Prototüübi käigus on peamised funktsionaalsused valminud, misjärel viiakse sisse parandused ja antakse selle parema kasutamise soovitused. Võistkond soovib rakendust kasutada Rahvaliiga hooajal 2023/2024.

## 7 Tulemused

Lõputöö raames loodi veebirakenduse prototüüp, kus saab jälgida informatsiooni võrkpallimängude kohta ning sisestada reaajas punktiseise ja statistikat. Rakenduse skoobis püstitatud nõuded said kõik täidetud.

Autori võrkpallivõistkond on lahendusest huvitatud ning veebirakendust arendatakse edasi, võttes arvesse testimise tulemusel saadud tagasisidet võistkonna eestvedajalt ja statistikult. Praegu on võrkpallivõistkonnas statistika täitmine veel paberi peal, kuid järgmiseks hooajaks soovitakse rakendus kasutusele võtta. Autor ei ole loodud lahenduse kohta teistele võistkondadele reklaami teinud.

Veebirakenduse prototüübi olulisemad vaated on lisas 2.

Serverrakenduse ja klientrakenduse lähtekood on avalikult saadaval GitHubi hoidlas, mille lingid asuvad lisas 3.

## 8 Kokkuvõte

Lõputöö eesmärk oli arendada veebirakenduse prototüüp, kus statistika ja punktiseisu täitmine on toodud võrkpallivõistkonna tasemele, mis annab võimaluse võistkonnal saada suuremat kajastust, sest informatsiooni mängude kohta saab jälgida reaalajas.

Arenduse algul analüüsiti olemasolevaid lahendusi, mis täidavad peamisi funktsionaalsusi ning võeti arvesse nende nõrku ja tugevaid külgi. Seejärel pandi kirja loodava lahenduse skoop ja nõuded ning analüüsiti võimalikke tehnoloogiaid, millega veebirakenduse prototüüpi arendada. Analüüsi järel otsustas autor kasutada eesrakenduse arendamiseks TypeScripti programmeerimiskeelt ja Reacti teeki. Serverrakenduses otsustati kasutada C# programmeerimiskeelt ja .NET raamistikku ning andmebaasiks valiti PostgreSQL.

Lõputöö tulemusena valmis veebirakenduse prototüüp, kus võrkpallivõistkonnad saavad reaalajas sisestada statistikat ja punktiseisu ning huvilistel on võimalik seda jälgida.

Veebirakenduse prototüübi loomise järel testis autor seda eri seadmetes ja veebilehitsejates, et veenduda rakenduse kõikide funktsioonide toimimises. Seejärel testis võrkpallivõistkond prototüüpi võrkpalliturniiril kahes mängus. Järgnevas arutelus saadi tagasisidet rakenduse funktsioonide ja kasutajaliidese kohta. Võistkond andis soovitusi, mis funktsioone võiks juurde arendada ning mis võiks visuaalselt paremini olla. Võistkond soovib veebirakendust kasutama hakata Rahvaliiga hooajal 2023/2024.



## Kasutatud kirjandus

- [1] Eesti Võrkpalli Liit, „Rahvaliiga 21. hooajal asub võistlustulle 120 võistkonda!“, 08.10.2022. [Võrgumaterjal]. [https://volley.ee/rahvaliiga-21-hooajal-asub-voistlustulle-120-voistkonda/?fbclid=IwAR0psisgcw8YtSWA4d0oQ8GD9SOrIx3KYw\\_f94Hra9\\_F1kudpW9Y84uXTQw](https://volley.ee/rahvaliiga-21-hooajal-asub-voistlustulle-120-voistkonda/?fbclid=IwAR0psisgcw8YtSWA4d0oQ8GD9SOrIx3KYw_f94Hra9_F1kudpW9Y84uXTQw). [Vaadatud 07.04.2023].
- [2] Eesti Võrkpalli Liit, „Rahvaliiga AJAKAVA 2022/2023 (vastused) - Google'i arvutustabelid“. [Võrgumaterjal]. <https://docs.google.com/spreadsheets/d/1-b---TID7mDhAEH2Rma58NOYmxeXq1Dmm08TIJvscpA/edit#gid=1670431374>. [Vaadatud 07.04.2023].
- [3] Eesti Võrkpalli Liit, „Juhend“, 2022. [Võrgumaterjal]. <https://volley.ee/harrastajad/saalivorkpall/rahvaliiga/2022-2023/juhend/>. [Vaadatud 10.04.2023].
- [4] Data Project, „Data Volley 4“. [Võrgumaterjal]. <https://www.dataproject.com/Products/EN/en/Volleyball/DataVolley4>. [Vaadatud 10.04.2023].
- [5] Data Project, „Software for the scouting and analysis of volleyball matches“. [Võrgumaterjal]. [https://bli.is/wp-content/uploads/2019/09/dvwin2007\\_handbook.pdf](https://bli.is/wp-content/uploads/2019/09/dvwin2007_handbook.pdf). [Vaadatud 10.04.2023].
- [6] Data Project, „Scout and video analysis for your needs“, 2017. [Võrgumaterjal]. [https://dataprojectweb.blob.core.windows.net/productbrochure/VO/English/ENG\\_DV4\\_BROCHURE.pdf](https://dataprojectweb.blob.core.windows.net/productbrochure/VO/English/ENG_DV4_BROCHURE.pdf). [Vaadatud 10.04.2023].
- [7] Data Project, „Data Project“. [Võrgumaterjal]. <https://www.dataproject.com/GLOBAL/en/Volleyball>. [Vaadatud 10.04.2023].
- [8] Data Project, „Data Volley 4“. [Võrgumaterjal]. <https://www.dataproject.com/Products/GLOBAL/en/Volleyball/DataVolley4#Feature3>. [Vaadatud 10.04.2023].
- [9] Data Project, „Click and Scout“. [Võrgumaterjal]. <https://www.dataproject.com/Products/GLOBAL/en/Volleyball/ClickAndScout>. [Vaadatud 10.04.2023].
- [10] Tooming Solutions, „Skoor.me User Guide“, 2017, [Võrgumaterjal]. [https://static.skoor.ee/pdf/skoor\\_manual.pdf](https://static.skoor.ee/pdf/skoor_manual.pdf). [Vaadatud 10.04.2023].
- [11] Tooming Solutions, „Volleyball statistics | Skoor.ee“. [Võrgumaterjal]. <https://skoor.ee/s/pages/pricing>. [Vaadatud 10. aprill 2023].
- [12] Lucaino Rila, „What is the significance of a flowchart? [UPDATED] | Culture Online - UCL – University College London“. [Võrgumaterjal]. <https://www.ucl.ac.uk/culture-online/ask-expert/your-questions-answered/what-significance-flowchart-updated>. [Vaadatud 10. aprill 2023].
- [13] AVP, „Beach Volleyball vs Indoor Volleyball: Rules and Regulations - AVP Beach Volleyball“, 15.07.2019. [Võrgumaterjal]. <https://avp.com/news/beach-volleyball-vs-indoor-volleyball-rules-and-regulations/>. [Vaadatud 11. aprill 2023].

- [14] Jiri Popelka, „3 Reasons Why Volleyball Statistics Are Important“, 13.11.2017. [Võrgumaterjal]. <https://volleycountry.com/training/3-reasons-volleyball-statistics-important>. [Vaadatud 11.04.2023].
- [15] AVCA, „Making Volleyball Statistics Simple“. [Võrgumaterjal]. [www.avca.org](http://www.avca.org) [Vaadatud 11.04.2023]
- [16] HPE, „What is a Front-End Application?“. [Võrgumaterjal]. <https://www.hpe.com/us/en/what-is/front-end-applications.html>. [Vaadatud 12.04.2023].
- [17] Stack Overflow, „Stack Overflow Developer Survey 2022“, 2022. [Võrgumaterjal]. <https://survey.stackoverflow.co/2022/#technology-most-popular-technologies> [Vaadatud 11. aprill 2023].
- [18] MDN, „What is JavaScript?“. [Võrgumaterjal]. [https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First\\_steps/What\\_is\\_JavaScript](https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript). [Vaadatud 11.04.2023].
- [19] Cordenne Brewster, „What Is JavaScript Used For?“. [Võrgumaterjal]. <https://www.trio.dev/blog/what-is-javascript-used-for>. [Vaadatud 11.04.2023].
- [20] TypeScript, „TypeScript“. [Võrgumaterjal]. <https://www.typescriptlang.org/>. [Vaadatud 11.04.2023].
- [21] Nihar Raval, „TypeScript vs JavaScript: Which One Should You Choose?“ . [Võrgumaterjal]. <https://radixweb.com/blog/typescript-vs-javascript>. [Vaadatud 11.04.2023].
- [22] Harshil Agrawal, „What is TypeScript and why should you use it?“, 18.08.2022. [Võrgumaterjal]. <https://www.contentful.com/blog/what-is-typescript-and-why-should-you-use-it/>. [Vaadatud 11.04.2023].
- [23] David Herbert, „What is React.js? (Uses, Examples, & More)“, 27.06.2022. [Võrgumaterjal]. <https://blog.hubspot.com/website/react-js>. [Vaadatud 12. aprill 2023].
- [24] KnowledgeHut, „What is a React component?“, 25.11.2022. [Võrgumaterjal]. <https://www.knowledgehut.com/blog/web-development/react-component>. [Vaadatud 12.04.2023].
- [25] Chinmayee Deshpande, „The Best Guide to Know What Is React“, 7.02.2023. [Võrgumaterjal]. <https://www.simplilearn.com/tutorials/reactjs-tutorial/what-is-reactjs>. [Vaadatud 12.04.2023].
- [26] Eric Baer, „1. What Is React? - What React Is and Why It Matters?“. [Võrgumaterjal]. <https://www.oreilly.com/library/view/what-react-is/9781491996744/ch01.html>. [Vaadatud 12.04.2023].
- [27] StackOverflow, „Stack Overflow Developer Survey 2022“, 2022. [Võrgumaterjal]. <https://survey.stackoverflow.co/2022/#technology-most-popular-technologies> [Vaadatud 12.04.2023].
- [28] Vue, „Introduction“. [Võrgumaterjal]. <https://vuejs.org/guide/introduction.html>. [Vaadatud 12.04.2023].
- [29] Kofi Group, „7 Reasons Why VueJS Is So Popular“. [Võrgumaterjal]. <https://www.kofi-group.com/7-reasons-why-vuejs-is-so-popular/>. [Vaadatud 12.04.2023].
- [30] Angular, „What is Angular?“. [Võrgumaterjal]. <https://angular.io/guide/what-is-angular>. [Vaadatud 12.04.2023].
- [31] Ian Deed, „Angular vs AngularJS: What Are the Differences?“, 22.03.2023. [Võrgumaterjal]. <https://www.pangea.ai/angular-resources/angular-vs-angularjs-what-are-the-differences/>. [Vaadatud 12.04.2023].

- [32] Microsoft, „A tour of C#“, 13.03.2023. [Võrgumaterjal].  
<https://learn.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/>. [Vaadatud 12.04.2023].
- [33] Microsoft, „What is .NET?“. [Võrgumaterjal]. <https://dotnet.microsoft.com/en-us/learn/dotnet/what-is-dotnet>. [Vaadatud 12.04.2023].
- [34] IBM, „What is Java?“. [Võrgumaterjal]. <https://www.ibm.com/topics/java> [Vaadatud 12.04.2023].
- [35] Spring, „Spring Security“. [Võrgumaterjal]. <https://docs.spring.io/spring-security/reference/index.html>. [Vaadatud 12.04.2023].
- [36] Spring, „Accessing Relational Data using JDBC with Spring“. [Võrgumaterjal]. <https://spring.io/guides/gs/relational-data-access/>. [Vaadatud 12.04.2023].
- [37] StackOverflow, „Stack Overflow Developer Survey 2022“. [Võrgumaterjal]. <https://survey.stackoverflow.co/2022/#technology-most-popular-technologies>. [Vaadatud 12.04.2023].
- [38] PHP, „What is PHP? - Manual“. [Võrgumaterjal]. <https://www.php.net/manual/en/intro-what-is.php>. [Vaadatud 12.04.2023].
- [39] Oracle, „What Is a Database?“. [Võrgumaterjal]. <https://www.oracle.com/database/what-is-database/>. [Vaadatud 13.04.2023].
- [40] Lihtmee, „Difference Between Object Oriented Database and Object Relational Database - Pediaa.Com“, 02.05.2019. [Võrgumaterjal]. <https://pediia.com/difference-between-object-oriented-database-and-object-relational-database/>. [Vaadatud 13.04.2023].
- [41] PostgreSQL, „PostgreSQL: About“. [Võrgumaterjal]. <https://www.postgresql.org/about/>. [Vaadatud 13.04.2023].
- [42] Stack Overflow, „Stack Overflow Developer Survey 2022“. [Võrgumaterjal]. <https://survey.stackoverflow.co/2022/#most-popular-technologies-database-prof>. [Vaadatud 13.04.2023].
- [43] MySQL, „MySQL 8.0 Reference Manual“. [Võrgumaterjal]. <https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html>. [vaadatud 13.04.2023].
- [44] Mark Smallcombe, „PostgreSQL vs MySQL: The Critical Differences“, 15.02.2023. [Võrgumaterjal]. <https://www.integrate.io/blog/postgresql-vs-mysql-which-one-is-better-for-your-use-case/>. [Vaadatud 13.04.2023].
- [45] SQLite, „SQLite Home Page“. [Võrgumaterjal]. <https://sqlite.org/index.html>. [Vaadatud 13.04.2023].
- [46] SQLite, „Appropriate Uses For SQLite“. [Võrgumaterjal]. <https://www.sqlite.org/whentouse.html>. [Vaadatud 13.04.2023].
- [47] Mark Drake, „SQLite vs MySQL vs PostgreSQL“, 9.03.2022. [Võrgumaterjal]. <https://www.digitalocean.com/community/tutorials/sqlite-vs-mysql-vs-postgresql-a-comparison-of-relational-database-management-systems>. [Vaadatud 13.04.2023].
- [48] Ant Design Mobile, „Ant Design Mobile“. [Võrgumaterjal]. <https://mobile.ant.design/>. [Vaadatud 18.04.2023].
- [49] Ant Design, „Components Overview - Ant Design“. [Võrgumaterjal]. <https://ant.design/components/overview/>. [Vaadatud 18.04.2023].
- [50] Redux, „Getting Started with Redux“. [Võrgumaterjal]. <https://redux.js.org/introduction/getting-started>. [Vaadatud 18.04.2023].
- [51] Simplilearn, „Why Should You Use a Router in React.js?“, 20.02.2023. [Võrgumaterjal]. [https://www.simplilearn.com/tutorials/reactjs-tutorial/routing-in-reactjs#what\\_is\\_a\\_react\\_router](https://www.simplilearn.com/tutorials/reactjs-tutorial/routing-in-reactjs#what_is_a_react_router). [Vaadatud 18.04.2023].

- [52] Lodash, „Lodash“. [Võrgumaterjal]. <https://lodash.com/>. [Vaadatud 18.04.2023].
- [53] Microsoft, „ASP.NET Core SignalR JavaScript client“, 27.06.2022. [Võrgumaterjal]. <https://learn.microsoft.com/en-us/aspnet/core/signalr/javascript-client?view=aspnetcore-7.0&tabs=visual-studio>. [Vaadatud 18.04.2023].
- [54] „What is Entity Framework?“. [Võrgumaterjal]. <https://www.entityframeworktutorial.net/what-is-entityframework.aspx>. [Vaadatud 17.04.2023].
- [55] „3-Tier Architecture in C#“. [Võrgumaterjal]. <https://www.javatpoint.com/3-tier-architecture-in-c-sharp>. [Vaadatud 16.04.2023].
- [56] GeeksForGeeks, „Data-Access Layer “. [Võrgumaterjal]. <https://www.geeksforgeeks.org/data-access-layer/>. [Vaadatud 16.04.2023].
- [57] „Repository Pattern in ASP.NET Core REST API“. [Võrgumaterjal]. <https://www.pragimtech.com/blog/blazor/rest-api-repository-pattern/>. [Vaadatud 16.04.2023].
- [58] „What is the Business-Logic Layer“. [Võrgumaterjal]. <https://www.tutorialspoint.com/what-is-the-business-logic-layer>. [Vaadatud 16.04.2023].
- [59] Alexande S. Gillis, „What is REST API (RESTful API)?“. [Võrgumaterjal]. <https://www.techtarget.com/searchapparchitecture/definition/RESTful-API>. [Vaadatud 16.04.2023].
- [60] Jamie Juviler, „What Is an API Endpoint? (And Why Are They So Important?)“, 27.07.2022. [Võrgumaterjal]. <https://blog.hubspot.com/website/api-endpoint> [Vaadatud 16.04.2023].
- [61] Okta, „Data Transfer Object DTO Definition and Usage | Okta“, 22.06.2022. [Võrgumaterjal]. <https://www.okta.com/identity-101/dto/>. [Vaadatud 16.04.2023].
- [62] Microsoft, „Overview of ASP.NET Core SignalR“, 14.02.2023. [Võrgumaterjal]. <https://learn.microsoft.com/en-us/aspnet/core/signalr/introduction?view=aspnetcore-7.0#transports>. [Vaadatud 16.04.2023].
- [63] GeeksForGeeks, „What is web socket and how it is different from the HTTP? - GeeksforGeeks“. [Võrgumaterjal]. <https://www.geeksforgeeks.org/what-is-web-socket-and-how-it-is-different-from-the-http/>. [Vaadatud 16.04.2023].
- [64] „JSON Web Token Introduction - jwt.io“. [Võrgumaterjal]. <https://jwt.io/introduction>. [Vaadatud 17.04.2023].
- [65] „4.1. Olemi-suhte diagramm ja andmebaaside loomine“. [Võrgumaterjal]. <https://enos.itcollege.ee/~priit/1.%20Andmebaasid/1.%20Loengumaterjalid/04/4.html>. [Vaadatud 17.04.2023].
- [66] Alexander S. Gillis, „What is GUID?“. [Võrgumaterjal]. <https://www.techtarget.com/searchwindowsserver/definition/GUID-global-unique-identifier>. [Vaadatud 17.04.2023].

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

Mina, Henry Veetamm

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Reaalajaline punktiseisu ja statistika veebirakenduse prototüüp võrkpallivõistkondadele“, mille juhendaja on Meelis Antoi
  - 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.

15.05.2023

---

<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 loomingu tegevusega ning lõputöö kaas- või ühisautor(id) ei ole andnud lõputööd kaitsvale üliõpilasele kindlaksmääratud tähtajaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktile 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.

## Lisa 2 – Klientrakenduse vaated

Avaleht Mängud Võistkonnad Minu võistkond Minu mängud Logi välja

Info


Nimi: Vki Tsunaami Kodusaal: Tallinna Tehnika G

Muuda võistkonda  
Lae üles tiimi logo  
Lisa mängija  
+ Loo mäng  
Muuda parooli

<b>Rauno Pihel</b> #3 Sidemängija Vanus: 26, 15.09.1996	<b>Andres Palsi</b> #8 Temporündaja Vanus: 28, 30.11.1994	<b>Kardo Liitmäe</b> #10 Nurgaründaja Vanus: 25, 23.03.1998
<b>Siim Melles</b> #2 Diagonaalründaja Vanus: 24, 16.03.1999	<b>Henry Veetamm</b> #28 Temporündaja Vanus: 23, 28.04.1999	<b>Kristo Mettus</b> #7 Libero Vanus: 25, 22.11.1997
<b>Sander Melles</b> #5 Nurgaründaja Vanus: 25, 16.06.1997	<b>Taavi Nõmm</b> #10 Libero Vanus: 27, 13.11.1995	<b>Egon Vaiksaar</b> #1 Nurgaründaja Vanus: 28, 07.02.1995

Av
ängud
Lo

**Info**



**Rauno Pihel**


**#3 Sidemängija**  
Vanus: 26, 15.09.1996

**Siim Melles**

**#2 Diagonaalründaja**  
Vanus: 24, 16.03.1999

**Sander Melles**

**#5 Nurgaründaja**  
Vanus: 25, 16.06.1997



Nimi	Henry Veetamm
Positsioon	#28 Temporündaja
Vanus	23

---

**Rünnakud**

Kõik rünnakud	13
Rünnakud punktiks	10
Rünnakuvead	2

---

**Servid**

Servipunktid	11
Servivead	1

---

**Blokid**

Blokipunktid	2
Blokivead	2

---

**Vastuvõtt**

Kõik vastuvõtud	8
Suurepärased vastuvõtud	5
Head vastuvõtud	1
Vastuvõtu vead	2

**Tegevused**

Tehnika Gümnaasium

**tmäe**

**10 Nurgaründaja**  
Vanus: 25, 23.03.1998


**ettus**

**7 Libero**  
Vanus: 25, 22.11.1997

**Egon Vaiksaar**

**#1 Nurgaründaja**  
Vanus: 28, 07.02.1995


Vaatajaid: 1 Lõpeta mäng Alusta uue geimiga



**Vk Tsunaami**

+1
-1

18 : 9



**Plaan C**

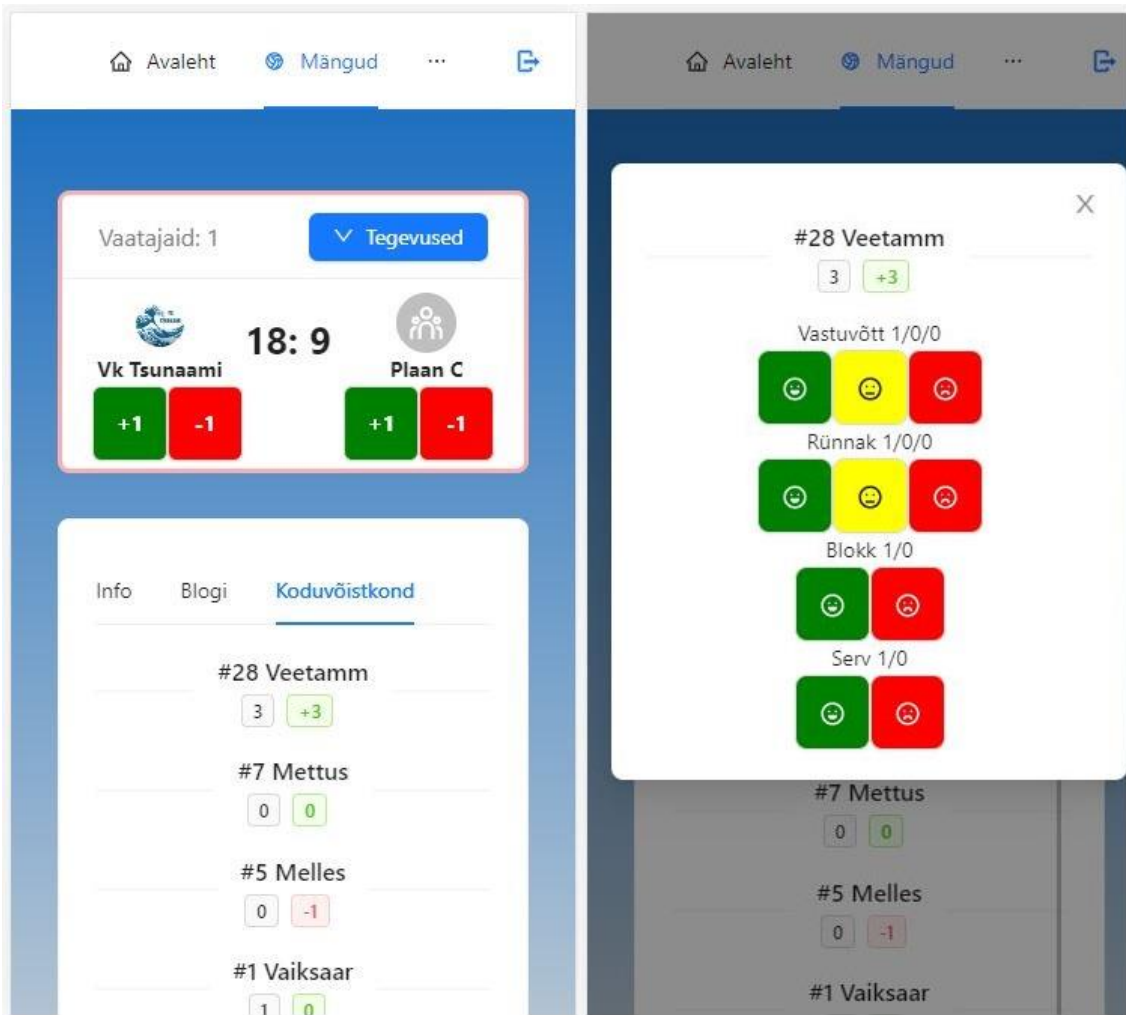
+1
-1

---

Info [Blogi](#) [Koduvõistkond](#)

<p>#28 Veetamm</p> <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">3</div> <span style="background-color: #e0ffe0; padding: 2px 5px;">+3</span>	<p>Rünnak 1/0/0</p> <div style="display: flex; justify-content: space-around;"> <div style="background-color: green; color: white; padding: 5px; border-radius: 50%; text-align: center;">⊖</div> <div style="background-color: yellow; color: black; padding: 5px; border-radius: 50%; text-align: center;">☹</div> <div style="background-color: red; color: white; padding: 5px; border-radius: 50%; text-align: center;">⊖</div> </div>	<p>Blokk 1/0</p> <div style="display: flex; justify-content: space-around;"> <div style="background-color: green; color: white; padding: 5px; border-radius: 50%; text-align: center;">⊖</div> <div style="background-color: red; color: white; padding: 5px; border-radius: 50%; text-align: center;">⊖</div> </div>	<p>Vastuvõtt 1/0/0</p> <div style="display: flex; justify-content: space-around;"> <div style="background-color: green; color: white; padding: 5px; border-radius: 50%; text-align: center;">⊖</div> <div style="background-color: yellow; color: black; padding: 5px; border-radius: 50%; text-align: center;">☹</div> <div style="background-color: red; color: white; padding: 5px; border-radius: 50%; text-align: center;">⊖</div> </div>	<p>Serv 1/0</p> <div style="display: flex; justify-content: space-around;"> <div style="background-color: green; color: white; padding: 5px; border-radius: 50%; text-align: center;">⊖</div> <div style="background-color: red; color: white; padding: 5px; border-radius: 50%; text-align: center;">⊖</div> </div>
<p>#7 Mettus</p> <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">0</div> <span style="background-color: #e0ffe0; padding: 2px 5px;">0</span>	<p>Rünnak 0/0/0</p> <div style="display: flex; justify-content: space-around;"> <div style="background-color: green; color: white; padding: 5px; border-radius: 50%; text-align: center;">⊖</div> <div style="background-color: yellow; color: black; padding: 5px; border-radius: 50%; text-align: center;">☹</div> <div style="background-color: red; color: white; padding: 5px; border-radius: 50%; text-align: center;">⊖</div> </div>	<p>Blokk 0/0</p> <div style="display: flex; justify-content: space-around;"> <div style="background-color: green; color: white; padding: 5px; border-radius: 50%; text-align: center;">⊖</div> <div style="background-color: red; color: white; padding: 5px; border-radius: 50%; text-align: center;">⊖</div> </div>	<p>Vastuvõtt 1/1/0</p> <div style="display: flex; justify-content: space-around;"> <div style="background-color: green; color: white; padding: 5px; border-radius: 50%; text-align: center;">⊖</div> <div style="background-color: yellow; color: black; padding: 5px; border-radius: 50%; text-align: center;">☹</div> <div style="background-color: red; color: white; padding: 5px; border-radius: 50%; text-align: center;">⊖</div> </div>	<p>Serv 0/0</p> <div style="display: flex; justify-content: space-around;"> <div style="background-color: green; color: white; padding: 5px; border-radius: 50%; text-align: center;">⊖</div> <div style="background-color: red; color: white; padding: 5px; border-radius: 50%; text-align: center;">⊖</div> </div>
<p>#5 Melles</p> <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">0</div> <span style="background-color: #ffe0e0; padding: 2px 5px;">-1</span>	<p>Rünnak 0/1/1</p> <div style="display: flex; justify-content: space-around;"> <div style="background-color: green; color: white; padding: 5px; border-radius: 50%; text-align: center;">⊖</div> <div style="background-color: yellow; color: black; padding: 5px; border-radius: 50%; text-align: center;">☹</div> <div style="background-color: red; color: white; padding: 5px; border-radius: 50%; text-align: center;">⊖</div> </div>	<p>Blokk 0/0</p> <div style="display: flex; justify-content: space-around;"> <div style="background-color: green; color: white; padding: 5px; border-radius: 50%; text-align: center;">⊖</div> <div style="background-color: red; color: white; padding: 5px; border-radius: 50%; text-align: center;">⊖</div> </div>	<p>Vastuvõtt 0/0/0</p> <div style="display: flex; justify-content: space-around;"> <div style="background-color: green; color: white; padding: 5px; border-radius: 50%; text-align: center;">⊖</div> <div style="background-color: yellow; color: black; padding: 5px; border-radius: 50%; text-align: center;">☹</div> <div style="background-color: red; color: white; padding: 5px; border-radius: 50%; text-align: center;">⊖</div> </div>	<p>Serv 0/0</p> <div style="display: flex; justify-content: space-around;"> <div style="background-color: green; color: white; padding: 5px; border-radius: 50%; text-align: center;">⊖</div> <div style="background-color: red; color: white; padding: 5px; border-radius: 50%; text-align: center;">⊖</div> </div>
<p>#1 Vaiksaar</p> <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">1</div> <span style="background-color: #e0ffe0; padding: 2px 5px;">0</span>	<p>Rünnak 0/0/0</p> <div style="display: flex; justify-content: space-around;"> <div style="background-color: green; color: white; padding: 5px; border-radius: 50%; text-align: center;">⊖</div> <div style="background-color: yellow; color: black; padding: 5px; border-radius: 50%; text-align: center;">☹</div> <div style="background-color: red; color: white; padding: 5px; border-radius: 50%; text-align: center;">⊖</div> </div>	<p>Blokk 1/0</p> <div style="display: flex; justify-content: space-around;"> <div style="background-color: green; color: white; padding: 5px; border-radius: 50%; text-align: center;">⊖</div> <div style="background-color: red; color: white; padding: 5px; border-radius: 50%; text-align: center;">⊖</div> </div>	<p>Vastuvõtt 0/0/0</p> <div style="display: flex; justify-content: space-around;"> <div style="background-color: green; color: white; padding: 5px; border-radius: 50%; text-align: center;">⊖</div> <div style="background-color: yellow; color: black; padding: 5px; border-radius: 50%; text-align: center;">☹</div> <div style="background-color: red; color: white; padding: 5px; border-radius: 50%; text-align: center;">⊖</div> </div>	<p>Serv 0/1</p> <div style="display: flex; justify-content: space-around;"> <div style="background-color: green; color: white; padding: 5px; border-radius: 50%; text-align: center;">⊖</div> <div style="background-color: red; color: white; padding: 5px; border-radius: 50%; text-align: center;">⊖</div> </div>
<p>#8 Palsi</p> <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">1</div> <span style="background-color: #e0ffe0; padding: 2px 5px;">+1</span>	<p>Rünnak 0/0/0</p> <div style="display: flex; justify-content: space-around;"> <div style="background-color: green; color: white; padding: 5px; border-radius: 50%; text-align: center;">⊖</div> <div style="background-color: yellow; color: black; padding: 5px; border-radius: 50%; text-align: center;">☹</div> <div style="background-color: red; color: white; padding: 5px; border-radius: 50%; text-align: center;">⊖</div> </div>	<p>Blokk 0/0</p> <div style="display: flex; justify-content: space-around;"> <div style="background-color: green; color: white; padding: 5px; border-radius: 50%; text-align: center;">⊖</div> <div style="background-color: red; color: white; padding: 5px; border-radius: 50%; text-align: center;">⊖</div> </div>	<p>Vastuvõtt 0/0/0</p> <div style="display: flex; justify-content: space-around;"> <div style="background-color: green; color: white; padding: 5px; border-radius: 50%; text-align: center;">⊖</div> <div style="background-color: yellow; color: black; padding: 5px; border-radius: 50%; text-align: center;">☹</div> <div style="background-color: red; color: white; padding: 5px; border-radius: 50%; text-align: center;">⊖</div> </div>	<p>Serv 1/0</p> <div style="display: flex; justify-content: space-around;"> <div style="background-color: green; color: white; padding: 5px; border-radius: 50%; text-align: center;">⊖</div> <div style="background-color: red; color: white; padding: 5px; border-radius: 50%; text-align: center;">⊖</div> </div>
<p>#3 Pihel</p> <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">0</div> <span style="background-color: #e0ffe0; padding: 2px 5px;">0</span>	<p>Rünnak 0/0/0</p> <div style="display: flex; justify-content: space-around;"> <div style="background-color: green; color: white; padding: 5px; border-radius: 50%; text-align: center;">⊖</div> <div style="background-color: yellow; color: black; padding: 5px; border-radius: 50%; text-align: center;">☹</div> <div style="background-color: red; color: white; padding: 5px; border-radius: 50%; text-align: center;">⊖</div> </div>	<p>Blokk 0/0</p> <div style="display: flex; justify-content: space-around;"> <div style="background-color: green; color: white; padding: 5px; border-radius: 50%; text-align: center;">⊖</div> <div style="background-color: red; color: white; padding: 5px; border-radius: 50%; text-align: center;">⊖</div> </div>	<p>Vastuvõtt 0/0/0</p> <div style="display: flex; justify-content: space-around;"> <div style="background-color: green; color: white; padding: 5px; border-radius: 50%; text-align: center;">⊖</div> <div style="background-color: yellow; color: black; padding: 5px; border-radius: 50%; text-align: center;">☹</div> <div style="background-color: red; color: white; padding: 5px; border-radius: 50%; text-align: center;">⊖</div> </div>	<p>Serv 0/0</p> <div style="display: flex; justify-content: space-around;"> <div style="background-color: green; color: white; padding: 5px; border-radius: 50%; text-align: center;">⊖</div> <div style="background-color: red; color: white; padding: 5px; border-radius: 50%; text-align: center;">⊖</div> </div>





Vaatajaid: 1

Lõpeta mäng

Alusta uue geimiga



Vlk Tsunaami

+1

-1

18 : 9



Plaan C

+1

-1

Info [Blogi](#) Koduvõistkond

#1 Egon Vaiksaar : eksis serviga

#8 Andres Palsi : tõi serviga punkti

#7 Kristo Mettus : võttis palli vastu

#5 Sander Melles : ründas, aga pall jäi mängu

#5 Sander Melles : eksis rünnakul

#1 Egon Vaiksaar : tõi blokiga punkti.

#7 Kristo Mettus : tegi suurepärase vastuvõtu

#28 Henry Veetamm : tõi serviga punkti

#28 Henry Veetamm : tegi suurepärase vastuvõtu

#28 Henry Veetamm : tõi blokiga punkti.

#28 Henry Veetamm : tõi rünnakuga punkti.

## **Lisa 3 – Klientrakenduse ja serverrakenduse lähtekood**

Klientrakenduse lähtekood: <https://github.com/HenryVeetamm/livescorefront>

Serverrakenduse lähtekood: <https://github.com/HenryVeetamm/livescoreback>