

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Evert Padari 179160IADB

Virtuaalautoralli võistluste portaali arendus

Bakalaureusetöö

Juhendaja: Tiina Zingel
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: Evert Padari

12.05.2023

Annotatsioon

Käesoleva bakalaureusetöö eesmärgiks on luua veebiportaal, mille abil oleks võimalik läbi viia virtuaalautoralli võistluseid. Vajadus uue rakenduse järele tuleneb võistlusplatvormide korraldamisvõimaluste erinevustest ja alternatiivse lahenduse puudumisest.

Töös vaadeldakse teemaga kokkupuutuvaid lahendusi, et nende tugevusi ära kasutada uue portaali loomisel. Eelmainitud analüüsile ja autori varasemale võistluste korraldamise kogemusele toetudes seatakse nõuded uuele lahendusele. Töös põhjendatakse rakenduse loomiseks kasutatavate tehnoloogiate valikuid.

Töö tulemusena valmib veebiportaal, mille abil on võimalik korraldada virtuaalautoralli võistluseid ja sõitjana saada hea ülevaade toimuvatest ralliüritustest. Organiseerija saab seadistada võistluseid oma soovide kohaselt. Loodud rakendus saab hakkama punktitablete koostamisega, võttes arvesse korraldaja poolt seatud reegleid. Võistlejad saavad loodud veebikeskkonna abil võistluste kohta käiva informatsiooni kätte ühest kohast.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 30 leheküljel, 5 peatükki, 14 joonist, 4 tabelit.

Abstract

Development of a Portal for Virtual Rally Competitions

The aim of this bachelor's thesis is to create a web portal that provides the means to conduct virtual rally competitions. The need for a new application arises from the differences in competition platform configuration options and the lack of alternative solutions.

The thesis examines existing solutions related to the topic in order to leverage their strengths in creating the new portal. Based on the aforementioned analysis and the author's previous experience in organizing competitions, requirements are set for the new solution. The thesis also justifies the choices of technologies used in developing the application.

As a result of this work, a web portal will be developed that allows for the organization of virtual rally competitions and provides participants with a comprehensive overview of ongoing rally events. Organizers will be able to configure competitions according to their preferences. The created application will be capable of generating scoreboards, taking into account the rules set by the organizer. Competitors will be able to access all relevant information about the competitions from an unified source.

The thesis is in Estonian and contains 30 pages of text, 5 chapters, 14 figures, 4 tables.

Lühendite ja mõistete sõnastik

ACID	<i>Atomicity, Consistency, Isolation, Durability</i> , andmebaasitransaktsioonide omadused
API	<i>application programming interface</i> , rakendusliides
<i>backend</i>	tagarakendus
BLL	<i>Business Logic Layer</i> , äriloogika kiht
CI/CD	<i>continuous integration/continuous delivery</i> , pidev integratsioon/pidev tarnimine
CRA	Create React App, tööriist, millega saab luua React.js projekte
CRUD	<i>create, read, update, delete</i> , andmete ladustamise põhitoimingud
CSS	<i>Cascading Style Sheets</i> , kaskaadlaadistik
CSV	<i>comma-separated values</i> , failivorming, kus andmed on üksteisest komadega eraldatud
DAL	<i>Data Access Layer</i> , andmepääsu kiht
EF Core	<i>Entity Framework Core</i> , ORM raamistik ASP.NET Corele
ERD	<i>Entity Relationship Diagram</i> , olemi-suhte diagramm, andmemudelite esitamise meetod
<i>frontend</i>	eesrakendus
HTML	<i>HyperText Markup Language</i> , veebilehtede märgenduskeel
IDE	<i>Integrated Development Environment</i> , integreeritud programmeerimiskeskond
JSON	<i>JavaScript Object Notation</i> , andmevahetusvorming, mis kasutab andmeobjektide edastamiseks inimloetavat teksti
JVM	<i>Java Virtual Machine</i> , Java virtuaalmasin
JWT	<i>JSON Web Token</i> , väljapakutud standard turvaliseks tõendite (<i>claims</i>) vahetamiseks
MM	autoralli maailmameistrivõistlused
MVC	<i>Model-View-Controller</i> , mudel-vaade-kontroller tarkvara arhitektuurimuster
npm	Node Package Manager, Node.js paketiholdus tööriist
npx	Node Package eXecute, npm pakettide käivitaja

OCR	<i>optical character recognition</i> , optiline märgituvastus
ORM	<i>object-relational mapping</i> , objekt-relatsiooniline kaardistamine
RBR	<i>Richard Burns Rally</i> , autoralli simulaator/mäng
<i>repository</i>	andmehoidla
RSF	rallysimfans.hu, veebileht, mille abil toimuvad virtuaalautoralli võistlused
SPA	<i>single page application</i> , üheleherakendus
SQL	<i>Structured Query Language</i> , andmebaasi päringukeel
UOW	<i>unit of work</i> , tööühik
vastendi	<i>mapper</i> , andmevahetusobjektide teisendaja
WCAG	<i>Web Content Accessibility Guideline</i> , veebi sisu juurdepääsetavussuunised
WORA	<i>write once, run everywhere</i> , slogan, mille sõnum viitab sellele, et programmeerimiskeel on kasutatav platvormide üleselt
WRC	<i>World Rally Championship</i> , autoralli maailmameistrivõistlused
võistlusplatvorm	virtuaalse autoralli simulaator või mäng, kus toimub võidusõit

Sisukord

1 Sissejuhatus	11
1.1 Taust ja aktuaalsus	12
1.2 Probleemipüstitus	12
1.3 Eesmärk	13
2 Analüüs	14
2.1 Olemasolevad sarnased lahendused	14
2.1.1 Veebileht ewrc-results.com	14
2.1.2 Autoralli maailmameistrivõistluste lehekülg	14
2.1.3 Veebirakendus rallysimfans.hu	15
2.2 Nõuete kirjeldamine	16
2.2.1 MoSCoW prioriseerimise meetod	16
2.2.2 Funktsionaalsed nõuded	17
2.2.3 Mittefunktsionaalsed nõuded	19
2.3 Tehnoloogiate valik	20
2.3.1 Rakenduse arhitektuuriline suund	20
2.3.2 Tagarakenduse tehnoloogiate valik	20
2.3.3 Eesrakenduse tehnoloogiate valik	22
2.3.4 Andmebaasihaldussüsteemi valik	24
2.3.5 Töövahendite valik	25
3 Ülevaade rakendusest ja arendusprotsessist	26
3.1 Rakenduse üldine kirjeldus	26
3.1.1 Katsetulemuste salvestamine rakendusse	27
3.1.2 Meetmed pettuse piiramiseks	27
3.1.3 Tulemuste põhjal punktide arvutamine	28
3.2 Andmebaasi arendus	29
3.2.1 Andmebaasi loomine ja andmete manipuleerimine	29
3.2.2 Andmemudel	29
3.3 Serveripoolse rakenduse arendus	31

3.4 Kliendipoolse rakenduse arendus	33
3.4.1 Eesrakenduse loomine	33
3.4.2 Eesrakenduse ja kasutajaliidese tutvustus	34
4 Tulemus	39
4.1 Edasiarenduse võimalused.....	39
5 Kokkuvõte	40
Kasutatud kirjandus	41
Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks	45

Jooniste loetelu

Joonis 1. Rakenduse voog.	26
Joonis 2. Punktide arvestamise voodiagramm.....	28
Joonis 3. Andmebaasi olemi-suhte diagramm.....	30
Joonis 4. Kihiline arhitektuur.	31
Joonis 5. npx käsk React.js rakenduse loomiseks.	33
Joonis 6. Võistluste nimekirja vaade.	34
Joonis 7. Võistlusetappide nimekirja vaade.	34
Joonis 8. Võistlusetapi lisamise plokk (olles korraldaja rollis).	35
Joonis 9. Etapitulemuste detailvaade.....	35
Joonis 10. Katsetulemuse lisamise plokk (olles korraldaja rollis).....	36
Joonis 11. Registreerunute nimekirja vaade.	36
Joonis 12. Võistlusetapi võistlusklasside haldamise vaade (olles korraldaja rollis).	37
Joonis 13. Võistluse punktisüsteemi vaade.	37
Joonis 14. Punktitabeli vaade.	38

Tabelite loetelu

Tabel 1. Autoralli maailmameistrivõistluste etapipunktide süsteem.....	13
Tabel 2. MoSCoW meetodi kategooriate kirjeldused.	17
Tabel 3. Tagarakenduse tehnoloogiate võrdlus.	22
Tabel 4. Eesrakenduse tehnoloogiate võrdlus.	23

1 Sissejuhatus

Virtuaalautoralli võistluseid korraldatakse mitmetel võistlusplatvormidel. Puudub ühine rakendus, mis koondaks erinevate platvormide võistlused ühte kohta kokku. Seetõttu on võistlustel osalemine keerulisem, kui olema peaks.

Ralliürituste korraldamine ilma abistava rakendusega võib kiiresti muutuda keeruliseks ja ajamahukaks, kui on vaja arvestada võistlustega, mis toetavad mitut võistlusklassi, kus on palju osalejaid, millele on vaja kohandada spetsiaalne punktisüsteem jms.

Eesmärk on luua rakendus, kus korraldaja saaks lihtsasti hallata rallivõistluseid ja sõitja saaks hea ülevaate toimuvatest üritustest. Rakenduse abil on soov parandada virtuaalrallivõistluste kogemust – nii võistlussarjades osaleja kui ka organiseerija perspektiivist.

Lõputööga valmiva rakenduse sihtrühmaks on virtuaalse autoralli huvilised, kes soovivad korraldada e-autoralli võistluseid või nendest osa võtta.

Esimeses peatükis tutvustatakse probleemi tausta ja seatakse eesmärk. Teises peatükis analüüsitakse olemasolevate sarnaste lahenduste võimekusi. Seejärel kaardistatakse ja prioriseeritakse nõuded loodavale rakendusele tuginedes analüüsi tulemustele ja autori varasemale võistluste korraldamise kogemusele. Järgneb tehnoloogiate analüüs, selgitamiseks välja kõige sobivamad vahendid rakenduse arenduseks. Kolmandas peatükis kirjeldatakse arendusprotsessi ja loodud rakendust. Viimases peatükis hinnatakse valminud lahenduse sobivust nõuetele ja tuuakse välja loodud süsteemi puudused.

1.1 Taust ja aktuaalsus

Virtuaalautospordis kasutatakse võidusõidu simuleerimiseks tarkvara, kus võistlejad läbivad raja virtuaalses keskkonnas. Auto juhtimiseks kasutavad sõitjad spetsiaalset riistvara: arvuti/mängukonsooliga ühendatud rooli, pedaale, käigukange, käsipidureid jms. On võimalik sõita ka tavalise mängupuldi või klaviatuuriga, aga üldiselt eelistavad sõitjad parema kogemuse (ja tulemuse) saavutamiseks kasutada eraldi roolimiseadet [1]. Rool annab tee profiili ja autole mõjuvate jõudude kohta tagasisidet (*force feedback*), mis omakorda muudab sõidukogemuse realistlikumaks [2].

Virtuaalautosport on kiiresti arenev ala – COVID-19 pandeemia järel on virtuaalsõitude vastu huvi tundma hakanud üha rohkem harrastajaid aga ka professionaalseid võidusõitjaid [3]. Eestis alustati virtuaalautospordi meistrivõistluste korraldamisega 2021. aastal [4]. E-autoralli Eesti meistrivõistluste esimesel hooajal oli osalejaid ligi 130 [5].

Virtuaalsed autorallivõistlused, nagu ka päriselus läbiviidavad rallivõistlused, koosnevad etappidest, mis omakorda koosnevad kiiruskatsetest. Levinud võistluse formaadiks e-autoralli puhul on see, et korraldaja kehtestab etappidele algusajad ja lõpuajad. Võistlejad peavad läbima kiiruskatsed omale sobival hetkel, kuid jääma etteantud ajavahemikku. Võistlushooaja vältel jagatakse iga etapi eest punkte. Punktide arvestus käib võistlussarjades erinevalt. [6] [7]

1.2 Probleempüstitus

E-autoralli võistluste puhul toimub kasutajate registreerimine üritusele, vajadusel registreerimiste muutmine, võistlusklasside määramine ja punktide jagamine. Igal platvormil (võidusõidusimulaatoril ja -mängul) on eelmainitud tegevuste läbiviimise võimalused erinevad, mistõttu oleks otstarbekas luua üldine lahendus.

Korraldamise poolelt – punktide arvestamine võib muutuda keeruliseks, kui võistlussarjas osaleb palju sõitjaid. Võistlusplatvormidel kasutatakse osalejate identifitseerimiseks kasutajanimed, mis võivad aja jooksul muutuda. Sõitja pärisnimi ei pruugi ühtida tema kasutajanimiga, aga korraldajal võib olla soov koostada finišiprotokolle ja punktitableid võistlejate pärisnimede järgi. Olenevalt

võistlusplatvormi konfigureerimise võimalustest võib olla vajalik kontrollida seda, kas sõitja kasutas raja läbimiseks sama võistluskategooria autot, millega ta võistlusele registreeris. Punktide arvutamine käsitsi, kasutades tabeltöötlusprogramme, võib sellistel juhtudel võtta liigselt aega ning olla veaohklik.

Punktide arvutamisel tuginedes ainult võistlusplatvormist jooksvalt kättesaadavatele tulemustele peab arvesse võtma asjaolu, et mingi aja jooksul võivad need tulemused muutuda kättesaamatuks ja hakata võistluste korraldamist häirima.

Võistlustel võib osaleda misiganes arv sõitjaid. Sellest tulenevalt võib korraldaja soovida populaarset WRC (*World Rally Championship*) punktide jagamise süsteemi (Tabel 1) [8] asemel kasutada oma nõudmistele kohandatud skoorisüsteemi. Näiteks tuhande osavõtjaga e-autoralli võistlusel võib WRC punktisüsteemi kasutamine tekitada keskpärasel sõitjal motivatsioonipuuduse hooaja lõpetamiseks, kuna punkte teenib vaid kiirem 1% sõitjatest.

Tabel 1. Autoralli maailmameistrivõistluste etapipunktide süsteem.

Lõpp-positsioon	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.
Punktide arv	25	18	15	12	10	8	6	4	2	1

1.3 Eesmärk

Lõputöö raames on soov luua veebiportaal, mis muudaks virtuaalautoralli võistluste korraldamise lihtsamaks, nendel osalemise mugavamaks ja võistlusalaga tutvust tegevatele uutele tulijatele rohkem ligipääsetavamaks.

Rakendus peaks hõlmama kogu võistluse korralduslikku poolt, et vähendada organiseerimisele kuluvat aega ning korraldamise käigus tekkivate vigade arvu. Võistlusel osaleja peab saama ühest kohast kätte kogu informatsiooni, mis käib võistluse kohta. Loodav rakendus peab salvestama võistluste tulemused andmebaasi tähtajatult, et vähendada sõltuvust kolmandatest osapooltest (sõiduplatvormid võivad omalt poolt iga hetk kustutada kogu võistluste andmestiku).

Lõputöö eesmärk on analüüsida vajadusi, kaardistada nõuded ja nende põhjal luua sobiv veebirakendus.

2 Analüüs

Enne rakenduse arendamise alustamist teostatakse analüüs, millega seatakse paika nõuded rakendusele. Vaadatakse üle hetkel olemasolevad sarnased lahendused ning tuuakse välja funktsionaalsused, mida võiks rakendada ka loodavas lahenduses. Nõuete kaardistamiseks kasutatakse ka autori varasemat kogemust e-autoralli võistluste korraldamisega. Rakenduse arenduseks kasutatavate tehnoloogiate valimiseks tehakse samuti analüüs, et leida sobivad vahendid lõputöö eesmärkide täitmiseks.

2.1 Olemasolevad sarnased lahendused

Kuigi virtuaalralli on populaarne, siis avalikke rakendusi, mis viiksid kogu organisatoorse loogika ühtsesse veebikeskkonda, ei ole. Seega vaadeldakse siin peatükis teemaga kokkupuutuvaid lahendusi, milledest oleks võimalik uue rakenduse arendamisel eeskuju võtta.

2.1.1 Veebileht ewrc-results.com

eWRC on rallisündmuste, -sõitjate ja -autode andmebaas, kuhu kogutakse infot ülemaailmselt. Platvorm on tavakasutajale vaid vaatamiseks. Andmeid sisestavad ainult rakenduse administraatorid. eWRC lehtedel ei kuvata infot virtuaalautoralli võistluste kohta.

Veebileht paistab silma selle poolest, et kiiruskatsete lehel on samal ajal näha konkreetse katse tulemused ja etapi üldseis peale antud kiiruskatset. Tänu sellele saab portaali külastaja tõetruu pildi võistluse hetkeseisust ilma üleliigse vaevata – ei ole vajadust liikuda mõnele teisele lehele, et üldtulemusi näha. Katsete tulemusi saab sorteerida võistlusklassi järgi, mis annab kasutajale hea ülevaate ralli seisust iga kategooria kohta.

2.1.2 Autoralli maailmameistrivõistluste lehekülg

WRC ametlikul veebilehel wrc.com kuvatakse infot autoralli maailmameistrivõistluste (MM) kohta. Hooaegade kaupa saab näha andmeid võistluskalendri, etappide,

võistlustulemuste ja punkttabelite kohta. Etappide tulemusi ja punktiarvestuse tabelleid on võimalik sorteerida võistlusklassi järgi. Lisaks individuaalsele sõitjate arvestusele peetakse WRCs eraldi punktiarvet meeskondade ja kaardilugejate arvestuses. Punktitali lehel on kasutajal võimalik filtrist valida just see arvestus, mida ta näha soovib. Tänu minimalistlikule disainile on lehekülje kasutamine lihtne ja navigeerimine erinevate lehtede vahel kerge.

WRCs jagatakse punkte ka etapi viimase kiiruskatse (punktikatse) eest [9]. Punktikatse punktid on toodud tabeliseisus välja eraldi – „25+5“ tähistab, et põhisõidu eest sai sõitja sel etapil 25 ja punktikatse eest 5 punkti. Tabelis võib näha ka väärtuseid „R.“ ja „-“, mis märgistavad vastavalt sõitja katkestamist (*retired*) ja mittestartimist. Punktide eristamine sellisel moel annab kasutajale parema arusaama sõitjate sooritustest.

Aastatel 2021 ja 2022 toimusid ka ametlikud WRC e-autoralli võistlused vastavalt WRC 9 ja WRC 10 mängudel [10]. Funktsionaalsuselt jääb virtuaalralli veebikeskkond alla päris MM-ralli omale, kuna puudub võimekus võistlusklasside filtreerimiseks.

2.1.3 Veebirakendus rallysimfans.hu

Rallysimfans.hu (RSF) on Ungari veebikeskkond, mis keskendub virtuaalrallile, spetsiifiliselt *Richard Burns Rally* (RBR) simulaatoril korraldatavatele võistlustele. Leht hõlmab endas enamjaolt kõike, mida on eduka võistluse korraldamise jaoks vaja. Võistlushooaegu saab paindlikult üles sättida – autoklassid, punktisüsteem, litsentsikontroll – kõik on seadistatav. Kasutajal, kel on RSF keskkonnas täitunud 50 sõidukilomeetrit, avaneb võimalus luua enda võistluseid.

Rakenduses on iga võistleja kohta profiil, kus näeb viimast viite rallietappi, kus ta osalenud on. Profiilil kuvatakse ka statistikat sõitja kohta: rallidel osalemise arvu, lõpetatud rallide osakaalu ja arvu.

RSFi negatiivseks küljeks on see, et see toetab vaid RBR simulaatorit – teiste võistlusplatvormide puhul seda kasutada ei ole võimalik. Veebilehe disain on iganenud, mistõttu võib see uusi sõitjaid eemale tõrjuda.

2.2 Nõuete kirjeldamine

Kavandava portaali sujuvaks arenduseks pannakse kirja nõuded rakendusele ja seatakse need olulisuse järjekorda. Funktsionaalsed nõuded esitatakse kasutajalugude näol. Kasutajalugu on lõppkasutaja vaatest esitatud rakenduse funktsionaalsuse kirjeldus, mis tavaliselt jälgib malli: „<teatud tüüpi kasutajana> soovin <teha midagi>, et <saavutada mingi tulemus>“ [11].

Antud töö puhul on defineeritud kaks kasutajatüüpi:

- võistleja – kasutaja, kes tunneb virtuaalautoralli võistluste vastu huvi ja soovib nendest osa võtta;
- korraldaja – kasutaja, kes tegeleb võistluste organiseerimisega.

Nõuete kirjeldamisel on arvestatud asjaoluga, et ka korraldaja võib olla võistleja – ehk kõik kasutajalood, mis on esitatud kasutajatüübile „võistleja“, kehtivad ka „korraldaja“ tüüpi kasutajale.

2.2.1 MoSCoW prioriseerimise meetod

Kõik kirjapandud nõuded on rakenduse eesmärgi täitmiseks olulised, kuid lõputöö kontekstis on aeg piiratud, mistõttu tuleb seada prioriteedid, et etteantud ajavahemikus saaks teostatud võimalikult tähtis hulk rakenduse funktsionaalsusest. Suure hulga (20 ja rohkem) nõuete puhul peetakse kasutusväärseks tehnikaks nõuetele numbrilise hinnangu määramist [12]. Tulenevalt sellest osutub nõuete prioritseerimiseks valituks MoSCoWi meetod.

MoSCoW meetod on juhtimises, ärianalüüsis, projektijuhtimises ja tarkvaraarenduses kasutatav prioriteetide seadmise tehnika, mis võimaldab osapooltel jõuda ühise arusaamani käsilolevate nõuete rahuldamise tähtsusest. Meetodi arendas välja Dai Clegg. [13]

Kuna analüüsi teostatakse MoSCoW meetodi abil, jagatakse nõuded nelja kategooriasse: *must have* (peab olema), *should have* (peaks olema), *could have* (võiks olla) ja *won't have* (ei pea olema). Täpsem kirjeldus kategooriate kohta [14] on välja toodud allpool (Tabel 2).

Tabel 2. MoSCoW meetodi kategooriate kirjeldused.

Kategooria	Kirjeldus
<i>Must have</i>	Kirjeldab nõuet, mis peab olema rahuldatud, et saaks lõpplahenduse lugeda õnnestunuks.
<i>Should have</i>	Tähistab kõrge prioriteediga nõuet, mis tuleks võimaluse korral lahendusse kaasata. Sageli on tegemist kriitilise nõudega, mida saab vajadusel rahuldada teistel viisidel.
<i>Could have</i>	Märgistab nõuet, mida peetakse soovitatavaks, aga mitte hädavajalikuks. Rahuldatakse kui ressursid seda lubavad.
<i>Won't have</i>	Kujutab endas nõuet, mille puhul on kokku lepitud, et praeguses iteratsioonis seda ei rahuldata, kuid mida võidakse tulevikus uuesti kaaluda.

2.2.2 Funktsionaalsed nõuded

Funktsionaalsed nõuded kirjeldavad süsteemi käitumist ja rakenduse poolt kasutatavat andmestikku. Nõuded näitavad, milliseid toiminguid peab rakendus olema võimeline teostama [14]. Alljärgnevalt on välja toodud loodavale rakendusele seatud funktsionaalsed nõuded, sorteerituna MoSCoW kategooriate järgi.

Must have:

- võistlejana soovin sisselogimata näha infot käimasolevate võistluste kohta, et saaksin otsustada, kas võtta nendest osa või mitte
- võistlejana soovin luua kasutaja ja logida rakendusse sisse, et saaksin end võistlustele registreerida
- võistlejana soovin panna end kirja võistlussarjadesse, et saaksin võistlustest osa võtta
- võistlejana soovin näha, kes on võistlusele registreerinud, et teada kellega konkureerin
- võistlejana soovin näha, millal võistlusetapid algavad ja lõppevad, et saaksin planeerida oma aega
- võistlejana soovin näha, milliste reeglite alusel jaotatakse etappide eest punkte, et teaksin mille nimel pingutada
- võistlejana soovin näha kiiruskatsete tulemusi ja punktitableid, sorteerida neid võistlusklasside kaupa, et võrrelda sõitjate tulemusi
- korraldajana soovin luua uusi võistluseid, lisada võistlustele etappe ja etappidele kiiruskatseid, et sõitjatel oleks võimalik neist osa võtta

- korraldajana soovin, et rakendus peaks erinevate võistluste ja võistlusklasside kohta punktiarvestusi, et ei tuleks seda ise teha
- korraldajana soovin hallata punktisüsteeme, et punktide jagamine toimuks mõistlikult
- korraldajana soovin häälestada võistluskategooriaid – valida millised klassid on võistlusel lubatud ja millised võistlusautod kuuluvad nendesse klassidesse, et luua sõitjatele valikuvõimalus osalemaks oma sobivas kategoorias
- korraldajana soovin peale etapi lõppu sisestada katse/ralli tulemused rakendusse, et süsteem saaks arvutada punktiseisu ja et sõitjatel oleks võistlusplatvormiväliselt ülevaade tulemustest

Should have:

- võistlejana soovin vahetada oma registreerimisel märgitud kasutajanime, et see vastaks alati minu kasutajanimele võistlusplatvormis
- korraldajana soovin peale etapi lõppu sisestada kõik katsetulemused korraga, kasutades selleks JSON (*JavaScript Object Notation*) või CSV (*comma-separated values*) formaadis sisendit, et tulemuste lisamine mööduks kiiremini
- korraldajana soovin, et võimalusel toimuks võistlusplatvormilt tulemuste laadimine rakendusse automaatselt, et vähendada käsitöö mahtu

Could have:

- korraldajana soovin muuta etapi/katse tulemusi, et parandada võimalikke vigu
- korraldajana soovin anda sõitjatele trahvipunkte, et karistada reeglite rikkumist
- võistlejana soovin näha võistluste nimekirjas lühikokkuvõtet iga võistluse kohta – TOP3 sõitjate nimesid ja nende hetke punktiskoori, et saaksin võistluse tulemustest üldpildi kätte ilma detailvaatesse navigeerimata
- võistlejana soovin näha võistluse detailvaates lühikokkuvõtet iga etapi kohta – TOP3 sõitjate nimesid ja nende sõiduaega, et saaksin etapi tulemustest üldpildi kätte ilma etapi detailvaatesse navigeerimata

Won't have:

- võistlejana soovin registreerimisel üles anda meeskonna, et teenida punkte meeskondlikus arvestuses

2.2.3 Mittefunktsionaalsed nõuded

Mittefunktsionaalsed nõuded määratlevad tarkvarasüsteemi kvaliteediattribuudid. Nõuded hõlmavad omadusi nagu jõudlus, turvalisus, hooldatavus, skaleeritavus ja kasutusmugavus [15]. Järgnevalt on kirjeldatud rakendusele seatud mittefunktsionaalsed nõuded, sorteerituna MoSCoW kategooriate järgi.

Must have:

- rakendus peab autoriseerima autenditud kasutajaid vastavalt kasutajarollidele
- rakendus peab olema kergesti hooldatav – kood peab olema puhas ja arhitektuuriliselt loogiline
- rakendus peab olema kasutatav enamike veebibrauseritega
- rakendust peab saama kasutada mistahes ekraanisuurusega seadmest

Should have:

- rakendusel peab olema lihtne ning intuiitiivne kasutajaliides
- rakendust peab olema võimalik CI/CD (*continuous integration/continuous delivery*) protsesside abil kasutajatele kättesaadavaks teha

Could have:

- rakendus peab toetama internatsionaliseerimist ja lokaliseerimist

Won't have:

- rakendus peab vastama WCAG 2.2 (*Web Content Accessibility Guidelines*) kriteeriumitele

2.3 Tehnoloogiate valik

Kasutusse võetavate tehnoloogiate valimisel võetakse arvesse erinevaid aspekte. Tehnoloogiate valikul lähtutakse nende funktsionaalsusest, populaarsusest ja autori kogemusest valitud tehnoloogiaga.

2.3.1 Rakenduse arhitektuuriline suund

Üldiselt on veebirakenduste arendamiseks kaks lähenemisviisi [16]:

- nõ klassikaline veebirakendus, kus enamik äriloogikat koos kasutajale kuvatava HTMLi genereerimisega toimub serveris;
- üheleherakendused (SPA, *single page application*), kus kasutajaliidese loogika toimub brauseris, aga suhtlus veebiserveriga toimub kasutades rakendusliideseid (API, *application programming interface*).

Lõputöös valmiv rakendus plaanitakse koostada SPA printsiibil, mis tähendab, et andmeid laetakse serverist kasutaja tegevuse peale dünaamiliselt, ilma et tehtaks lehe värskendus (*refresh*). See lähenemine muudab veebilehe kasutamise sujuvamaks ja kasutajakogemuse paremaks [17].

2.3.2 Tagarakenduse tehnoloogiate valik

Viis populaarsemat tagarakenduste (*backend*) arendamiseks kasutatud programmeerimiskeelt on JavaScript, TypeScript, Python, Java ja C# [18] – siin peatükis kirjeldatakse neid, et tuua välja keelte eelised ja puudused, et rakenduse loomiseks valida sobivaim variant.

JavaScript on keel mida kasutatakse interaktiivsete veebilehtede loomiseks. Raamistike nagu Node.js, Express.js, Next.js jm abil on võimalik arendada serveripoolseid rakendusliideseid ka JavaScriptiga [19]. JavaScripti kasutamine nii kliendipoolses kui serveripoolses rakenduses võimaldab arendajatel sujuvalt liikuda ees- ja tagarakenduste arendamise vahel. JavaScripti kasutamine võib dünaamilise tüüpimise tõttu olla veaohalik, kui just ei kasutata kokkulepitud häid tavaid [20]. JavaScripti teekide hooletu installimise puhul võib tekkida nn sõltuvuste põrgu (*dependency hell*) – rakendus saab muutuda väga mahukaks, kuigi lisati vaid üks teek [21].

TypeScript on JavaScriptil põhinev keel, mis toetab tugevat tüüpimist [22]. Dünaamiliselt tüübitud keelte (JavaScript) puhul võib arendamine muutuda ajamahukaks kui on vaja hulgaliselt kontrollida süsteemi osade vahelisi sidemeid. TypeScript annab süsteemiliideste ühildatavuse kohta jookvalt tagasisidet, tänu millele saab arendaja säästa aega ja olla kindel, et tema muudatused ei tekita rakenduses vigu [23]. TypeScript koodi käivitamiseks transpileeritakse see esmalt JavaScriptiks [24].

Python on kõrgtasemeline programmeerimiskeel, mis sobib mitmete ülesannete lahendamiseks – veebiarendus, andmeanalüüs, tehisintellekt ja masinõpe jms [25]. Pythonit on lihtne õppida selle kerge süntaksi ja loetavuse tõttu [26].

Java on objektorienteeritud programmeerimiskeel, mis võimaldab lihtsasti luua taaskasutatavat koodi [27]. Javat tuntakse WORA (*write once, run everywhere*) slogani poolest, mis ideeliselt tähendab seda, et peale Java programmi arendamist ükskõik milliste vahenditega võib eeldada, et programmi saab jooksutada ükskõik millises JVMiga (*Java Virtual Machine*) varustatud seadmes [28]. Java puudusteks on üleliigne ressursikasutus ja käitusajal JVMis masinkoodi interpreteerimise võimalik negatiivne mõju jõudlusele [27] [29].

C# on objektorienteeritud keel, millega on võimalik ehitada suurel hulgal erinevaid tüüpi rakendusi, mida jooksutatakse .NET keskkonnas [30]. .NET on platvormideülene (*cross-platform*) arendusraamistik, mille abil on võimalik rakendusi kirjutada C#, F# ja Visual Basic keeltes [31]. C# üheks eeliseks on fakt, et see on staatiliselt tüübitud keel. Miinuseks saab C# puhul lugeda selle tugevat sõltuvust .NET platvormist [32].

Tagarakenduse tehnoloogia valikul võetakse arvesse ülaltoodud analüüsi keelte ja raamistike kohta. Vahendite võrdlemisel peetakse oluliseks programmeerimiskeele ja raamistiku koos käsitlemist. Autor ei pea otstarbekaks kasutada mõnda ülalmainitud programmeerimiskeelt ilma raamistikuta, kuna need pakuvad lahendusi üldlevinud probleemidele, nagu näiteks kasutajate autentimisele või suhtlusele andmebaasiga.

Allpool on toodud võrdlustabel (Tabel 3) siin peatükis käsitletud programmeerimiskeeltest (+ nende kõige populaarsema raamistiku) ja autori varasema kogemusega nendest. Populaarsuse veerg tähistab kirjeldatud raamistiku kasutajate osakaalu *Stack Overflow Developer Survey 2022* vastanute hulgas. Mida laiem on tehnoloogia kasutajaskond, seda suurem on arendajate hulk, kes on valmis jagama

teadmisi ja abistama probleemide tekkimise korral. Ka tuleviku peale tasub mõelda – kui lõputöö raames loodav rakendus peaks saama laialdaselt levinuks ja rakenduse haldamisel või uute funktsionaalsuste arendamisel hakkaks ühest arendajast vajaka jääma, oleks vajalike oskustega inimeste leidmine populaarsemate tehnoloogiate kasutamise puhul lihtsam.

Tabel 3. Tagarakenduse tehnoloogiate võrdlus.

Tehnoloogia	Kogemus keelega	Kogemus raamistikuga	Populaarsus [18]
JavaScript + Express	suur	puudub/väga väike	22,99%
TypeScript + Express	suur	puudub/väga väike	22,99%
Python + Django	väike	puudub	14,65%
Java + Spring	suur	suur	16,13%
C# + ASP.NET Core	keskmine	keskmine	34,55%

Kõik kirjeldatud tehnoloogiad sobiksid lõputöö rakenduse arendamiseks, aga kuna loodava rakenduse arendamiseks on piiratud aeg, ei ole mõistlik hakata õppima uusi programmeerimiskeeli või raamistike. Kuigi autori kogemus ei ole C# keskkonnas nii hea kui Javaga seda oleks, osutub tehnoloogia populaarsuse tõttu valituks C# koos ASP.NET Core veebiraamistikuga.

2.3.3 Eesrakenduse tehnoloogiate valik

Eesrakenduste (*frontend*) põhitehnoloogiateks on HTML (*HyperText Markup Language*), CSS (*Cascading Style Sheets*) ja JavaScript. Punktis 2.3.1 tehtud otsuse põhjal kasutada SPA arendusviisi, tuleb leida sobiv raamistik, et lahendus realiseerida.

Tuginedes *Stack Overflow Developer Survey 2022* tulemustele [18] valitakse võimalikeks kandidaatideks kolm populaarsemat SPA raamistikku: React.js, Angular ja Vue.js. Peatükis kirjeldatakse neid raamistike, et tuua välja nende eelised ja puudused, et rakenduse loomiseks valida kõige optimaalsem variant.

Kandidaatide valikul peetakse oluliseks TypeScripti – tugevalt tüübitud JavaScripti täienduse – tuge. On nõutud, et raamistik kasutaks vaikumisi TypeScripti või et seda oleks võimalik ise JavaScripti asemel kasutusele võtta. TypeScripti kasutamine muudab arenduse kiiremaks (vt peatükk 2.3.2), koodi loetavamaks ja muudatuste tegemise vähem veaohtrikumaks [33].

React.js on teek veebilehtede kasutajaliideste loomiseks [34]. React.js kasutab jõudluse optimeerimiseks virtuaalset (*virtual*) DOMi (*Document Object Model*), tänu millele uuendatakse veebilehel kuvatavat infot vaid siis kui vaja [35]. Kuivõrd React.js on teek, mitte täisfunktsionaalne raamistik nagu seda on näiteks Angular, siis ei sea ta ette ühtegi nõuet projekti struktuurile – seda saab lugeda nii eeliseks kui puuduseks [36]. Kindlate struktuurireeglite puudumine pakub rohkem paindlikust – teek ei sunni arendajat kasutama MVC (*Model-View-Controller*) või muid arhitektuurilisi mustreid. Samas võib etteantud tarkvaramustri kasutamine olla kasulik koodistruktuurist tulenevate probleemide lahendamiseks – on lihtsam saada abi internetist ja teistelt arendajatelt, kes muustriga on kokku puutunud.

Angular on TypeScriptil põhinev arendusplatvorm, mille abil on võimalik ehitada skaleeritavaid veebirakendusi [37]. Angulari eeliseks võib lugeda komponentide selget jagamist HTML mallideks, TypeScript loogikaklassideks ja CSS failideks. Angulari miinusteks loetakse tema järsku õpikõverat, uute projektide esialgseks püstipanemiseks kuluvat aega [38].

Vue.js on kasutajaliideste arendamiseks mõeldud JavaScripti raamistik. Vue.js kasutab komponentidele tuginevat programmeerimismudelit, mis aitab hõlpsalt luua nii lihtsaid kui ka kompleksseid kasutajaliideseid [39]. Tuginedes *JS web frameworks benchmark Chrome 112* katse tulemustele [40] saab öelda, et Vue.jsi jõudlus on vaatluse all olevatest raamistike omadest kõige parem. Vue.jsi puuduseks on väike kasutajaskond, mistõttu on raamistikuga kokkusobivate teekide ja lisatööriistade arv väiksem võrreldes React.jsi ja Angulariga [41].

Allpool toodud tabelis (Tabel 4) on esile toodud autori varasem kogemus käsitletavate tehnoloogiatega. Lisaks on taas toodud välja raamistike populaarsus *Stack Overflow Developer Survey 2022* tulemuste [18] põhjal.

Tabel 4. Eesrakenduse tehnoloogiate võrdlus.

Raamistik	Kogemus	Populaarsus [18]
React.js	suur	42,62%
Angular	puudub	20,39%
Vue.js	väike	18,82%

Tuginedes ülaltoodud analüüsile, autori varasemale kogemusele, lõputöö raames valmiva rakenduse arendamise ajapiirangule ja tehnoloogia populaarsusele valitakse eesrakenduse raamistikuks React.js.

2.3.4 Andmebaasihaldussüsteemi valik

Rakenduse andmete haldamiseks valitakse andmebaasisüsteem andmemudeli keerukuse, jõudlusnõuete ja arendaja kogemuse põhjal.

Peamiselt kasutatakse infosüsteemide arendamisel kahte tüüpi andmebaase [42]:

- relatsioonilisi, kus andmed on struktureeritud ja järgitakse ACID (*atomicity, consistency, isolation, durability*) transaktsiooniomadusi [43], tänu millele tagatakse andmete usaldusväärsus;
- mitterelatsioonilisi ehk NoSQL süsteeme, kus andmemudel ei ole struktureeritud, ACID põhimõtteid alati ei järgita, aga saavad hästi hakkama väga suurte andmehulkadega.

Kuna plaanitava rakenduse andmestruktuur on kindel, ega pea dünaamiliselt muutuma ja tagatud võiks olla transaktsionaalsus, valitakse sobivaks lahenduseks relatsiooniline andmebaasisüsteem. Analüüsis vaadeldakse nelja populaarsemat [18] relatsiooniliste andmebaaside haldussüsteemi: MySQL, PostgreSQL, SQLite ja Microsoft SQL Server.

MySQL on osaliselt avatud lähtekoodiga (*open-source*) andmebaasihaldussüsteem, mida haldab Oracle. MySQLi eeliseks loetakse selle lihtsat süntaksit ja väikest keerukust, negatiivseks küljeks aga SQL (*Structured Query Language*) standardist kõrvale kaldumist [44]. MySQLi *Community Edition* versioon on tasuta kättesaadav, aga paremaid võimalusi ja kasutajatuge pakuvad *Standard*, *Enterprise* ja *Cluster* variandid on tasulised – litsentsitasud algavad ~2000 eurost aastas [45].

PostgreSQL on avatud lähtekoodiga andmebaasihaldussüsteem. PostgreSQL toetab vaikumisi suurel hulgal andmetüüpe, mida teistes andmebaasides alati ei pakuta – näiteks JSON ja XML (*Extensible Markup Language*). See võimaldab teataval määral kasutada NoSQL andmebaasidest tuntud omadusi, nagu näiteks struktureerimata andmete hoiustamist. Lisaks on kasutajatel võimalik PostgreSQL süsteemides defineerida oma vajadustele kohandatud andmetüüpe. [44]

SQLite on avatud lähtekoodiga andmebaasimootor, mis erinevalt teistest tüüpilistest andmebaasisüsteemidest hoiustab andmebaasi ühes failis – seega on vajadusel andmebaasi kopeerimine ja jagamine lihtne [46]. SQLite sobib kasutamiseks väikeste andmebaaside ja väheste kasutajate puhul, kuna andmebaas toetab vaid ühte kirjutamisoperatsiooni samal ajahetkel [47].

SQL Server on Microsofti poolt arendatav andmebaasihaldussüsteem. Microsoft SQL Server toetab lihtsat kolimist pilvelahenduste (*Microsoft cloud, Azure SQL Database* või *SQL Server on Azure Virtual Machines*) peale, kui selleks vajadus peaks tekkima [44]. Microsoft SQL Serveri *Developer* ja *Express* versioone on võimalik kasutada tasuta – need on mõeldud arendamiseks, testimiseks ja kasutamiseks rakendustes, mis ei nõua suurt mälumahtu ega tugevat jõudlust. Mastaapsemaks äriliseks kasutamiseks on olemas SQL Serveri teised variandid, mille litsentsi(de) maksumus algab ~900 eurost aastas [48].

Sobivaks andmebaasihaldussüsteemiks otsustati litsentsitasude puudumise, süsteemi paindlikkuse ja autori varasema kogemuse tõttu valida PostgreSQL.

2.3.5 Töövahendite valik

Koodist tagavarakoopiate hoidmiseks otsustati kasutada versioonihaldustarkvara Git. Versioonihaldus aitab jälgida koodile tehtud muudatusi – selle abil on võimalik tagasi keerata (*revert*) mittesoovitud muutuseid. Samuti on Giti toel, luues harusid, võimalik katsetada uusi ideid, mida võib-olla põhilisse koodibaasi lisada ei tahaks.

Tagarakenduse arendamiseks kasutatakse spetsiifiliselt .NET arenduse jaoks mõeldud JetBrains Rider integreeritud programmeerimiskeskonda (IDE, *Integrated Development Environment*). Eesrakenduse loomiseks on kasutusel JetBrains IntelliJ IDEA Ultimate, millel on vaikimisi olemas tugi React.js rakenduste arendamiseks [49]. JetBrainsi tooteid kasutab autor igapäevaselt ja nende kasutusmugavuse tõttu otsustati muid variante mitte kaaluda.

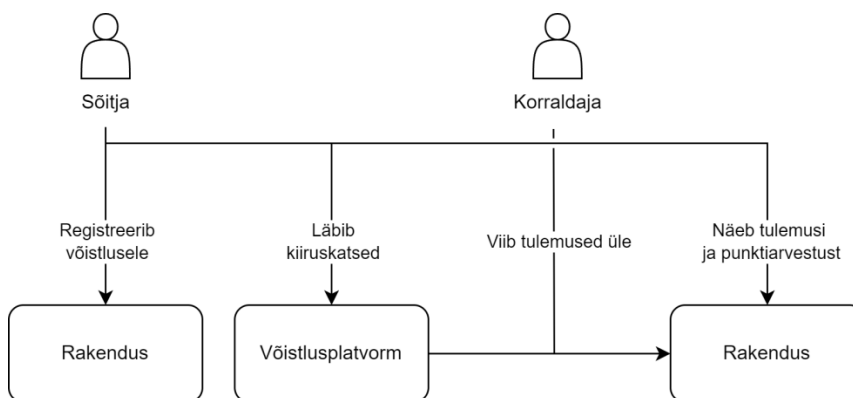
3 Ülevaade rakendusest ja arendusprotsessist

Lõputöös valmiva rakenduse töötab autor välja üksinda. Autor vastutab tarkvara analüüsi, kavandamise, arendamise ja testimise eest ainuisikuliselt. Arendusprotsess hõlmab eelmises peatükis mainitud tööriistu ja tehnoloogiaid. Peatükis kirjeldatakse virtuaalautoralli võistluseid toetava veebirakenduse loomist, sealhulgas tuues välja detailid andmebaasi struktuurist ning taga- ja eesrakenduste arendamisest.

3.1 Rakenduse üldine kirjeldus

Kogu rakenduse abil läbiviidava võistluse voog lihtsustatult ja kronoloogilises järjekorras on selline (Joonis 1):

1. Korraldaja loob uue võistluse, seadistab võistlusklassid, punktisüsteemid vastavalt oma vajadustele.
2. Sõitja registreerib veebiportaalis võistlusele.
3. Etapi kestvuse vältel valib võistleja omale sobiva aja ja läbib kiiruskatsed võistlusplatvormil.
4. Peale etapi lõppemist viib korraldaja kiiruskatsete tulemused üle võistlusplatvormilt loodavasse veebiportaali.
5. Rakendus määrab võistleja tulemusele staatuse ja teenitud punktide arvu vastavalt võistluse punktisüsteemile.
6. Sõitja näeb portaalist võistluse tulemusi ja punktiarvestust.



Joonis 1. Rakenduse voog.

3.1.1 Katsetulemuste salvestamine rakendusse

Kiiruskatsete tulemusi on hetkel võimalik rakendusse salvestada ainult manuaalselt. Tulemuse sisestamiseks rakendusse on vaja nelja parameetrit: kiiruskatse järjekorranumbrit, sõitja kasutajanime võistlusplatvormil, kiiruskatse aega ja auto nime, millega sõitja kiiruskatse läbis – need saab korraldaja võtta võistlusplatvormi veebilehelt või võistlusplatvormi rakendusest. Kasutajanime järgi seostatakse katsetulemus sõitja registreerimisega, mis võimaldab võistlejal läbida iga etapp erineva kasutajanimega – kui ta seda soovib.

Töö skooopi arvestades ei jõutud automaatse tulemuste importimiseni, kuigi seda võimaldab kindlasti näiteks Dirt Rally 2.0 võistlusplatvorm – sel on olemas API, kust on võimalik pärida katsete tulemusi. Kuna kõikidel võistlusplatvormidel sellist avalikku rakendusliidest ei ole, saab alternatiivina kasutada võistlustulemuste laadimiseks näiteks veebikoorijaid (*web scraper*), kui tulemused on saadaval võistlusplatvormi veebilehel või OCR (*optical character recognition*) tööriistu, kui tulemustele pääseb ligi ainult võistlusplatvormi rakenduse kaudu.

3.1.2 Meetmed pettuse piiramiseks

Pettuse – võistleja poolt ebaeetiliselt saavutatud edu – vähendamiseks on tavaliselt ehitatud võistlusplatvormidele süsteemid, mis aitaksid sohitegemist piirata [50]. Loodud rakenduse kontekstis eeldatakse, et võistlusplatvormilt kättesaadavad tulemused on usaldusväärsed.

Küll aga peab silmas pidama loodud veebiportaali turvalisust – võistluste tulemusi peab saama sisestada ja uuendada ainult korraldaja. Tavasõitjatel või sisselogimata kasutajatel peab olema võimalus neid andmeid näha, aga mitte muuta.

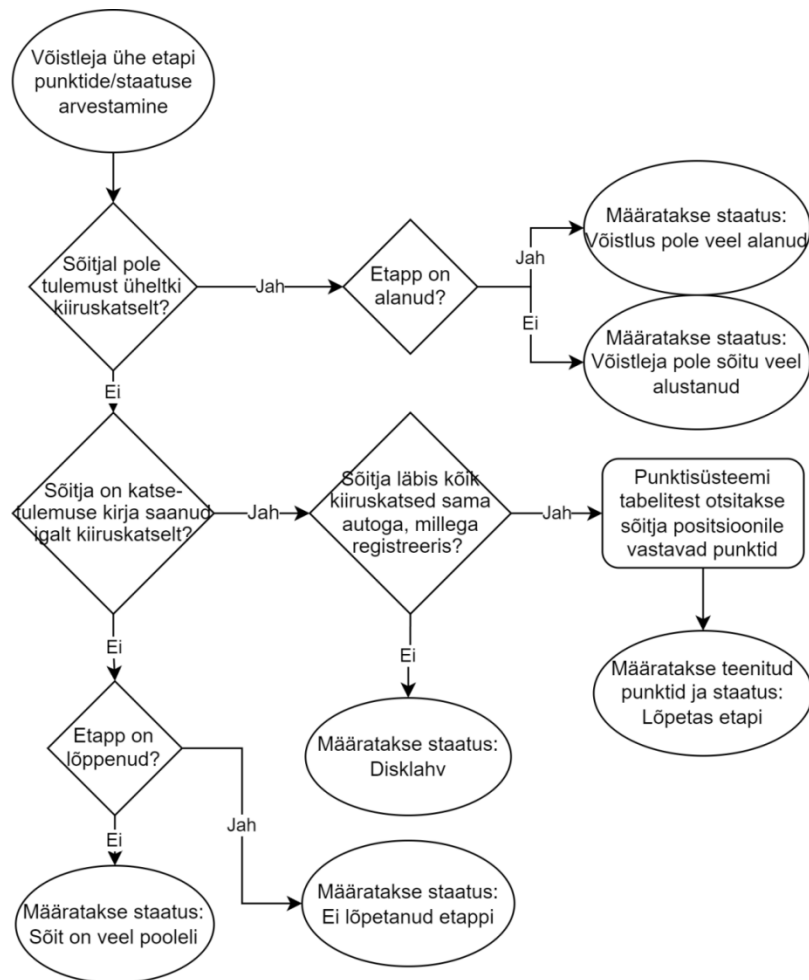
Teatud võistlusplatvormide puhul on kiiruskatsete vahel võistlusauto muutmine võimalik ja seda võimalust võib mõni sõitja proovida ära kasutada eelise saamiseks. Selle vältimiseks kontrollib rakendus punktide arvutamisel (peatükk 3.1.3) seda, kas võistleja läbis kõik kiiruskatsed sama autoga, mille registreerimisel märkis.

3.1.3 Tulemuste põhjal punktide arvutamine

Hetkel on võimalik korraldajal uuendada võistluse punktitabletit nupuvajutusega. Tulevikus võiks mõelda, kas mõistlik oleks punktitabletit uuendada peale igat katsetulemuse sisestamist või näiteks perioodiliselt, kasutades selleks ajastatud töid (*scheduled jobs*). Tihedam uuendamine võib olla kasulik, et kajastada alati kõige värskemad pingerida.

Võistlustele registreerunud sõitjate etapitulemuste kohta hoitakse infot Standings tabelis. Iga etapi kohta on võistlejal üks Standings tabeli kirje, mis koosneb staatusest ja teenitud punktidest. Võistleja etapistaatuse järgi on võimalik rakenduse kasutajatele kuvada näiteks seda, kas sõitja on kiiruskatsete läbimist alustanud, disklahvi saanud või etapi edukalt lõpetanud.

Allpool on toodud punktide ja staatuste määramise voodiagramm (Joonis 2).



Joonis 2. Punktide arvestamise voodiagramm.

3.2 Andmebaasi arendus

Peatükis antakse ülevaade andmemudelist ja kirjeldatakse, kuidas andmebaasile muudatuste tegemine ja sealt andmete pärimine käib. Olemi-suhte diagramm (ERD, *Entity Relationship Diagram*) on toodud välja peatüki lõpus (Joonis 3).

3.2.1 Andmebaasi loomine ja andmete manipuleerimine

Andmebaasi tabelite esmakordne loomine ja edaspidine struktuuri uuendamine käib EF Core (*Entity Framework Core*) raamistikuga, kasutades migratsioone (*migrations*). Migratsioonide kasutamine on mugav – nende abil on võimalik andmestruktuuris tehtud muudatusi näha versioonihaldussüsteemis, seekaudu tuvastada võimalikke vigu ja vajadusel muudatusi tagasi keerata.

EF Core on objekt-relatsioonilise kaardistuse (ORM, *object-relational mapping*) raamistik. ORM lihtsustab koodis andmebaasipäringute tegemist. EF Core võimaldab andmebaasi tabelite loomiseks kasutada kood-enne lähenemist (*code-first approach*), mis tähendab, et esmalt luuakse koodis domeenobjektid, seejärel genereeritakse nende põhjal migratsioon ja installeeritakse see migratsioon andmebaasi. Kood-enne lähenemist otsustas autor kasutada, et arendusprotsessi kiirendada.

3.2.2 Andmemudel

Rakenduse kasutajate autentimiseks ja autoriseerimiseks kasutatakse ASP.NET Core Identity tabelleid (lähemalt punktis 3.3). Sõitjate pärisnimede hoidmiseks laiendati kasutajate olemit `AspNetUsers` ees- ja perekonnanime atribuutidega.

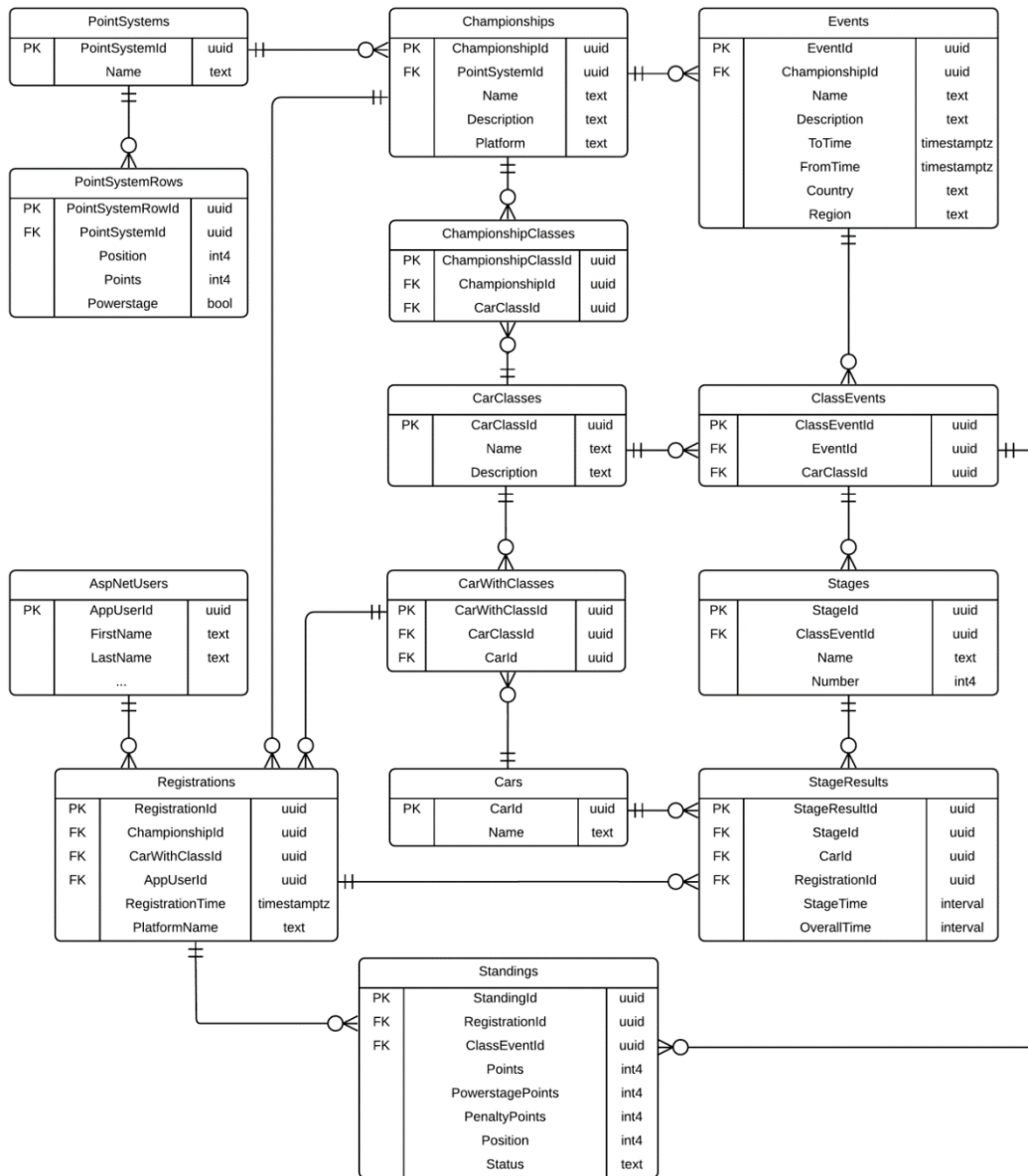
Võistluste korraldamiseks erinevates võistlusklassides peavad olema ära defineeritud autod ja autoklassid. Autode tabelisse `Cars` on võimalik korraldajal lisada autosid ja `CarClasses` tabelisse autoklasse. Autoklasside ja autode vahele loodi mitu-mitmele seos `CarWithClasses` tabeli näol, et pakkuda korraldajale paindlikku autoklasside haldamise võimalust.

Punktide jagamise süsteemi jaoks on tabelid `PointSystems` ja `PointSystemRows`. Punktisüsteemi iga rida tähistab mitu punkti annab teatud saavutatud koht. Lisaks on võimalus ära märkida, kas tegemist oli punktikatsega.

Championships tabelis hoitakse infot võistluste kohta. Võistluse punktisüsteem ja võistlusklassid (ChampionshipClasses) on seotud Championship olemiga – võistlus on seotud ühe punktisüsteemiga ja võib hõlmata mitut võistlusklassi.

Võistlused koosnevad rallietappidest – nende jaoks on loodud tabel Events. ClassEvents tabeliga seostatakse etapid võistlusklassidega. Etapid jagunevad kiiruskatseteks, mis on kirjeldatud Stages tabelis. Kiiruskatsete tulemused sisestatakse StageResults tabelisse.

Sõitjate võistlustele registreerimised koos võistlusplatvormil kasutatavate kasutajanimedega salvestatakse Registrations tabelisse. Etappide eest teenitud punktid ja sõitja staatus etapil pannakse kirja tabelisse Standings.

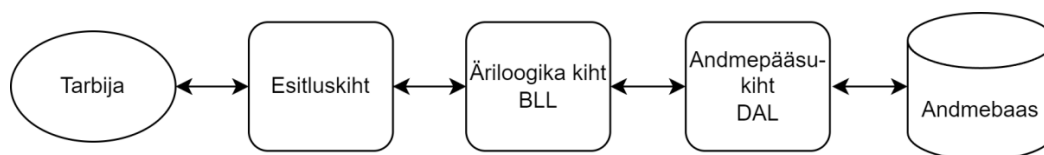


Joonis 3. Andmebaasi olemi-suhte diagramm.

3.3 Serveripoolse rakenduse arendus

Rakendusliidese ehk API arendamiseks on kasutusel ASP.NET Core 6 veebiraamistik. Projekti loomisel valitakse rakenduse kasutajate haldamiseks ASP.NET Core Identity (edaspidi Identity) teek. Identity loob kasutajaandmete hoiustamiseks andmebaasitabelid. Teegi abil on võimalik rakenduse kasutajaid autentida ja autoriseerida. Kuna soov on lisaks sõitjate kasutajanimedele kuvada ka nende pärisnimesid, laiendatakse Identity kasutajate klassi IdentityUser ees- ja perekonnanime atribuutidega. Rakenduse administraatorile luuakse kohandatud kasutajaroll. Sellise rolliga kasutajad saavad uusi võistluseid luua ja neid hallata – selleks märgistatakse vajalikud otspunktid [Authorize] atribuudiga, andes kaasa rollid, kes võivad otspunktile ligi pääseda.

Rakendus luuakse kihelist arhitektuurimustrit (Joonis 4) järgides. Eraldatud on andmepääsu- (DAL, *Data Access Layer*), äriloogika- (BLL, *Business Logic Layer*) ja andmeesitluskihid. Andmepääsukihi toimub suhtlus andmebaasiga. Äriloogika kihis tehakse andmetega teisendusi ja/või arvutusi vastavalt ärireeglitele. Esitluskiht tegeleb sissetulnud päringute käsitlemisega, andmete edastamisega äriloogika kihti ja tarbijale äriloogika kihist saadud vastuse edastamisega. Rakenduse kihilisus täidab huvide eraldatuse (*separation of concerns*) printsiipi – iga programmiosa vastutab kindla funktsionaalsuse eest, mis muudab rakenduse haldamise lihtsamaks, kuna kihid ei mõjuta üksteist.



Joonis 4. Kihiline arhitektuur.

Iga kihi juurde kuuluvad vastavad andmevahetusobjektid (DTO, *data transfer object*), et tagada kihtidele ligipääs vaid neile vajalikule teabele. Näiteks võib andmepääsukihi DTO sisaldada kõiki olemi kohta käivaid väljasid, aga esitluskihist tarbijale väljasaadetavas DTOs tohib olla vaid alamhulk neist atribuutidest. Kihtide vaheliseks DTOde teisendamiseks kasutatakse vastendeid (*mappers*), mis vajalikke väärtuseid ühelt objektilt teisele kopeerivad.

Esialgsed CRUD (*create, read, update, delete*) teenuste otspunktid tekitatakse kasutades ASP.NET Core *scaffolding* mootorit. *Scaffoldingu* abil genereeritakse peamised andmete manipuleerimiseks vajalikud kontrollid, mida arendaja saab oma soovide järgi kohandada. Antud rakenduse puhul eemaldatakse ebavajalikud otspunktid ja genereeritud otsepöördumised andmebaasi vahetati teenuskihis loodud teenuste vastu.

Kasutajate autentimine ja autoriseerimine klientrakenduse ja rakendusliidese vahel käib JWTd (*JSON Web Token*) kasutades. Antud rakenduse puhul hoitakse JWTs näiteks kasutajarollide tõendit (*claim*). Eduka sisselogimise ja registreerimise puhul genereerib ja signeerib tagarakendus JWT ning edastab selle eesrakendusele. Klientrakendus salvestab tokeni ja saadab selle edasiste päringute puhul kaasa HTTP Authorization päises. Server kontrollib sissetulnud päringu JWT olemasolul selle valiidsust, kasutades selleks sama sümmeetrilist võtit, millega token algul signeeritud sai.

Andmebaasist pärimine ja sinna salvestamine toimub EF Core abil. Koodikorduse vähendamiseks koondatakse enamkasutatavad CRUD andmebaasi päringud üldisesse andmehoidlasse (*repository*). Süsteem ehitatakse üles paindlikult – andmehoidlates on võimalik kasutada üldise andmehoidla meetodeid või lisada juurde olemi spetsiifilisi päringuid.

Järgitakse tööühiku (*UOW, unit of work*) mustrit, et tagada andmete usaldusväärsus. Kui korruga on vaja teostada mitu andmebaasitehingut, käsitletakse neid ühe transaktsioonina, mis tähendab, et andmeid ei salvestata poolikult vaid järgitakse *kõik või mitte midagi* põhimõtet – kui üks tehing ebaõnnestub, keeratakse kõik muudatused tagasi.

Loodava rakendusliidese dokumenteerimiseks ja testimiseks seatakse üles Swagger UI keskkond. API kohta genereeritakse OpenAPI spetsifikatsioonile vastav masinloetav liides, mille järgi Swagger UI kuvab otspunktide infot ja pakub kasutajasõbralikku otsesuhtlust rakendusliideselega.

3.4 Kliendipoolse rakenduse arendus

Eesrakenduse loomiseks kasutatakse React.js kasutajaliidese teeki koos TypeScriptiga. Arendus toimub tagarakenduse arendamisega paralleelselt – kui mingi teenus valmib tagarakenduses, võetakse see klientrakenduses kohe ka kasutusele. Paketihaldus käib Node Package Manageri (NPM) abil.

3.4.1 Eesrakenduse loomine

Rakenduse loomiseks otsustatakse kasutada Create React App (CRA) tööriista. Selleks käivitatakse `npx` (Node Package eXecute) käsk (Joonis 5), mis loob uue React.js rakenduse põhja. Rakenduse põhi sisaldab endas eelkonfigureeritud arenduskeskkonda ja toodangversiooni ehitamise jaoks vajalikke `babel` ja `webpack` seadistusi. CRAd kasutades on võimalik oluliselt vähendada seadistamisele kuluvat aega ja keskenduda uute funktsionaalsuste arendamisele.

```
npx create-react-app rally-hub-client --template typescript
```

Joonis 5. `npx` käsk React.js rakenduse loomiseks.

Eesrakenduse arendamiseks kasutatakse Bootstrap tööriista. Kasutajaliides koostatakse Bootstrapi komponentide abil, mis tagab ühtlase disaini üle kogu rakenduse. Bootstrapi abil on võimalik rakendust lihtsalt kohandada erinevate ekraanisuuruste ja seadmete jaoks.

Kasutaja autentimise ja kasutajaõiguste infot läheb kasutajaliidese kuvamisel vaja erinevates kohtades, seega otsustati kasutada React Context APIt, et kasutajainfot globaalselt hoiustada. Eduka sisselogimise või registreerimise puhul tagarakendusest saadud JWT salvestatakse loodud AuthContexti. JWT sisu sõeludes (*parse*) saab teada, mis rollidesse kasutaja kuulub ja ühtlasi saab selle põhjal otsustada, kas kasutaja peaks rakenduse mingile osale ligi pääsema või mitte.

Serveriga suhtluseks võetakse kasutusele Axios HTTP klient. Koodikorduse vähendamiseks luuakse üldine teenusklass, mis sisaldab meetodeid põhiliste HTTP päringute – GET, PUT, POST, DELETE – teostamiseks. Üldise teenusklassi meetodeid on mujal rakenduses mugav kasutada – tuleb ette anda vaid API teenuse aadress, sisend- ning tagastustüübid ja vajadusel päringu keha (*payload*).

3.4.2 Eesrakenduse ja kasutajaliidese tutvustus

Rakendust on võimalik kasutada sisselogimata vaid võistluste info lugemiseks. Võistlustele registreerimiseks peab kasutaja olema sisse logitud. Sisselogimisel kuvatakse kasutajale tervitusleht koos võistluste nimekirja vaatesse (Joonis 6) viiva nupuga. Rakendus on hetkel ainult ingliskeelne, aga korraldaja saab sisestada andmeid (võistluste nimed, kirjeldused jms) omale sobivas keeles.

Championships

Name	Description	Platform	
Virtuaalne rahvaralli		Dirt Rally 2.0	Details
World Virtual Rally Championship		WRC Generations	Details
E-autoralli Eesti meistrivõistlused	Toimuvad alates 2021	Richard Burns Rally	Details

Joonis 6. Võistluste nimekirja vaade.

Võistluste nimekirja vaade on suhteliselt minimalistlik, välja kuvatakse võistluste nimed, korraldaja poolt neile määratud kirjeldused (kui on määratud) ja platvormi nimed, kus sõitmine toimub. *Details* nupule vajutades suunatakse kasutaja võistlusetappide vaatesse (Joonis 7), kus on informatsioon kõikide rallietappide kohta, mis antud võistluse alla kuuluvad. Teavet näeb etappide virtuaalse võistluspaiga ning alguse- ja lõpuaja kohta. Lehe ülemises pooles on navigeerimisriba, mille abil on võimalik liikuda teistesse antud võistluse kohta käivatesse – registreerunute nimekirja, võistlusklasside, punktisüsteemi ja punktitableti – vaadettesse.

E-autoralli Eesti meistrivõistlused

Name	Description	Country	Region	Starts	Ends	
Kreeka		Kreeka	Argolis	6.2.2023, 23:46:00	27.2.2023, 23:46:00	Details
Eesti		Eesti	Lõuna-Eesti	11.4.2023, 00:50:00	2.5.2023, 00:50:00	Details
Soome		Soome	Jyväskylä	25.4.2023, 00:50:00	25.4.2023, 00:50:00	Details

[Back](#)

Joonis 7. Võistlusetappide nimekirja vaade.

Siin vaates on korraldajal võimalik lisada uusi võistlusetappe (Joonis 8). Tavakasutajale seda võimalust ei kuvata. Rallietapi lisamiseks tuleb märkida etapi nimi, toimumiskoht ja ajavahemik, mille jooksul sõitjad raja läbida võivad.

E-autoralli Eesti meistrivõistlused

Events Entries Categories Point system Standings

Add an event to the championship ✕

Event name

Country Region

Description

Start time

End time

Joonis 8. Võistlusetaipi lisamise plokk (olles korraldaja rollis).

Võistlusetaipide nimekirjas *Details* nupule vajutades avaneb detailvaade antud etapi tulemustest (Joonis 9). Vaate ülemises servas on kaks sorteerimisriba: üks võistlusklasside, teine kiiruskatse jaoks. All kuvatakse vastavalt kasutaja tehtud sorteerimisvalikutele tulemusi – kiiruskatse tulemusi vasakpoolses tabelis ja üldseisu peale valitud kiiruskatset parempoolses tabelis. Näidatakse ka seda, kas sõitja sõitis katse sama autoga, millega registreeris.

Events

Rally2 Rally1

SS1 Corfu SS2 Athens SS3 Pelion

SS3 Pelion

STAGE					OVERALL						
#	Name	Registered car	Car	OK	Time	#	Name	Registered car	Car	OK	Time
1	Marko Maasikas	Ford Puma	Ford Puma	✓	00:05:39	1	Marko Maasikas	Ford Puma	Ford Puma	✓	00:34:57
2	Juhan Juurikas	Toyota Yaris	Toyota Yaris	✓	00:05:50	2	Juhan Juurikas	Toyota Yaris	Toyota Yaris	✓	00:36:06

Joonis 9. Etapitulemuste detailvaade.

Korraldaja saab siin lisada uusi katseid ja katsete tulemusi (Joonis 10). Katsetulemuse lisamiseks peab sisestama auto nime, millega sõitja raja läbis, sõitja kasutajanime ja katse läbimiseks kulunud aja.

Events

Rally2 Rally1

SS1 Corfu SS2 Athens SS3 Pelion

SS1 Corfu [Delete stage](#)

Add new stage result

Joonis 10. Katsetulemuse lisamise plokk (olles korraldaja rollis).

Tagasi võistluse üldvaatesse liikudes ja *Entries* vahelehele navigeerides kuvatakse võistlusele registreerunud sõitjate nimekirja (Joonis 11). Sisselogitud kasutajatele, kes ei ole juba võistlusele registreerunud, kuvatakse võimalus end võistlusele kirja panna – selleks tuleb sisestada võistlusplatvormil kasutatav kasutajanimi ning valida omale sobiv võistlusklass ning auto.

E-autoralli Eesti meistrivõistlused

Events **Entries** Categories Point system Standings

First name	Last name	Platform name	Car class	Car	Registration time
Juhan	Juurikas	JuhanJ	Rally1	Toyota Yaris	25.4.2023, 00:58:17
Marko	Maasikas	Marko99	Rally1	Ford Puma	25.4.2023, 00:59:07

Joonis 11. Registreerunute nimekirja vaade.

Categories vahelehel kuvatakse nimekiri võistlusel kasutatavatest võistlusklassidest ja nendesse klassidesse kuuluvatest autodest. Korraldaja rollis olles on kasutajal võimalus võistlusklasse hallata (Joonis 12) – saab muuta klassidesse kuuluvaid autosid, lisada uusi võistlusklasse või eemaldada olemasolevaid.

E-autoralli Eesti meistrivõistlused

Events Entries **Categories** Point system Standings

Add new

AVAILABLE CAR CLASSES

Rally1	Ford Puma Toyota Yaris	Remove
Rally2	Hyundai i20 Ford Fiesta	Remove

Back

- Toyota Yaris
- Ford Puma

Joonis 12. Võistlusetapi võistlusklasside haldamise vaade (olles korraldaja rollis).

Point system vahelehel (Joonis 13) kuvatakse kasutajale teavet selle kohta, kuidas punktiarvestus käib – mitu punkti on võimalik iga etapil saavutatud koha eest teenida. Korraldaja saab seadistada punktide jaotust vastavalt oma soovidele. Muuta saab nii etapi eest teenitud põhipunktide kui ka punktikatse eest jagatavate punktide arvu.

E-autoralli Eesti meistrivõistlused

Events Entries Categories **Point system** Standings

Points distribution		Power stage	
Position	Points	Position	Points
1	25	1	5
2	18	2	4
3	15	3	3
4	12	4	2
5	10	5	1
6	8		
7	6		
8	4		
9	2		
10	1		

Joonis 13. Võistluse punktisüsteemi vaade.

Standings vahelehel (Joonis 14) kuvatakse võistluse punkttabelit – võistlejate nimesid ja nende teenitud punktide arvu või staatust etappidel (ei lõpetanud, ei osalenud, disklahv). Tabeleid kuvatakse võistlusklasside kaupa. Kuvatava klassi vahetamiseks on tabeli ülaservas võistlusklasside nimelised nupud.

E-autoralli Eesti meistrivõistlused

Events	Entries	Categories	Point system	Standings
Rally2	Rally1			
Driver	Kreeka	Soome	Eesti	Total
Marko Maasikas	25+4	18+5		52
Juhan Juurikas	18+5	DNF		23

Back

Joonis 14. Punkttabeli vaade.

4 Tulemus

Enne rakenduse arendamist teostati analüüs, mille käigus selgitati välja kõige prioriteetsemad funktsionaalsused, et rakendus oleks eesmärgiliselt kasutatav. Rakenduse nõuetele vastavust testis autor arenduse käigus jooksvalt. Lõpplahenduse saab lugeda õnnestunuks, sest analüüsi põhjal seatud kõige kõrgema prioriteedi saanud nõuded on täidetud. Madalama prioriteediga nõudeid rakendus hetkel ei rahulda.

Rakendus täidab kõik põhinõuded, mis tähendab, et selle abil on võimalik võistluseid korraldada. Sõitjad saavad logida sisse, registreerida võistlustele omale sobivas võistlusklassis, näha võidusõidu tulemusi ja punktiarvestuse tabeleid. Korraldaja saab luua uusi võistluseid, kohandada võistlusklasse ja punktisüsteeme vastavalt oma soovile.

4.1 Edasiarenduse võimalused

Rakenduse kasutuskogemuse parandamiseks tuleks arendada juurde funktsionaalsuseid, mis täidaksid ära madalama prioriteedi saanud nõuded. Eraldi saab välja tuua võistluste korralduslikust poolest andmete sisestamise – katsetulemuste ükshaaval sisestamine ei ole mõistlik ega jätkusuutlik variant.

5 Kokkuvõte

Lõputöö eesmärgiks oli luua veebiportaal, mis koondaks erinevatel virtuaalautoralli platvormidel toimuvad võistlussarjad ühte kohta kokku ja abistaks võidusõiduürituste korraldajal võistluseid läbi viia.

Tehnoloogiate analüüsi tulemusena valiti veebiportaali tagarakenduse arendamiseks C# ja ASP.NET Core. Rakendus kasutab andmebaasina PostgreSQL. Eesrakenduse loomiseks võeti kasutusele React.js teek koos TypeScript keelega.

Käesoleva töö tulemusena valmis rakendus, mille toel on võimalik virtuaalautoralli võistluseid läbi viia väiksema vaevaga, olenemata võistlusplatvormist. Rakendusse on kokku koondatud võistlustele registreerimine, võistlusklasside haldamine, punktisüsteemide seadistamine ja automaatne punktijaotuse arvutamine peale võistlustulemuste sisestamist.

Valminud lahendus sisaldab kõiki funktsionaalsuseid, millele määrati analüüsi käigus kõige kõrgem prioriteet. Veebiportaali on võimalik kasutajakogemuse parandamiseks edasi arendada – selleks võib järgida analüüsis väljatoodud madalama prioriteediga nõudeid.

Kasutatud kirjandus

- [1] C. Groom, „Controller vs Steering wheel: Which is better for sim racing?“, [Võrgumaterjal]. Available: <https://racinggames.gg/hardware/controller-vs-steering-wheel-which-is-better-for-sim-racing>. [Kasutatud 15 4 2023].
- [2] R. Baxter, „Force Feedback: what is FFB, how does it work and how do I set it up?“, [Võrgumaterjal]. Available: <https://simracingcockpit.com/setup-force-feedback>. [Kasutatud 15 4 2023].
- [3] J. M. Kovács ja Á. Szabó, „Esport and simracing markets – The effects of COVID-19, difficulties and opportunities“, Institute of Business Economics, Corvinus University Budapest, Budapest, Hungary, [Võrgumaterjal]. Available: <https://doi.org/10.1556/204.2022.00012>. [Kasutatud 15 4 2023].
- [4] K. Sooper, „Algavad Eesti meistrivõistlused e-autospordis“, [Võrgumaterjal]. Available: <https://uus.autosport.ee/e-autosport/algavad-eesti-meistrivoistlused-e-autospordis>. [Kasutatud 15 4 2023].
- [5] Eesti Virtuaalse Autospordi Liit, „EMV e-autoralli punktitable 2021“, [Võrgumaterjal]. Available: <https://docs.google.com/spreadsheets/d/1lmleEvKMXIga0GcHeKtip2Eh26fwwxm8f0d1GmKE7Pc>. [Kasutatud 15 4 2023].
- [6] The Virtual Rally Championship, „What is VRC?“, [Võrgumaterjal]. Available: <https://www.virtualrallychampionship.com/what-is-vrc>. [Kasutatud 15 4 2023].
- [7] Eesti Virtuaalse Autospordi Liit, „E-autoralli Eesti meistrivõistluste 2023 üldjuhend“, [Võrgumaterjal]. Available: https://www.simracing.ee/media/rules/23RC/E_AUTORALLI_EESTI_MEISTRIVOISTLUSTE_2023_YLDJUHEND_kinnitatud-10.01.2023.pdf. [Kasutatud 15 4 2023].
- [8] World Rally Championship, „What is the WRC?“, [Võrgumaterjal]. Available: <https://www.wrc.com/en/more/about-wrc/what-is-wrc>. [Kasutatud 15 4 2023].
- [9] Federation Internationale de l'Automobile, „2023 FIA World Rally Championship Sporting Regulations“, [Võrgumaterjal]. Available: https://www.fia.com/sites/default/files/wrcsr_2023_2022_12_15_green_highlights_0.pdf. [Kasutatud 15 4 2023].
- [10] WRC Promoter GmbH, „Season #6 Rulebook“, [Võrgumaterjal]. Available: <https://www.wrc.com/en/more/gaming/rules>. [Kasutatud 15 4 2023].
- [11] M. Cohn, „What is a user story?“, [Võrgumaterjal]. Available: <https://www.mountangoatsoftware.com/agile/user-stories>. [Kasutatud 15 4 2023].
- [12] M. Vestola, „A Comparison of Nine Basic Techniques for Requirements Prioritization“, Helsinki University of Technology, [Võrgumaterjal]. Available: http://www.mvnet.fi/publications/software_development_seminar.pdf. [Kasutatud 2023 4 15].
- [13] Amoeboids, „MoSCoW Technique- What it is & Insights on MoSCoW

- Prioritization,“ [Võrgumaterjal]. Available: <https://amoeboids.com/blog/moscow-building-better-products>. [Kasutatud 15 4 2023].
- [14] International Institute of Business Analysis, *A Guide to the Business Analysis Body of Knowledge (BABOK Guide)*, 2009, pp. 6, 102.
- [15] M. Martin, „What is Non-Functional Requirement in Software Engineering?“, [Võrgumaterjal]. Available: <https://www.guru99.com/non-functional-requirement-type-example.html>. [Kasutatud 15 4 2023].
- [16] S. Smith, „Choose Between Traditional Web Apps and Single Page Apps (SPAs)“, [Võrgumaterjal]. Available: <https://learn.microsoft.com/en-us/dotnet/architecture/modern-web-apps-azure/choose-between-traditional-web-and-single-page-apps>. [Kasutatud 16 4 2023].
- [17] G. Petrov, „Enhance the User Experience with Single Page Apps“, [Võrgumaterjal]. Available: <https://medium.com/wappler-io/enhance-the-user-experience-with-single-page-apps-43a8228c4646>. [Kasutatud 15 4 2023].
- [18] Stack Overflow, „Stack Overflow Developer Survey 2022“, [Võrgumaterjal]. Available: <https://survey.stackoverflow.co/2022/#most-popular-technologies-language>. [Kasutatud 16 4 2023].
- [19] N. Swanner, „5 Backend JavaScript Frameworks Experts Love“, [Võrgumaterjal]. Available: <https://www.dice.com/career-advice/5-backend-javascript-frameworks-experts-love>. [Kasutatud 16 4 2023].
- [20] M. Miguel, „JavaScript is not an untyped language“, [Võrgumaterjal]. Available: <https://dev.to/miguelmj/javascript-is-not-an-untyped-language-1jkg>. [Kasutatud 16 4 2023].
- [21] N. Đuza, „Ride Down Into JavaScript Dependency Hell“, [Võrgumaterjal]. Available: <https://blog.appsignal.com/2020/04/09/ride-down-the-javascript-dependency-hell.html>. [Kasutatud 16 4 2023].
- [22] TypeScript, „TypeScript: JavaScript with syntax for types.“, [Võrgumaterjal]. Available: <https://www.typescriptlang.org>. [Kasutatud 16 4 2023].
- [23] TypeScript, „TypeScript: Why does TypeScript exist?“, [Võrgumaterjal]. Available: <https://www.typescriptlang.org/why-create-typescript>. [Kasutatud 16 4 2023].
- [24] Visual Studio Code, „Compiling TypeScript“, [Võrgumaterjal]. Available: <https://code.visualstudio.com/docs/typescript/typescript-compiling>. [Kasutatud 16 4 2023].
- [25] R. Sharma, „Top 12 Fascinating Python Applications in Real-World“, [Võrgumaterjal]. Available: <https://www.upgrad.com/blog/python-applications-in-real-world>. [Kasutatud 16 4 2023].
- [26] Thinkful, „How Hard Is It to Learn Python?“, [Võrgumaterjal]. Available: <https://www.thinkful.com/blog/how-hard-is-it-to-learn-python>. [Kasutatud 16 4 2023].
- [27] JavaTpoint, „Advantages and disadvantages of Java“, [Võrgumaterjal]. Available: <https://www.javatpoint.com/advantages-and-disadvantages-of-java>. [Kasutatud 16 4 2023].
- [28] GeeksForGeeks, „Why is Java ‘write once and run anywhere’?“, [Võrgumaterjal]. Available: <https://www.geeksforgeeks.org/why-is-java-write-once-and-run-anywhere>. [Kasutatud 16 4 2023].
- [29] P. Trifonov, „Why JVM Is Great“, [Võrgumaterjal]. Available:

- <https://medium.com/@PavelTrifonov/why-jvm-is-great-4b3d8b224eae>. [Kasutatud 16 4 2023].
- [30] Microsoft, „A tour of the C# language,“ [Võrgumaterjal]. Available: <https://learn.microsoft.com/en-us/dotnet/csharp/tour-of-csharp>. [Kasutatud 16 4 2023].
- [31] Microsoft, „What is .NET?,“ [Võrgumaterjal]. Available: <https://dotnet.microsoft.com/en-us/learn/dotnet/what-is-dotnet>. [Kasutatud 16 4 2023].
- [32] AltexSoft, „The Good and the Bad of C# Programming,“ [Võrgumaterjal]. Available: <https://www.altexsoft.com/blog/c-sharp-pros-and-cons>. [Kasutatud 16 4 2023].
- [33] S. Person, „TypeScript: What's the Point?,“ [Võrgumaterjal]. Available: <https://dev.to/shaundai/typescript-what-s-the-point-5116>. [Kasutatud 17 4 2023].
- [34] React, „React,“ [Võrgumaterjal]. Available: <https://react.dev>. [Kasutatud 17 4 2023].
- [35] Codecademy, „React: The Virtual DOM,“ [Võrgumaterjal]. Available: <https://www.codecademy.com/article/react-virtual-dom>. [Kasutatud 17 4 2023].
- [36] KnowledgeHut, „What are the Pros and Cons of React,“ [Võrgumaterjal]. Available: <https://www.knowledgehut.com/blog/web-development/pros-and-cons-of-react>. [Kasutatud 17 4 2023].
- [37] Angular.io, „What is Angular?,“ [Võrgumaterjal]. Available: <https://angular.io/guide/what-is-angular>. [Kasutatud 17 4 2023].
- [38] S. Yadav, „Cons of AngularJS & Angular 2+,“ [Võrgumaterjal]. Available: <https://www.linkedin.com/pulse/cons-angularjs-angular-2-siyaram-ray>. [Kasutatud 17 4 2023].
- [39] Vue.js, „What is Vue?,“ [Võrgumaterjal]. Available: <https://vuejs.org/guide/introduction.html>. [Kasutatud 17 4 2023].
- [40] S. Krause, „JS web frameworks benchmark,“ [Võrgumaterjal]. Available: https://krausest.github.io/js-framework-benchmark/2023/table_chrome_112.0.5615.49.html. [Kasutatud 17 4 2023].
- [41] AltexSoft, „The Good and the Bad of Vue.js Framework Programming,“ [Võrgumaterjal]. Available: <https://www.altexsoft.com/blog/engineering/pros-and-cons-of-vue-js>. [Kasutatud 17 4 2023].
- [42] R. Vettor ja S. Smith, „Relational vs. NoSQL data,“ [Võrgumaterjal]. Available: <https://learn.microsoft.com/en-us/dotnet/architecture/cloud-native/relational-vs-nosql-data>. [Kasutatud 24 4 2023].
- [43] IBM, „ACID properties of transactions,“ [Võrgumaterjal]. Available: <https://www.ibm.com/docs/en/cics-ts/5.4?topic=processing-acid-properties-transactions>. [Kasutatud 24 4 2023].
- [44] AltexSoft, „Comparing Database Management Systems: MySQL, PostgreSQL, MSSQL Server, MongoDB, Elasticsearch, and others,“ [Võrgumaterjal]. Available: <https://www.altexsoft.com/blog/business/comparing-database-management-systems-mysql-postgresql-mssql-server-mongodb-elasticsearch-and-others>. [Kasutatud 25 4 2023].
- [45] Oracle, „Oracle MySQL,“ [Võrgumaterjal]. Available: <https://shop.oracle.com/apex/product?p1=MySQL>. [Kasutatud 25 4 2023].

- [46] Codecademy, „What is SQLite?“, [Võrgumaterjal]. Available: <https://www.codecademy.com/article/what-is-sqlite>. [Kasutatud 25 4 2023].
- [47] SQLite, „Appropriate Uses For SQLite“, [Võrgumaterjal]. Available: <https://sqlite.org/whentouse.html>. [Kasutatud 25 4 2023].
- [48] Microsoft, „SQL Server 2022 pricing and licensing“, [Võrgumaterjal]. Available: <https://www.microsoft.com/en-us/sql-server/sql-server-2022-pricing>. [Kasutatud 25 4 2023].
- [49] JetBrains s.r.o, „React“, [Võrgumaterjal]. Available: <https://www.jetbrains.com/help/idea/react.html>. [Kasutatud 17 4 2023].
- [50] V. Moodley, „What’s going on with cheating in sim racing?“, [Võrgumaterjal]. Available: <https://simracinginfo.com/whats-going-on-with-cheating-in-sim-racing>. [Kasutatud 28 4 2023].

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks¹

Mina, Evert Padari

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Virtuaalautoralli võistluste portaali arendus“, mille juhendaja on Tiina Zingel
 - 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.

12.05.2023

¹ 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 punktidele 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.