

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Helena Väinmaa 185078IADB

Raamistik interaktiivse teatri rakenduste arendamiseks

Bakalaureusetöö

Juhendajad: Kristiina Hakk
PhD

Mihkel Seeder
MA

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: Helena Väinmaa

03.01.2023

Annotatsioon

Käesoleva bakalaureusetöö eesmärk on luua interaktiivsete teatriprojektide tarbeks mobiili- ja veebirakenduste raamistik, mis hõlbustaks nende projektide arendust ning uuendaks teatritehnoloogiat. Kavandatava raamistiku eesmärk on pakkuda baasprojektid ning rakenduse käivitamise süsteemid, mis elimineeriks korduva töö läbi rakenduste ühisosa ühtlustamise.

Töö käigus analüüsitakse raamistiku nõudeid ning arendatakse raamistik. Rõhku pannakse ühest küljest kasutatavusele teatris ja teisest küljest järgmise arendaja mugavusele, kelle jaoks peavad ülesseadmisprotsessid olema kiired ja lihtsad.

Lõputöö tulemusena valmib baasprojektide kogum, mis koos seadistuse ja juhenditega moodustab järgnevatele arendustele baasiks oleva raamistiku. Baasprojektide hulka kuuluvad esi- ja tagarakendus, andmebaas ja arvutivõrkude ning keskkondade seadistus.

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

Abstract

Framework to Develop Applications for Interactive Theatre

The aim of this bachelor's thesis is to create a framework for mobile and web applications for interactive theater projects, which would aid in the development of these projects and innovate theater technology. The goal of the proposed framework is to provide base projects and application launch systems which would eliminate repetitive work by harmonizing the common parts of applications.

In completion of the thesis, the requirements of the framework are analyzed and the framework is developed. Emphasis is placed on usability inside the theater on the one hand, and on the other hand for the next developer, for whom setup processes need to be quick and simple.

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

Lühendite ja mõistete sõnastik

API	<i>Application Programming Interface</i> , rakendusliides
CSS	<i>Cascading Style Sheets</i> - veebilehtede küljendamise märgistuskeel
EOL	<i>End of Life</i> - eluaja lõpp, tarkvara toetamise tähtaeg
HTML	<i>HyperText Markup Language</i> - veebilehe märgendamise keel
HTTP	<i>Hypertext Transfer Protocol</i> - hüperteksti edastusprotokoll
HTTPS	<i>Hypertext Transfer Protocol Secure</i> - turvaline hüperteksti edastusprotokoll
ID	<i>Identifier</i> , identifitseeriv number või kood
JWT	<i>JSON Web Token</i> , internetistandard andmete digitaalallkirjastamiseks ja/või krüpteerimiseks
liitreaalsus	reaalse ja virtuaalse keskkonna kombineerimine
LTS	<i>Long-term Support</i> - pikaajaline tugi, versioon tootest või tarkvarast, mida toetatakse kauem
MEVN <i>microcopy</i>	MongoDB, ExpressJS, VueJS ja NodeJS pinu lühend tarkvaras leiduvad lühikesed tekstiosad
MoSCoW	<i>Must have, Should have, Could have, Won't have</i> - akronüüm prioritseerimismeetodile
NFC	<i>Near-field communication</i> - lähiväljaside
npm	Algselt <i>Node Package Manager</i> , ümber nimetatud <i>npm is not an acronym</i> , stiliseeritud väiketähtedega
QR (code)	<i>Quick Response (code)</i> - ruutkood
REST	<i>Representational State Transfer</i> - veebiteenusega suhtlemise liidese arhitektuur
RFID	<i>Radio-frequency identification</i> - raadiosagedustuvastus

SCSS

Syntactically Awesome Style Sheets - märgistuskeel,
mis kompileeritakse CSSiks

WLAN

Wireless Local Area Network - traadita side kohtvõrk

Sisukord

1 Sissejuhatus	11
2 Probleemi kirjeldus.....	12
2.1 Lõputöö eesmärgid ja skoop.....	12
2.2 Metoodika.....	13
2.3 Raamistiku kavandamine.....	13
2.3.1 Raamistiku kasutamise visioon	14
2.3.2 Probleemid, mida raamistik lahendab	14
2.3.3 Raamistiku eesmärgid	15
2.3.4 Raamistiku komponendid.....	17
2.4 Tehnoloogia kasutamise võimalused interaktiivses teatris.....	17
2.4.1 Veebirakendused publiku mobiilseadmetele	17
2.4.2 Liikumise jälgimise tehnoloogiad	20
2.4.3 Virtuaal- ja liitreaalsus	20
2.4.4 Sinihammas, raadiosagedustuvastus ja lähiväljaside ning ruutkoodide skaneerimine	21
3 Loodava raamistiku analüüs ja disain.....	23
3.1 Raamistiku nõuete määramine.....	23
3.2 Raamistiku nõuete grupeerimine	23
3.3 Raamistiku nõuete prioriteetide määramine	24
3.3.1 Nõuded raamistikule järgmise arendaja seisukohast.....	24
3.3.2 Füüsilise keskkonna ja kasutatavuse analüüs. Nõuded raamistikule publiku seisukohast.....	26
3.3.3 Nõuded süsteemile administraatori ja etenduse läbi viimise seisukohast	26
3.4 Tehniline analüüs.....	27
3.4.1 Raamistiku täielikkus ja testimine.....	27
3.4.2 Rakenduse komponentide tehnoloogiate valik.....	28
3.4.3 Dokumentatsioon ja versioonihaldus	29
3.5 Prototüüpide disain Figma's	29
3.5.1 Rakenduste siseste protsesside defineerimine	29

3.5.2 Disain.....	30
4 Arendus ja testimine	34
4.1 Arendus.....	34
4.1.1 Taga- ja esirakenduse arendus.....	35
4.1.2 Andmebaasi loomine	36
4.1.3 Turvalisus	37
4.2 Raamistiku testimine	38
5 Hinnang loodule ja võimalused edasi arenduseks	42
5.1 Keeletoe lisamine	42
5.2 Jooksvad teavitused	42
5.3 Etenduse kokkuvõtete saatmise süsteem	42
5.4 Automaattestid.....	43
6 Kokkuvõte	44
Kasutatud kirjandus	45
Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks	49

Jooniste loetelu

Joonis 1. Iteratiivne raamistikul põhineva projekti arendamise protsess [4].....	14
Joonis 2. Foto Teatr Ludowy kasutatud tüvirakendusele ehitatud rakendusest [14].....	19
Joonis 3. Etenduse alustamist kujutav voodiagramm.....	30
Joonis 4. Publiku sisse logimise protsessi kujutav voodiagramm.....	30
Joonis 5. Vasakul esialgne publikupoolse rakenduse disain. Paremal publikupoolse rakenduse disain novembris 2022.....	31
Joonis 6. Halva kontrastiga näide disainist.....	32
Joonis 7. Esimese etenduse publikumängude algse disaini kuvatõmmised.	32
Joonis 8. Esialgne disain publikust ülevaate saamiseks.	33
Joonis 9. Raamistiku arhitektuur.	34
Joonis 10. Tagarakenduse arhitektuur.	35
Joonis 11. Raamistiku olemi-suhte diagrammi kavand.	37
Joonis 12. Publiku registreerimise ja sisse logimise süsteemipoolne voodiagramm.	38

Tabelite loetelu

Tabel 1. Hinnanguline rakenduse arendusele kulunud aeg.	16
Tabel 2. Olulisimate tehnoloogiate toe tähtajad.	16
Tabel 3. Prioriteetideks jaotatud nõudmised raamistikule arendaja seisukohast.....	24
Tabel 4. Prioriteetideks jaotatud nõudmised raamistikule administraatori seisukohast. 26	
Tabel 5. Kindlasti vajalike nõuete täitmise staatused.....	40
Tabel 6. Teisejärguliste nõuete täitmise staatused.....	41

1 Sissejuhatus

Järgmiste aastate jooksul plaanib Eestis tegutsev VAT Teater katsetada rahvusvahelise teatriprojekti "PlayOn!" raames interaktiivsemate ja tehnoloogiliselt keerukamate lavastuste loomist. Seni seda eriti palju tehtud ei ole, puudub nii teatrite sisene tehniline võimekus kui ka ülevaade tehnilistest võimalustest.

Käesolev töö püüab tuvastada mõned võimalused interaktiivsete lavastuste loomiseks, analüüsida nende keerukust ja maksumust ning jõuda järeldustele, millised neist on kõige suurema potentsiaaliga. Saadavat materjali saab rakendada uute lavastuse planeerimisel – lavastusi on hetkel kavandatud järgmise kahe aasta jooksul kaks VAT Teatris (2023 jaanuaris esietenduv "Vivaarium" ning 2023 veel nimeta etendus). Pilootprojekti õnnestumisel on võimalik tarkvara jagada kaheksa teatriga, mis lisaks VAT teatrile rahvusvahelisse ühendusse kuuluvad: Theater Dortmund (Saksamaa), Elsinor Centro di Produzione Teatrale (Itaalia), Kolibri Színház (Ungari), Teatr Ludowy (Poola), Landestheater Linz (Austria), Teatro O Bando (Portugal), Teatret Våt (Norra) ja Pilot Theatre (Ühendkuningriigid) [1].

Lisaks uute võimaluste avastamisele on töö eesmärk luua raamistik, mille põhjal interaktiivse teatri etenduste veebirakendusi arendada ja mida saavad kasutada kõik teatrivõrku "PlayOn!" võrku kuuluvad teatrid.

Töö teises peatükis analüüsitakse probleemi ning seatakse lõputöö jaoks eesmärgid ning skoop, defineeritakse meetodika ning kirjeldatakse raamistiku kavandamise protsessi. Lisaks analüüsitakse ka tehnoloogia kasutamise võimalusi interaktiivses teatris. Kolmandas peatükis defineeritakse nõuded, valitakse lahenduse jaoks tehnoloogiad ning kirjeldatakse prototüüpide disainimise protsessi. Neljandas peatükis kirjeldatakse süsteemi komponentide arendust. Viiendas peatükis kirjeldatakse tulemusi, analüüsitakse arendust, hinnatakse loodut, tehtud parandusi ja järeldusi tagasiside ning testimise põhjal ning planeeritakse võimalusi edasiarenguteks.

2 Probleemi kirjeldus

VAT Teater ja rahvusvaheline teatriühendus PlayOn! kavandavad lähitulevikus interaktiivseid lavastusi, kus publik mõjutab iga etenduse süžeed läbi veebirakenduse oma mobiilseadmetest. Nende lavastuste tarbeks planeeritakse luua raamistik, mis kiirendaks vastavate rakenduste arendust.

Kõik lavastused, mis raamistikku kasutaksid, on erinevad, kuid etendustel on kattuvad ühisosad, mis on raamistiku arendamise põhjuseks. Nende ühisosade hulka kuuluvad näiteks publiku liikmena rakendusse sisse logimine, etenduste loomine või erinevate publikumängude käivitamine.

Raamistik on vajalik nii VAT teatrile kui ka rahvusvahelisele teatriringkonnale tervikuna, kuna see loob kaasaegsele teatrile üle maailma täiesti uue võimaluste kategooria – interaktiivne teater, mis ei nõua lisatehnika soetamist.

2.1 Lõputöö eesmärgid ja skoop

Töö analüütilise osa eesmärk on tuvastada ja analüüsida erinevaid võimalusi, kuidas tehnoloogiat teatrisse integreerida, ning hinnata iga lahenduse keerukust ja kulukust.

Töö praktilise osa eesmärk on luua komplekt baasprojekte ning ülesseadmis- ja arendusjuhendeid, mis moodustavad teatrirakenduste arendamise raamistiku.

Raamistiku põhiline eesmärk on toetada järgmiste projektide teatrirakenduste valmimist. Raamistik peab lahendama mitmed igas rakenduses kattuvad alad, nagu näiteks publiku liikmete registreerimine ning etenduse alustamine ja lõpetamine. Igas rakenduses kattuvad alad defineeritakse nõuete kogumise ja analüüsimise käigus. Põhilised nõuded selgitati välja kevadel ja suvel aastal 2022.

Raamistiku loomine erineb lihtsalt baasprojekti loomisest mitme aspekti poolest. Tarkvaraprojektide puhul pakuvad üldjuhul raamistikud lihtsustuskihti mingile keelele, näiteks Django Pythonile ning ExpressJS JavaScriptile. Kuna lõputöös käsitletav

raamistik ei kuulu keelt lihtsustavate raamistike hulka, tuleks loodavat raamistikku analüüsida tervikuna, mitte ainult programmeerimise seisukohast.

Raamistiku skooopi kuuluvad esi- ja tagarakendus, andmebaas, arenduskeskkondade seadistused ning dokumentatsioon ja juhendid, kuidas uusi projekte raamistiku abil luua. Raamistik saab olema avalik teatrirühmitusse kuuluvatele teatritele.

2.2 Metoodika

Käesoleva lõputöö käigus selgitatakse lahti eksisteeriv probleem ning pakutakse välja probleemile sobiv lahendus, mis jääks seatud skooobi piiridesse, kuid võimaldaks edasiarendusi tulevikus.

VAT Teatri tiimiliikmed alustasid ajurünnakutest ja struktureerimata koosolekutest, kus pandi paika lavastuse visioon. Nende põhjal pandi kirja vabas vormis lavastuse faaside kirjeldused, millest tuletatakse rakenduse nõuded.

Paika pandud nõuded kogutakse kokku, grupeeritakse funktsionaalseteks ja mittefunktsionaalseteks nõueteks; eristatakse nõuded raamistiku arendaja seisukohalt ning nõuded funktsionaalsusele teatritarkvara seisukohalt; määratakse nõuete prioriteetid MoSCoW [2] meetodi abil.

Seejärel luuakse esimesele etendusele vastavad prototüübid, mis on abiks kontseptsioonide läbitöötamisel, valitakse välja tehnoloogiad, millega raamistiku osad luua, arendatakse raamistiku osad ning kirjutatakse dokumentatsioon. Raamistiku testimine ja edasiarendus toimuvad iteratiivselt ning esimest korda saab raamistiku tegelikku kasu hinnata alles järgmise lavastuse arenduse ajal.

2.3 Raamistiku kavandamine

Raamistiku kavandamisel tuleks jälgida kindlat protsessi, läbi mille [3]:

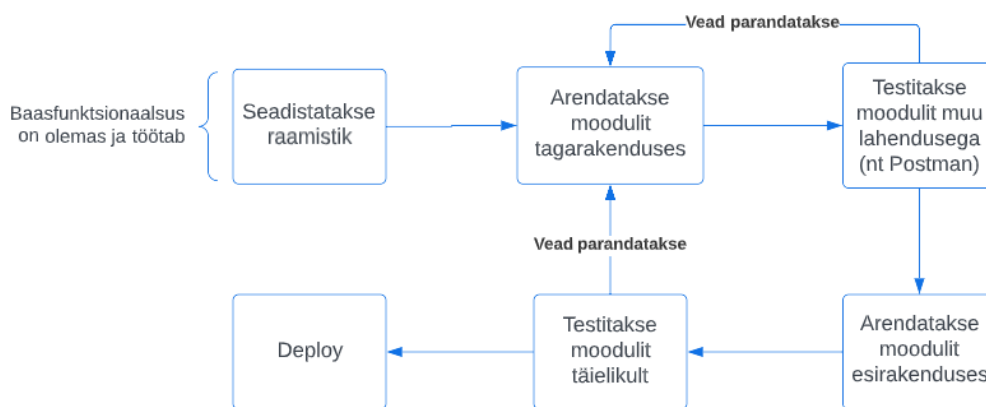
1. kirjeldatakse, kuidas raamistik kasutamiseks on kavandatud;
2. kirjeldatakse probleemi, mida raamistik lahendab, ja planeeritud tulusid, mida raamistiku kasutamine toob;
3. kirjeldatakse raamistiku eesmärgid;
4. määratakse raamistiku skoop;

5. tuvastatakse kõik komponendid, mis raamistikku kuuluvad;
6. planeeritakse raamistikusised protsessid näiteks läbi jooniste;
7. analüüsitakse ja testitakse raamistiku täielikkust ja vajadusel tehakse muudatusi.

2.3.1 Raamistiku kasutamise visioon

Raamistiku vajalikkus selgus siis, kui ilmnes, et sarnaseid lavastusi planeeritakse teha järgnevate aastate jooksul mitu. Kui defineerida visioon, võiks seda ühes lauses teha järgnevalt: Raamistik hõlbustab arendust, võimaldab luua lavastuste jaoks rakendusi rohkematel organisatsioonidel ning uuendab teatritehnoloogiat.

Rakenduse raamistiku baasil arendamise protsess on kujutatud joonisel 1. Planeeritav raamistiku kasutamise protsess peaks järgima iteratiivset mudelit, kus alustatakse raamistiku seadistamisest ning seejärel arendatakse esi- ja tagarakendust paralleelselt ning tsüklite kaupa [4].



Joonis 1. Iteratiivne raamistikul põhineva projekti arendamise protsess [4].

2.3.2 Probleemid, mida raamistik lahendab

Tarkvaraarenduses on tavaks minimeerida koodi- ja loogikakordusi ning luua võimalikult universaalseid lahendusi. Seda tehakse mitmel eesmärgil [5]:

- püütakse hoiduda üleliigsest tööst;

- suurendada rakenduste töökindlust;
- lihtsustada rakenduste hallatavust läbi väheste vajalike muudatuste;
- lihtsustada koodi loetavust;
- juurutada *clean code* praktikaid ja korrektseid koodimustreid.

Samu eesmärgid täidab ka raamistiku loomine.

Üleliigne töö teatrarakenduste loomisel tähendab kõigi lavastuste ühisosade, mida programm peab lahendama, taaskirjutamist. Ühisosade hulka kuuluvad publiku rakendusse sisselogimine, etenduse instantside haldus, iga etenduse faaside haldus, etenduse lõpetamise protsessid ja muu. Sellest lähtuvalt peaks raamistik pakkuma mähisfunktsioone, mis võimaldaksid lihtsustada tavapäraseid veebitoiminguid (päringute tegemine, vormide täitmine, vea- ja infoteadete logimine jne), ning nimetatud ühiseid katvaid funktsioone. Lisaks peaks raamistikuga lahendatud olema levinud ülesanded nagu arenduskeskkondade seadistamine, konteineriseerimine ja arvutivõrkude konfigureerimine.

2.3.3 Raamistiku eesmärgid

Üks raamistiku eesmärkidest on hoida kokku projektide arenduseks kuluvat aega. Aega, mis kulub tarkvaraprojektide arendusele, on keeruline määrata [6], kuid põhifunktsionaalsuste olemasolu ja ettevalmistatud konteineriseerimise lahendused kiirendavad mitmekordselt rakenduse arendamist. Kuna raamistiku arendusprotsess ja testimine toimuvad iteratiivselt, on võimalik raamistiku abil uue projekti arendusele kuluvat aega ainult umbkaudselt hinnata. Esialgne hinnang on kujutatud tabelis 1. Tõeline edukus selgub alles 2023. aasta kevadel, kui algab järgmise projekti arendus.

Tabel 1. Hinnanguline rakenduse arendusele kulunud aeg.

Raamistiku osa	Umbkaudne arendusele kulunud aeg
Ettevalmistus ja nõuete selgitamine	25h
Tagarakendus	60h
Esirakendus	60h
Virtualiseerimise ja rakenduse käivituse seadistus	10h

Raamistiku teine eesmärk on seadistuse hõlbustamine, mis jaguneb kaheks: arenduskeskkondade ja funktsionaalsuse seadistus ning rakenduse välimuse seadistus. Seadistuse hõlpsus saavutatakse keskkonnamuutujate abil, mis konfigureeritakse igas projektis eraldi. Võrkude ja virtualiseerimise seadistus peaks olema samuti raamistiku poolt lahendatud. Rakenduse välimise seadistus saavutatakse läbi SCSS (*Syntactically Awesome Style Sheets*) muutujate, millega saab defineerida näiteks primaar- ja sekundaarvärve rakenduses, elementide nurkade ümarust või tekstistiile.

Iga raamistiku osa eesmärk on ka tagada võimalikult pikk raamistiku kasutusaeg. See tähendab, et raamistik peaks kasutama pikaajalise toega (*long-term support - LTS*) tehnoloogiaid [7]. Teatriprojekte planeeritakse ette üsna pikalt – üldiselt mitu aastat. Kuna hetkel on juba kahe aasta plaanid ette tehtud, peavad kõik projektis kasutatavad keeled ja teegid kehtima vähemalt nii kaua, aga ideaalis kauem. Mõned tehnoloogiad on aktiivses kasutuses ning nende jaoks ei ole saadaval ei LTS ega täieliku toe lõpetamise (*end of life, EOL*) tähtaega. Töö käigus kasutatud tehnoloogiate toe kasutusse võetavate versioonide toe lõpptähtajad on toodud tabelis 2.

Tabel 2. Olulisimate tehnoloogiate toe tähtajad.

Tehnoloogia	Kasutus raamistikus	Versioon	LTS tähtaeg	EOL tähtaeg
NodeJS	JavaScripti käitussüsteem	v18.12.0 Hydrogen	oktoober 2023	aprill 2025
TypeScript	Programmeerimiskeel	v4.9.3	puudub, planeerimata	-
VueJS	Esirakenduse põhiraamistik	v3.2.45	puudub, planeerimata [8]	-
ExpressJS	Tagarakenduse põhiraamistik	v4.18.1	puudub, planeerimata [9]	-
MongoDB	Andmebaasimootor	v6.1	puudub, planeerimata	varaseim juuli 2025 [10]

2.3.4 Raamistiku komponendid

Raamistiku komponendid võib jagada tehnoloogilisteks ja toetavateks komponentideks. Tehnoloogiliste komponentide hulka kuuluvad tagarakendus, esirakendus, andmebaasi seadistus ja konteineriseerimise ning võrkude seadistuse lahendus. Toetavate komponentide hulka kuuluvad käesolev töö, raamistiku dokumentatsioon ja kasutusjuhend, raamistiku voodiagrammid ja protsesside kirjeldused.

2.4 Tehnoloogia kasutamise võimalused interaktiivses teatris

VAT teatri töötajatega läbi viidud intervjuude ning rahvusvaheliste teatriajakirjade, arvustuste ning uurimuste ja PlayOn! kokkutulekul (november 2022) kuulnud ettekannete ja arutelude põhjal viidi läbi analüüs kaasaegses teatris kasutatavatest võimalikest interaktiivsetest tehnoloogiatest.

Analüüsis pakutakse võimalikke tehnoloogiaid, mida saaks interaktiivses teatris kasutada, jättes defineerimata kunstilise tõlgenduse, kuidas tehnoloogiaid lavastusse integreerida. Oluline on iga tehnoloogia puhul eristada piiri immerstiivse (lava ja publiku piir kaob ning publik on samamoodi laval, nagu näitlejad [11]) ja interaktiivse (publikul palutakse etenduses osaleda, tihti etenduse käiku mõjutades [12]) teatri vahel – käesoleva töö raames analüüsitakse ainult interaktiivseid lahendusi. Üks eripära interaktiivse teatri juures on etenduse võimalik sarnasus installatsiooniga, mis mõnikord ei vaja isegi näitlejate olemasolu.

Tehnoloogia kasutamist teatris analüüsitakse järgmistes punktides:

- potentsiaalsed kasutusjuhud;
- kohaspetsiifilisus;
- ülesseadmise keerukus esmasel ülesseadmisel ja eri etendustel;
- ajakulu (sh spetsialisti ajakulu);
- tehnika, mida on tarvis soetada.

2.4.1 Veebirakendused publiku mobiilseadmetele

Veebirakendusi etenduse jaoks on võimalik kasutada mitut moodi:

- arendada rakendus spetsiaalselt etenduse jaoks;

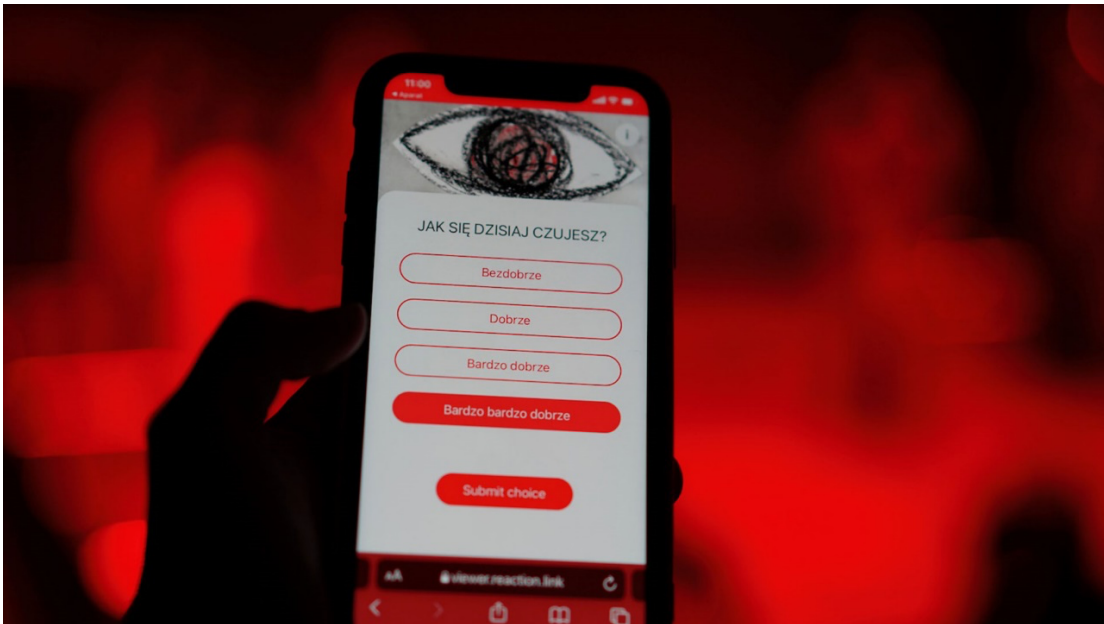
- kasutada mõnda juba eksisteerivat rakendust interaktiivsuse loomiseks, nt Kahoot [13] valikute tegemiseks;
- vahepealne variant, kus võetakse kohandatav tüvirakendus ning kohandatakse see konkreetsele etendusele.

Spetsiaalselt lavastuse jaoks terve rakenduse arendamine on kõigist variantidest kõige kallim ja töömahukam, kuid rakendus saab kõige mitmekülgsem ja võimekam. Käesolev töö on seotud sellise rakenduse arendusega. Jaanuaris esietenduva lavastuse rakendus võimaldab näiteks mobiilseadmega QR-koodide skaneerimist, küsitlusi ning publiku vastuste põhjal küllastajate gruppideks jagamist – need on vaid mõned ideed, mida rakendusse ehitada saab.

Ajakulu ning hind sõltuvad ise arendamise puhul rakenduse keerukusest, kuid arvestada tuleks minimaalselt kolme kuuga ühe arendaja puhul. Kuludele lisanduvad serveri- ja domeenikulud ning väliste integreeritavate lahenduste kulud. Füüsiliste objektide kulud võivad samuti olla märkimisväärsed – ehkki mitte otseselt lavastusega seotud, on vaja ikkagi arvestada igat tüüpi varulaadijatega ning varuseadmetega ning internetiühenduse tagamisega ruuterite näol.

Teiste variantide puhul on üldiselt töömaht palju väiksem, sest mitmeid selliseid lahendusi saab seadistada ka keegi, kes ei ole programmeerija. Küll aga võivad sellised lahendused ise olla tasulised ning need ei võimalda keerulisemaid ideid teostada.

Teoreetiliselt on eksisteeriva rakenduse kasutamine võimalik, kuid võib veidi rikkuda n-ö "teatrimaagiat", kus sellisel juhul siseneks saali või etenduspaika midagi liigselt välismaailma meenutavat. Lihtsamat varianti kohandatava tüvirakendusega kasutasid mõned PlayOn! rühmituse liikmed juba varasemalt [14] – foto Poola Teatr Ludowy rakendusest teatrikeskkonnas on toodud joonisel 2. Samuti on sarnaseid lahendusi kasutanud Eestis Vene Teater [15] ning Cabaret Rhizome [16].



Joonis 2. Foto Teatr Ludowy kasutatud tüvirakendusele ehitatud rakendusest [14].

Üks lisaboonustest veebirakenduse kasutamise puhul on see, et kui mitmed tehnoloogiad on võõrad ja natuke hirmutavad või ebamugavad kasutada, siis oma isiklikus mobiilseadmes tunnevad inimesed end kodusemalt. Siinkohal peame välistama täiesti tehnoloogivõõrad inimesed.

Veebirakendus või mobiilirakendus on ka eelduseks mitmetele järgnevatele tehnoloogiatele, mis hõlmavad kas publiku mobiilseadme või teatri poolt pakutud mobiilseadme kasutust.

Suuremad probleemid võivad tekkida sellest, et külalised hakkavad oma mobiilseadmes sooritama kõrvalisi tegevusi. Kuna rakenduse töötamiseks peab internetiühendus telefonis sees olema, tulevad paratamatult telefonile teavitused kõrvalistest rakendustest. Võimalik lahendus on luua teatris traadita kohtvõrk (WLAN, *wireless local area network*) ning selles võrgus ka rakendust käivitada. See garanteeriks ühenduse kao internetiga, tänu millele publik saaks oma teavitused alles mõnda internetivõrku ühendudes. Selle lahendusega võib aga probleemiks osutuda inimeste usaldamatus, mistõttu nad ei pruugi soovida seda teha.

2.4.2 Liikumise jälgimise tehnoloogiad

Erinevad andurid ja muud liikumise jälgimise tehnoloogiad võimaldavad mängida ruumiga, on selleks teatrisaal või mingi muu asukoht, kuid eristus tekib seal, kas jälgitakse publiku või näitlejate liikumist. Teatrietendus on interaktiivne vaid siis, kui publik otseselt või kaudselt etenduse käiku mõjutab, seega näitlejate jälgimine siinkohal analüüsi ei puutu [12]. Küll aga juhul, kui publiku liikumine etendust või lavakujundust mõjutab, on tegemist interaktiivsusega.

Cutting Room projekt kasutas liikumisandureid, et projekti külastajad saaksid valida, mis tunne neil seoses produktsiooni loo suunaga on, ning see mõjutas installatsioonivideote käiku. Cutting Roomi projekti puhul on tegemist pigem installatsiooni kui lavastusega, kuid sarnast tehnoloogiat saaks kasutada ka interaktiivsel lavastusel [17].

Liikumisanduritega etendusi saab teha nii kohaspetsiifiliselt kui ka mitte – modulaarsema süsteemi puhul on võimalik nendega üsna kergelt reisida. Andurite soetamine ja ülesseadmine ei ole võrreldes n-ö targemate tehnoloogiatega raha- ega ajakulukas – näiteks november 2022 seisuga algavad lihtsamate liikumisandurite hinnad Eestis kuuest eurost [18]. See, mida need andurid mõjutavad ja millega ühenduvad, defineerib lahenduse hinna. Liikumisandureid saaks näiteks kombineerida visuaalse tagasisidega lavakujunduses.

Keerukamad liikumise jälgimise tehnoloogiad (*motion tracking*), näiteks kehast avataride loomine ja nende etenduse käigus jooksvalt ekraanile projekteerimine on tehniliselt keerulisem, kulukam ning aeganõudvam üles seada, kuid võimalused on see-eest mitmekülgsemad. Selliseid tehnoloogiaid on juba ka kauem kasutatud, kui näiteks virtuaalreaalsust – animeerijad kasutavad liigutuste jaoks sarnast tarkvara, ning lahenduse jaoks eksisteerib kommertstarkvara [19].

2.4.3 Virtuaal- ja liitreaalsus

Teater, mis toimub üleni virtuaalreaalsuses, meenutab videomängu – kuid tegelaste asemel on päristegelased. Siin tuleb jällegi meeles pidada, et virtuaalreaalsuses aset leidev etendus ei pruugi olla interaktiivne, kui publiku panus etenduse käiku ei muuda. Küll aga muudab virtuaalreaalsus interaktiivsuse sisse- ja väljalülitatavaks, mis võib olla näitlejatele ja lavastajatele väga mugav lahendus. Virtuaalreaalsuses lavastuste etendamise suurim probleem on tehnika hind.

David Gochfeldi sõnul on virtuaalreaalsuses näitlemine üsna erinev „tavalisest“ näitlemisest. Näitleja peab esinema nendele tehnoloogiatele, mis salvestades või üle kandes tema küljes on¹, ja ta peab teadma, kuidas need virtuaalreaalsuses üle kanduvad. Ta peab ka jälgima, et kogu tehnoloogia toimiks [20].

Teatri poolt hangitud seadmetesse on võimalik paigaldada ka mõni 3D skaneerija, mis võimaldaks publikul luua virtuaalreaalsuses kasutatavaid objekte. Nõnda saaks publik aktiivselt osaleda näiteks lava kujundamises.

Liitreaalsuses on võimalik reaalsusesse kuvada kujundeid ja tegelasi, mida näeb läbi prillide või ekraaniga seadme. Need objektid võivad tekkida publiku lähenemisel (interaktiivsus läbi positsiooni) või muutuda publiku sisendi põhjal. Siin läheneb teatriprojekt jälle installatsioonile.

2.4.4 Sinihammas, raadiosagedustuvastus ja lähiväljaside ning ruutkoodide skaneerimine

Nende tehnoloogiate tööpõhimõtte teatris oleks see, et publik jõuab mingi füüsilise punktini ning käivitab sellega mingi sündmuse jada. See tähendab, et publik peab endaga füüsiliselt kaasas kandma tehnoloogiat, mis registreerub vastuvõtjaga.

Sinihammas edastab kogu aeg oma ID-d [21], ning kui vastuvõttev seade selle registreerib, saab hakata mängima näiteks audiofail või võib alata augmenteeritud reaalsuse osa etendusest. Bluetooth on väga täpne, mistõttu seda kasutatakse väga palju tarkades kodudes, kus mõõtmisviga peab olema minimaalne [21]. Lisaks saab Bluetooth signaali suunata, mis võib näiteks mitme publiku liikme puhul lisada täpsust.

Raadiosagedustuvastus (RFID ehk *radio-frequency identification*) [22] ja lähiväljaside (NFC ehk *near field communication*) kiipide puhul on põhimõtte sarnane – kui kiip on käes, siis publiku jõudmisel kuhugi midagi juhtub. NFC puhul peab publik kiibi asetama väga lähedale (maksimaalselt kahe sentimeetri kaugusele lugejast) [23].

¹ Virtuaalreaalsuses täielikult osaledes on näitleja külge kinnitatud üsna palju tehnikat: kindad või käeandurid, VR ülikond keha jälgimiseks, mikrofoni lindistamiseks, kaamera koos statiiviga näo filmimiseks.

Ruutkoodide (*QR code* ehk *quick response code*) puhul peab publik interaktiivsuses aktiivsemalt osalema ehk midagi oma seadmega skaneerima. QR-koodid on väga töökindlad ja võimaldavad edastada väga palju infot [24] – teatris enamasti nii palju vaja ei ole. Ka näiteks käesoleva töö ajend, etendus "Vivaarium", kasutab QR-koode, et esimeses publikumängus kujuteldavas poes ostelda.

Bluetoothi ülesseadmise kulusid ei ole, kuid vajalikud on seadmed. RFID ja NFC kiibid on võrdlemisi odavad [25] [26]¹, ning lahenduse maksumus sõltub jälle sellest, mida nendega tehakse. QR-koodid on tasuta vabavaraline tehnoloogia.

¹ NFC Pood pakub erilahendusi, mille alla teatriprojekt kindlasti kuuluks. Kui vaadata toote "NFC Komplekt - PRO" maksumust, pakub autor, et erilahenduse hind teatri mastaabis väga kulukas ei ole.

3 Loodava raamistiku analüüs ja disain

Käesolevas peatükis defineeritakse raamistiku nõuded. Vastavus nendele nõuetele määrab raamistiku täielikkuse ja edukuse.

3.1 Raamistiku nõuete määramine

Nõuete määramine algab 2022. aasta kevadel, kui etenduse autorite tuumik esimest korda koguneb. Läbi nende kogunemiste, mis korduvad kuni novembri lõpuni 2022, luuakse etenduse plaan ning määratakse veebirakenduse roll ning nõuded. Seejärel eraldatakse raamistiku nõuded esimese etenduse nõuete omadest.

Raamistiku nõuded kogutakse kokku vabas vormis märkmetesse, kust edasi hakatakse nõudeid grupeerima ning analüüsima. Nõuded grupeeritakse mittefunktsionaalseteks ja funktsionaalseteks ja siis seatakse prioriteetid MoSCoW [2] meetodit kasutades.

3.2 Raamistiku nõuete grupeerimine

Raamistiku nõuded jaotatakse esiti kahte gruppi:

1. mittefunktsionaalsed nõuded raamistiku kasutaja e järgmise arendaja seisukohast;
2. funktsionaalsed nõuded lõppkasutajate seisukohast teatris.

Nii grupeeritakse nõuded, sest selles jaotuses ilmneb suurim kontseptuaalne erinevus. Nõuded järgmise arendaja seisukohast on metanõuded, mis hõlmavad dokumentatsiooni, koodi taaskasutatavust, konteineriseerimist ja turvalisust. Nõuded teatripoolsete kasutajate seisukohast on seotud raamistiku võimekuse ja funktsionaalsusega, näiteks administraatorite ja publiku kasutajate haldus ning etenduste haldus.

Funktsionaalsed nõuded jagatakse edasi kasutajagruppide järgi:

1. administraatorid, kes juhivad etenduste faaside käiku ja saavad läbi töölaua pidevat tagasisidet etenduse jooksul toimunust;
2. publik, kes kasutab raamistiku põhjal ehitatud rakendust etenduse jooksul.

3.3 Raamistiku nõuete prioriteetide määramine

MoSCoW meetodi [2] põhjal jagatakse varasemalt grupeeritud nõuded prioriteetide järgi tabelisse, mis määrab iga nõude olulisuse ja seeläbi ka arendusjärjekorra.

3.3.1 Nõuded raamistikule järgmise arendaja seisukohast

Eelnevast alapeatükist tuletatud ja kokku kogutud nõuete prioriteetid määratakse MoSCoW meetodit kasutades. Need nõuded on aluseks hiljem raamistiku edukuse mõõtmisel. MoSCoW meetodit kasutades jagatakse kõik kogutud nõuded nelja gruppi [2]:

- *must have* ehk nõuded, mis on kindlasti vajalikud;
- *should have* ehk nõuded, mis peaksid olema;
- *could have* ehk nõuded, mis saaksid olla;
- *won't have* ehk nõuded, mis jäävad skoobist välja.

Kõige olulisemad *must have* nõuded on aluseks minimaalsele aktsepteeritavale tulemusele, mille täitmiseta ei saa analüüsitud ülesannet lugeda täidetuks. Järgmise arendaja seisukohast leitakse kõige rohkem just selliseid nõudeid.

Nõuded raamistiku järgmise arendaja seisukohast on kujutatud tabelis 3.

Tabel 3. Prioriteetideks jaotatud nõudmised raamistikule arendaja seisukohast.

<i>Must have</i>	<i>Should have</i>
<ul style="list-style-type: none">▪ raamistik peab kasutama LTS versioone kõigist kasutusel olevatest keeltest ja tehnoloogiatest;▪ peab eksisteerima dokumentatsioon ja ülesseadmisjuhend kõigile PlayOni liikmetele;	<ul style="list-style-type: none">▪ arendusserveri püstipanek peab olema kerge ja kiire;▪ disain peab olema lihtsalt konfigureeritav (S)CSS-i muutujate kaudu, et see sobiks iga etenduse stiiliga;

<i>Must have</i>	<i>Should have</i>
<ul style="list-style-type: none"> ▪ peavad olema olemas administraatoritele mõeldud liidesed konkreetsete toimingute haldamiseks ja vajalike andmete või konfiguratsioonide muutmiseks/värskendamiseks iga etenduse instantsi jaoks; ▪ peab olema olemas garderoobinumbritel ja valikulisel meiliväljal põhinev sisselogimissüsteem; ▪ peab olema olemas turvaline andmelahendus tundlike andmete salvestamiseks; ▪ peab olema olemas etenduste haldussüsteem, millel on faasid esituse erinevate osade jaoks, et võimaldada mängu ajal paindlikkust; ▪ peavad olema olemas lihtsasti seadistatavad keskkonnad ja visuaalsete elementide põhitunnused; ▪ peab olema olemas peamine API ja võimalus hõlpsasti mooduleid lisada uute poodide malli abil, et hõlbustada seadistamist. 	<ul style="list-style-type: none"> ▪ olemas olema peavad mallid erinevate rakenduse osade kiireks seadistamiseks; ▪ olemas peab olema konfigureeritud HTTPS-baasil kohaliku arenduse võimalus.
<i>Could have</i>	<i>Won't have</i>
<ul style="list-style-type: none"> ▪ peab olema lahendatud tõlgete lisamise süsteem nt Localise tarkvara abil. 	

3.3.2 Füüsilise keskkonna ja kasutatavuse analüüs. Nõuded raamistikule publiku seisukohast

Publikupoolse klientrakenduse arendusel tuleb arvesse võtta, et teater on ebatavaline keskkond nutirakenduste kasutamiseks, millel on unikaalsed eripärad. Selles alapeatükis analüüsitakse neid eripärasid ning pakutakse võimalikke lahendusi. Kõik need lahendused on MoSCoW süsteemis *must have* ehk kindlasti vajalik funktsionaalsus.

Harjumuspäraselt on teatrites pime, kuid sedasorti etenduste puhul tuleb kontrasti, rakenduse värvide ja silma võrkkestale jääva valgusjäljega arvestada nii arendajal kui lavastustiimil. Liiga suur kontrast kasutajate ekraanide ning lavakeskkonna vahel võib tekitada ebamugavaid või ka ohtlikke olukordi näiteks ebaühtlaste põrandapindade või lavadekoratsioonidega; liiga väike kontrast võib pärssida ligipääsetavust [27].

Iga rakenduse arendusel tuleb arvestada erinevate telefonide võimekuse ja ekraanisuurusega, ja isegi rakendus, mida kasutatakse mõni tund ühe õhtu jooksul ettemääratud ajavahemikus peab olema valmis igaks telefoniks, mis võib saali sattuda [28]. Rakendus peab olema kohanduv igale ekraanisuurusele, töötama kiiresti ning minimeerima kasutajate ärevust läbi mikrointeraktsioonide ning laadimisanimatsioonide [29]. Eriti oluline on rakenduse publikupoolne osa, mida kasutatakse ainult mobiilis.

Publiku hulgas võib olla igas vanuses ja iga kultuurilise ja tehnoloogilise taustaga inimesi. See tähendab, et rakendus peab olema läbi mõeldud kasutajakogemuse seisukohast: selge *microcopy* (tarkvaras leiduvad lühikesed tekstiosad, näiteks nuppude nimetused), taktiline tagasiside ning kasutaja eksimisvõimaluste minimeerimine on määrava tähtsusega.

3.3.3 Nõuded süsteemile administraatori ja etenduse läbi viimise seisukohast

Nõuded administraatori ja etenduse läbi viimise seisukohast on MoSCoW meetodit kasutades kujutatud tabelis 4.

Tabel 4. Prioriteetideks jaotatud nõudmised raamistikule administraatori seisukohast.

<i>Must have</i>	<i>Should have</i>
------------------	--------------------

<ul style="list-style-type: none"> ▪ administraatorid peavad saama sisse logida emaili ja parooliga; ▪ administraatorid peavad nägema publiku liikmetest ülevaadet; ▪ administraatorid peavad saama käivitada ja peatada etenduse faase; ▪ administraatorid peavad saama lisada etenduse instantse (st kuupäevi, mil etendus toimub). 	<ul style="list-style-type: none"> ▪ etendust alustades peab rakendus automaatselt looma uued mängude instantsid koos vajalike andmetega; ▪ esimene faas peab käivituma automaatselt; ▪ etenduse lõpetamine peab arhiveerima publiku ja nendega seonduva.
Could have	Won't have
<ul style="list-style-type: none"> ▪ publiku liikmed peavad saama tagantjärele oma emaili lisada; ▪ administraatorid peavad saada publikule jooksvalt teavitusi saata. 	<ul style="list-style-type: none"> ▪ etenduse lõpetamisel peab tekkima võimalus eksportida publiku emailid.

3.4 Tehniline analüüs

Käesolevas alapeatükis teostatakse tehniline analüüs, põhjendatakse tehnoloogia valikuid ning kirjeldatakse töö tulemuseks valmiva raamistiku arhitektuuri.

3.4.1 Raamistiku täielikkus ja testimine

Esimene veebirakendus raamistiku põhjal luuakse peaaegu paralleelselt raamistiku loomisega ja seeläbi rakendatakse agiilseid ja disainmõtlemise metoodikaid. Disainmõtlemise viis faasi (mõistmine ja empaatia, probleemi defineerimine, ideede genereerimine, disainimine ja prototüüpimine, testimine ja elluviimine [30]) on raamistiku ja esimese rakenduse arendusel läbivad. Faaside läbimine ei ole lineaarne ning pigem mõeldakse faasidest kui tööviisidest, mille vahel vahetatakse [31] – raamistiku arendusprotsessi alguses vahetati rohkem empaatia loomise, defineerimise ja ideede genereerimise vahel, ning lõpusirgel luuakse prototüüpe rakendusest ning testitakse neid paralleelselt.

Raamistiku täielikkust analüüsitakse läbi testimise ning esimese rakenduse edukuse. Raamistik on täielik, kui see täidab kõiki *must have* nõudeid ning kergendab arendustööd.

Tõeline testimine on võimalik alles järgmise projekti arendusega – hetkel saab täielikkust vaid umbkaudselt hinnata.

3.4.2 Rakenduse komponentide tehnoloogiate valik

Veebirakenduse puhul on tehnoloogia valik määrava tähtsusega ning sõltub mitmest olulisest faktorist. Koodi taaskasutatavuse ja võimalikult suure paindlikkuse nõudele vastamiseks otsustatakse eraldada esi- ja tagarakendus, mis annab võimaluse soovi korral kummagi täielikult asendada. Seega peaks pinu koosnema esi- ja tagarakenduse tehnoloogiast, andmebaasist ning käitussüsteemist.

Pinu valikul lähtutakse kahest aspektist:

- pinu tehnoloogiad peaksid olema omavahel kokkusobivad, nende kohta peaks olema korralik dokumentatsioon ja piisav kogus õppematerjale;
- pinu kõik või enamus osad peaksid olema võimalikult tuttavad töö autorile.

Lisatingimusena peab esirakenduse tehnoloogia võimaldama mitmeid väliseid teeke kasutada, et valmistuda erinevateks vajadusteks lavastuste planeerimisel.

Esiti valitakse tagarakenduse keel ning raamistik. Kuna autori tugevaim keel on JavaScript, valitakse väga populaarne [32] serveripoolse arenduse raamistik ExpressJS. Järgmisena on oluline andmebaasi valik ning andmebaasiga suhtlemise lahendus. Selleks sobib populaarne Mongoose, mis ühendub MongoDB andmebaasiga ja sobib ExpressJSiga [33]. Viimasena valitakse esirakenduse raamistik: ehkki VueJS on populaarsuselt kolmas [32] (mis tähendab, et seda kasutab pea neli miljonit veebilehte [34]) esirakenduse raamistik, on otsustav autori laialdane kogemus just selle raamistikuga.

Nende tingimuste põhjal valitakse MEVN pinu, mis on populaarne kombinatsioon rohkete abimaterjalide ja põhjaliku dokumentatsiooniga [35].

MEVN pinu, mille nimi on akronüüm, koosneb neljast tehnoloogiast:

- MongoDB – andmebaas [33];
- ExpressJS – tagarakenduse põhiraamistik, mis laiendab NodeJS võimekust veebirakenduste loomiseks [9];
- VueJS – esirakenduse põhiraamistik [8];

- NodeJS – avatud lähtekoodiga JavaScripti käitussüsteem, mis võimaldab luua veebiservereid [36].

Lokaalses keskkonnas kasutatakse arenduseks npm [37] ja Vite [38] tööriistu, mis võimaldavad jooksvalt rakendust värskendada ning arendusprotsesse kiirendada.

3.4.3 Dokumentatsioon ja versioonihaldus

Dokumentatsioonilahenduse valikul lähtutakse kolmest peamisest aspektist:

- hind (ideaalne lahendus on vabavara, st tasuta);
- automaatika (kui palju lahendus ise ära teeb ja automaatselt uueneb);
- visuaal (kui esteetiline on lahenduse vaikimisi seadistus).

Dokumentatsioonilahenduseks valitakse Material for MkDocs [39], sest see on vabavara, on lihtne üles seada ja kirjutatakse Markdownis, mida töö autor valdab hästi ja mida on võimalik kiiresti kirjutada, ning baseerub Material Designil [40]. Esialsed dokumentatsioonimustandid asuvad koodikaustades iga mooduli sees.

Versioonihalduseks kasutatakse Giti [41, 41], mis on kõige levinum versioonihaldustööriist maailmas [42], ja pilves versioneerimise säilitamiseks Gitiga ühilduvat GitHubi [43].

3.5 Prototüüpide disain Figma

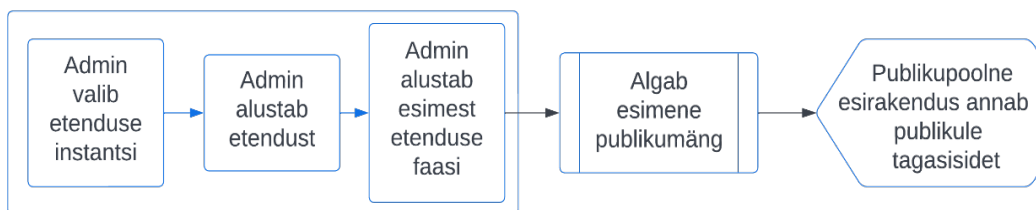
Järgnevates alapeatükkides kirjeldatakse prototüüpide loomise protsessi.

3.5.1 Rakenduste siseste protsesside defineerimine

Esiagu kirjeldatakse rakenduse osade käitumist tervikuna eristamata esi- ja tagarakenduse rolle. Need kirjeldused tõlgendatakse voodiagrammideks. See aitab selgelt määratleda sammud, mida rakendus peab täitma, ning annab ette plaani kasutajaliideste prototüüpide loomiseks. Voodiagrammid tehakse erinevatele kasutajalugudele eraldi eesmärgiga anda kõrgetasemeline ülevaade igal kasutajaloo toimumise sammul.

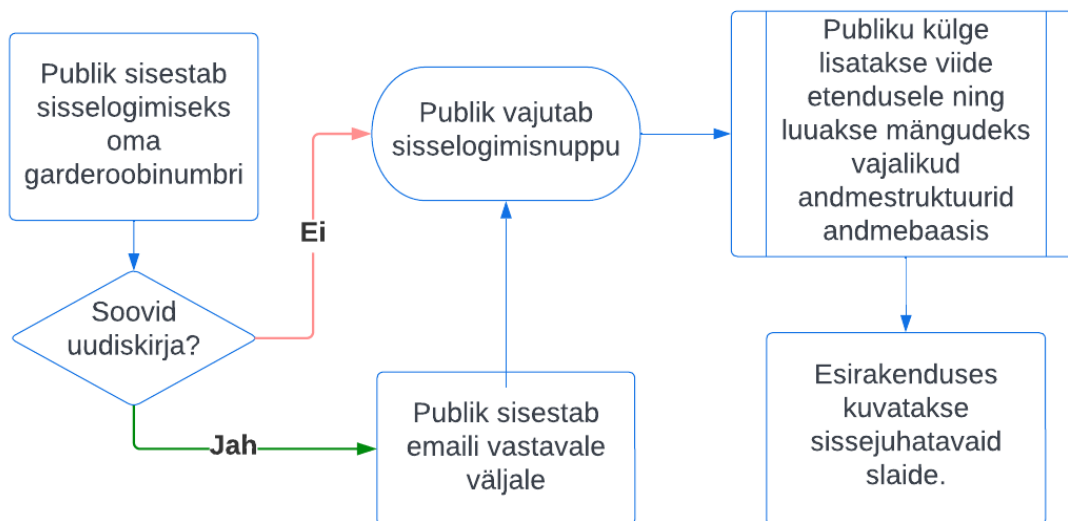
Joonisel 3 kujutatakse voodiagrammi, mis näitab valitud samme nii tagarakenduses kui esirakenduses toimuvast etenduse instantsi alustamisel. Esirakenduses aktiveerib administraator (lavatehnik või näitleja) selle päeva etenduse, alustab etenduse esimest

faasi ning tagarakenduses käivitatakse esimene publikumäng. Kui publikumängu käivitamine on õnnestunud, antakse publikule nende mobiilseadmes tagasisidet.



Joonis 3. Etenduse alustamist kujutav voodiagramm.

Joonisel 4 kujutatakse publikupoolsesse rakendusse sisselogimise protsessi. Publiku liige sisestab oma garderoobinumbri ning soovi korral emaili, logib sisse, ning talle kuvatakse sissejuhatavaid samme. Tagarakenduses luuakse andmebaasi lisaks uuele kasutajale kõik etenduse toimimiseks vajalikud andmestruktuurid, näiteks "Vivaariumi" puhul mängupoe ostukorv.

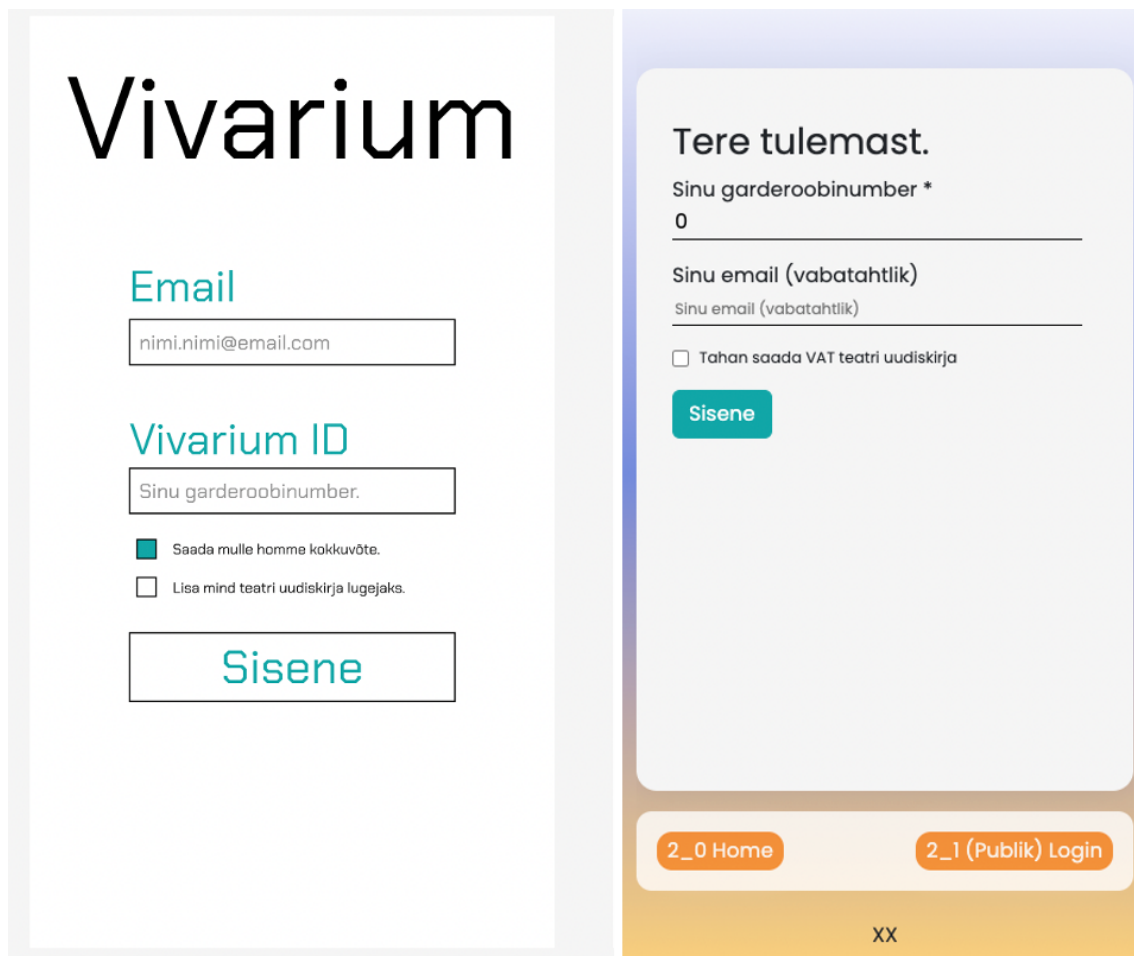


Joonis 4. Publiku sisse logimise protsessi kujutav voodiagramm.

3.5.2 Disain

Rakenduse prototüüpide disainimiseks kasutatakse Figma tarkvara, sest prototüüpimisvahenditest on autoril sellega suurim kogemus ning see rahuldab kõiki vajadusi.

Esialgne disain luuakse enne esimese etenduse visuaalse identiteedi välja kujunemist. Esialgset disaini ja disaini seisu novembris võrreldakse joonisel 5.



Joonis 5. Vasakul esialgne publikupoolse rakenduse disain. Paremal publikupoolse rakenduse disain novembris 2022.

Disainides kasutatakse äärest-ääreni komponente ning piisavaid kontraste. Esimese disaini probleemiks oli üleni valge taust, mis teatri hämaramas keskkonnas võib segada. Joonis 6 kujutab esimese lavastuse järgmist vaadet, kus on näha nõrka kontrasti, mis oleks teatris problemaatiline.

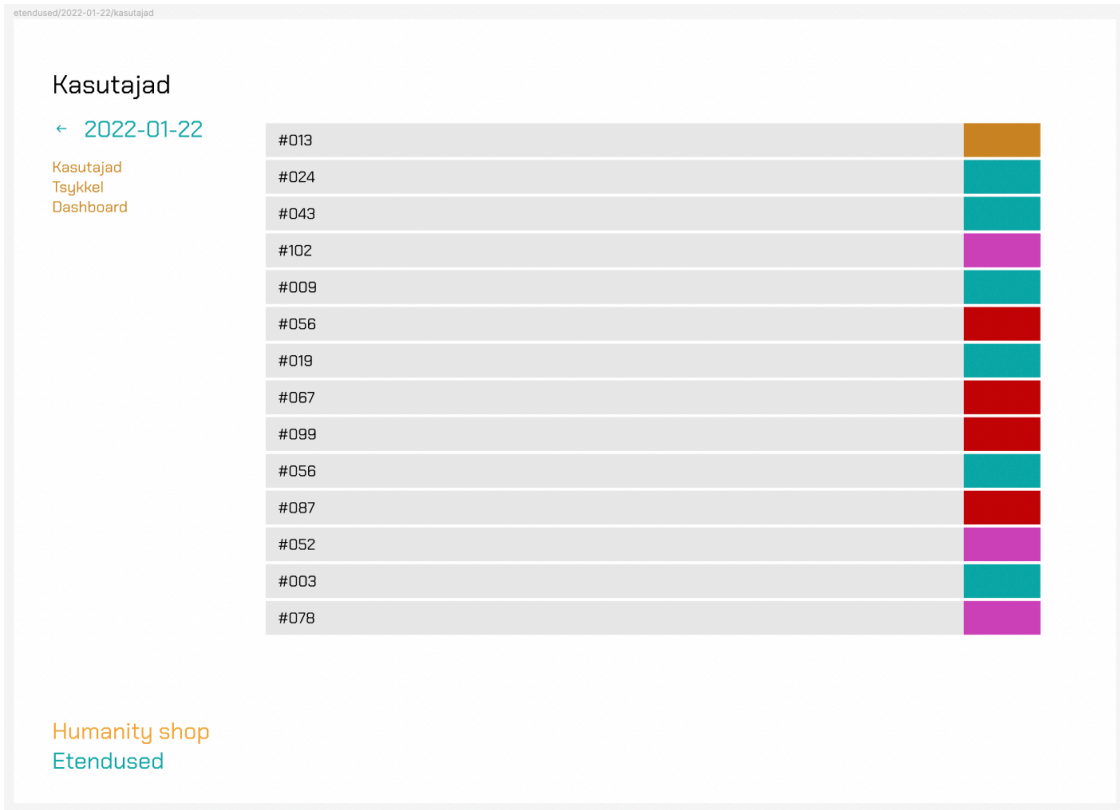


Joonis 6. Halva kontrastiga näide disainist.

Joonistel 7 ja 8 kujutatakse esimese etenduse publikumänge ning ülevaadet saalis hetkel olevatest publiku liikmetest tabelina.



Joonis 7. Esimese etenduse publikumängude algse disaini kuvatõmmised.



Joonis 8. Esialgne disain publikust ülevaate saamiseks.

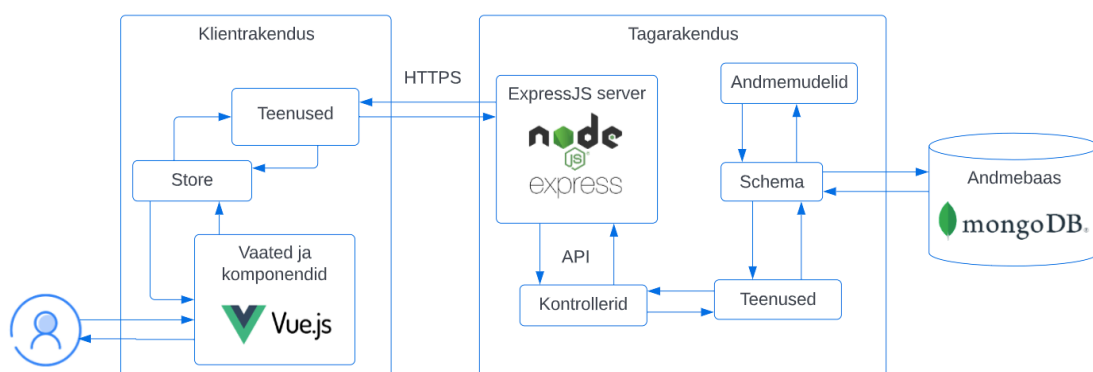
4 Arendus ja testimine

Eelmises peatükis kirjeldati raamistiku arendamiseks kasutatud tehnoloogiaid, nõuete analüüsi ning prototüüpide disainiprotsesse. Käesolev peatükk annab ülevaate rakenduse teostamisest, tekkinud probleemidest ja lahendustest ning edasistest võimalustest.

4.1 Arendus

Arenduseks kasutatakse JetBrainsi tarkvara WebStorm [44], mis on integreeritud arenduskeskkond JavaScriptile. MongoDB andmebaasile ligi pääsemiseks kasutatakse tarkvara Studio 3T [45]. Tagarakenduse arenduseks ja testimiseks kasutatakse API platvormi Postman, mis lihtsustab APIdega suhtlemist [46].

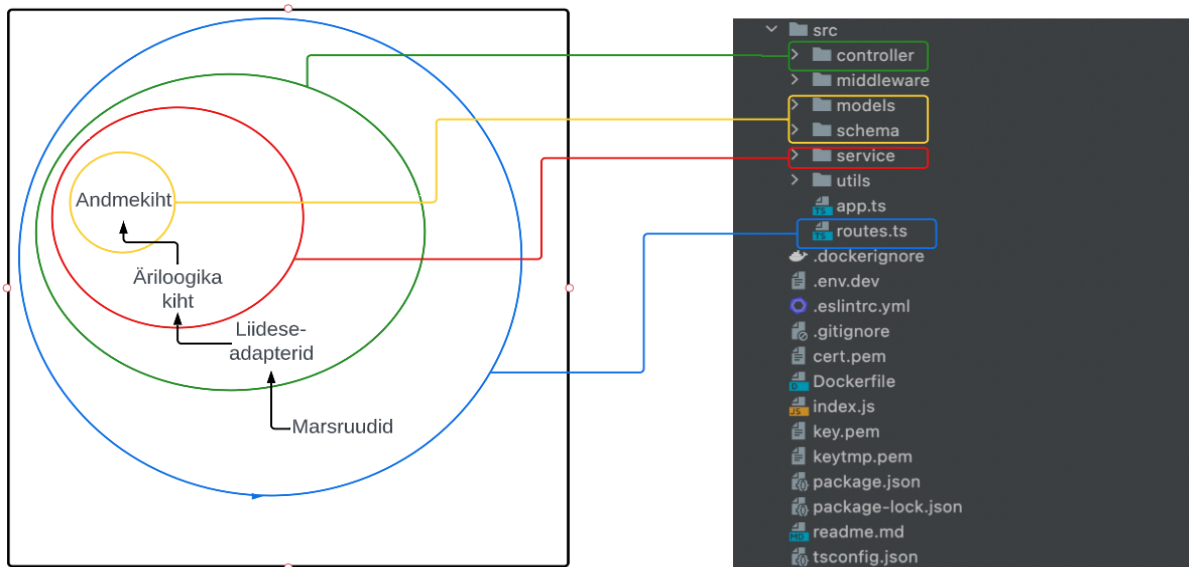
Raamistiku arhitektuuri kujutab joonis 9. Töö realiseerimiseks valitud MEVN pinu järgib traditsioonilist kolmekihilist arhitektuurimustrit, kus kihtideks on kliendipoolne rakendus (VueJS), tagarakendus (ExpressJS ja NodeJS) ning andmebaasikiht (MongoDB).



Joonis 9. Raamistiku arhitektuur.

4.1.1 Taga- ja esirakenduse arendus

Tagarakenduse eesmärk on vältida koodikordusi ning selle jaoks kapseldada rakenduse tuumloogikat võimalikult palju. Seetõttu on tagarakenduse kood jagatud eri osadesse ning osade järgi ka eri kaustadesse. Tagarakenduse arhitektuuri hierarhiat kujutab joonis 10. Joonisel kujutatud välimised ringid sõltuvad sisemistest ringidest, kuid sisemised ringid välimistele ligi ei pääse.



Joonis 10. Tagarakenduse arhitektuur.

Arhitektuuri keskmes on andmekiht, mis vastutab andmebaasi olemite loomise ning andmebaasiga suhtluse eest. Kuna rakenduses kasutatakse Mongoose'i, koosneb andmekiht omakorda kahest osast: olemite defineerimisest (kaustas "models") ja andmebaasiühenduste loomisest (kaustas "schema").

Äriloogika kihis toimub rakenduse põhiline töö – olemite vahelised interaktsioonid. Teenused on omakorda jagatud olemitega seotuse kaudu, näiteks on kokku grupeeritud kasutajate sisselogimisega seotud teenused.

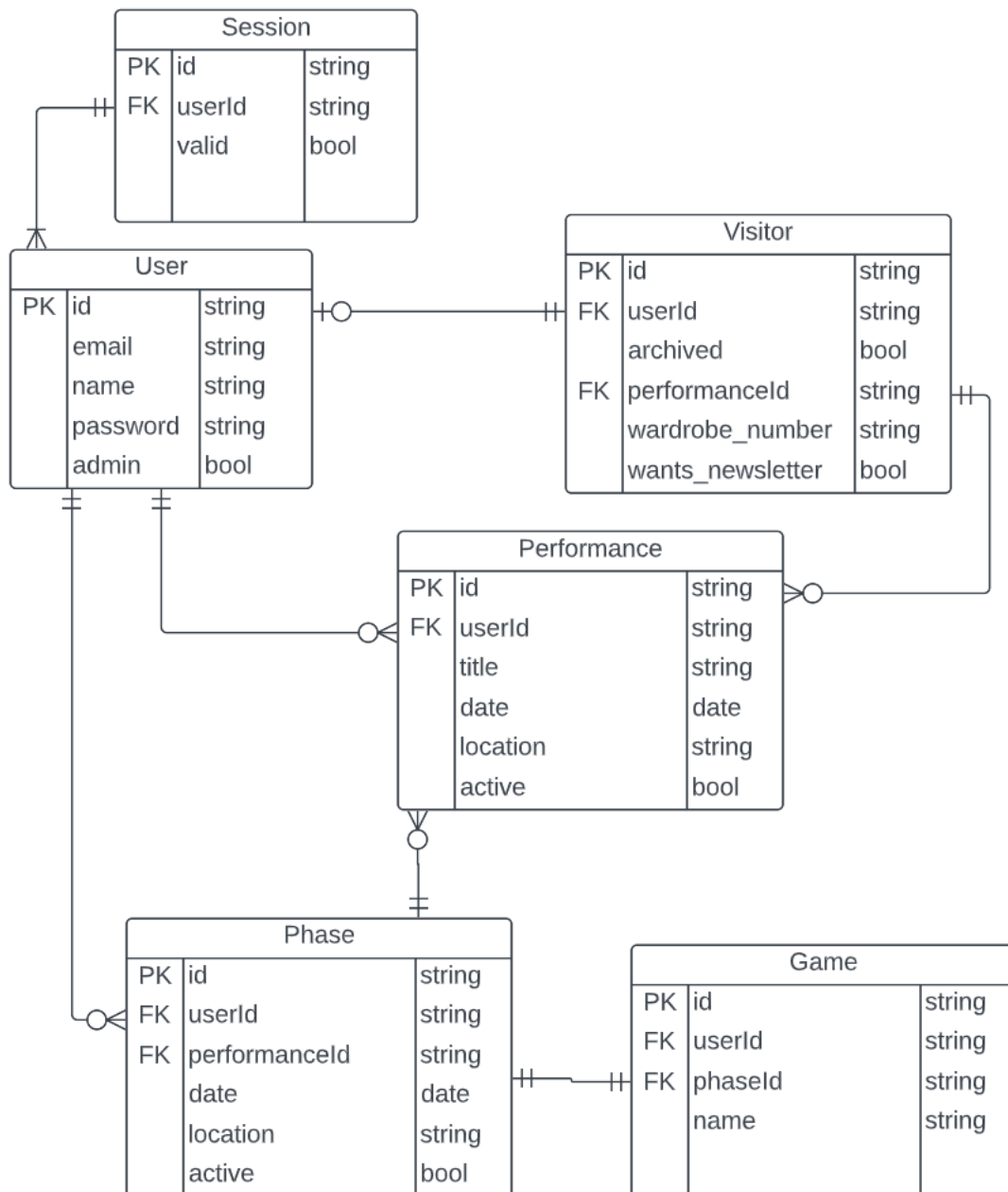
Liideseadapterid (kontrollerid) isoleerivad äriloogika kasutatavatest tööriistadest ning neid kutsuvad omakorda välja marsruudite kogum, millest moodustub REST API (*Representational State Transfer, Application Programming Interface*). REST API kaudu suhtleb esirakendus tagarakendusega läbi HTTP (*Hypertext Transfer Protocol*).

Sarnaselt tagarakendusele jaotub esirakendus kihtideks: vaated, liidesed ja teenused. VueJS rakenduste puhul on rakenduste arhitektuur üsna vaba, kuid tänu laiale kasutajaskonnale ja põhjalikule dokumentatsioonile on juurdunud standardid, millel baseerub ka raamistiku arhitektuur [47]. Vaated vastutavad läbi komponentide esirakenduse visuaalse poole eest ning kuvavad kasutajale infot. Ühtlasi võtavad komponendid vastu informatsiooni ja kutsuvad välja liideseid, mis omakorda kutsuvad välja teenuseid. Teenused ühendavad esirakendust ja tagarakendust läbi HTTP päringute.

4.1.2 Andmebaasi loomine

Prototüüpide ning nõuete analüüsi põhjal luuakse olemi-suhte diagramm keskkonnas Lucidchart. Olemi-suhte diagrammi abil luuakse andmebaas läbi ExpressJS teegi Mongoose [33], mis funktsioneerib andmevahetuskihina MongoDB ja ExpressJSi vahel. Raamistiku olemi-suhte diagrammi kujutatakse joonisel 11.

Raamistiku andmebaas jätab arendajale piisavalt ruumi, et ise oma mängude ja faaside loogika luua, aga katab kasutajate ja sisselogimissessioonide halduse ja mäletamise, etenduse ja selle faaside loomise ning etendusele publiku lisamise nõuded.



Joonis 11. Raamistiku olemi-suhte diagrammi kavand.

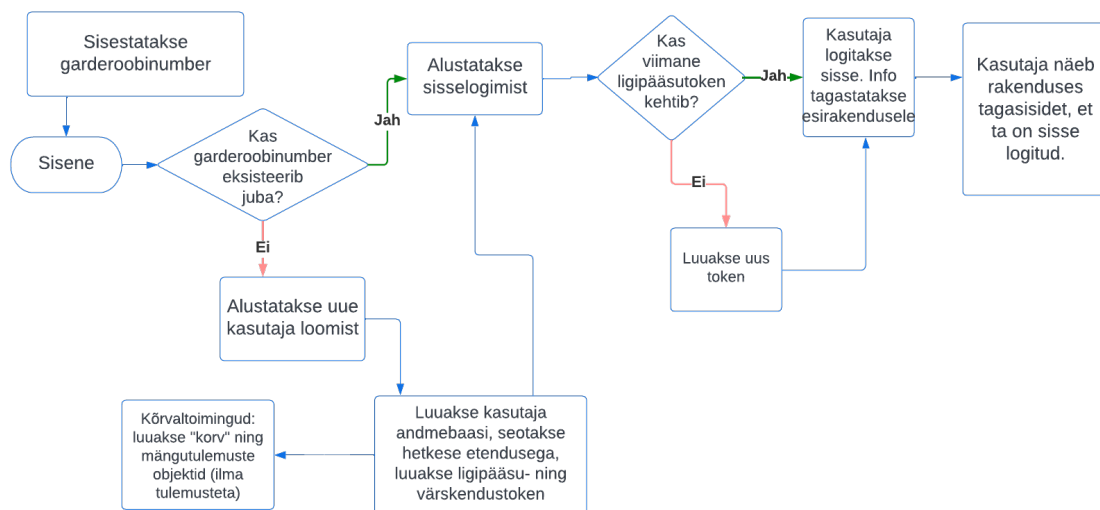
4.1.3 Turvalisus

Raamistik lahendab autentimise ja autoriseerimise JWT (*JSON Web Token*) süsteemiga, kusjuures registreerimise vaade puudub ning loogika on peidetud sisse logimise vaate taha. See otsus võeti vastu, et teatrisse tulles rakendusele ligipääs veelgi kergemaks teha.

JWT koosneb päisest, sisust ning allkirjast. Päises hoitakse infot algoritmi kohta, mille allusel digitaalne allkiri luuakse ning allkiri valideerib JWT tõesuse. Sisu on JWT

funktsionaalne osa, kus hoitakse väiteid kasutaja kohta, näiteks administraatoriväite puhul hoitakse seal väärtust, mille põhjal esirakenduses kas kuvatakse või ei kuvata teatud rakenduse osi.

Joonisel 12 kujutatakse publiku liikme kasutajaga seotud toiminguid. Publiku liikme sisselogimisel kontrollitakse, kas sellise garderoobinumbriga publikuliige juba eksisteerib, ning vastavalt kas luuakse uus konto või leitakse andmebaasist vana. Kasutaja sisse logimisel tagastab REST API kaks JWT tõendit: ligipääsutokeni ning värskendustokeni, mis edaspidi iga REST API kutsungiga tagarakendusse saadetakse. See aitab valideerida, et päringud on pärit usaldusväärsest allikast ning aitab hoida mälus konkreetse kasutaja infot ning õigusi.



Joonis 12. Publiku registreerimise ja sisse logimise süsteempoolne voodiagramm.

Luuakse ka HTTPS tugi lokaalsel arendamisel, et saaks telefonis testida ja ka veebilehitsejas kaamerat kasutada, mis võib mitmetel etendustel vajalikuks osutada.

4.2 Raamistiku testimine

Vaatamata mitmetele tehnilistele probleemidele oli esimene lühike testimisperiod edukas, kuna rakenduse võimekus ning ilmselgemad mittetoimivad osad said ilmsiks

kiiresti. Raamistiku kasutamise analüüs esialgsete seatud nõuete põhjal näitab samuti, et raamistiku kõigi osade arendus on olnud edukas.

Ainsaks probleemiks vajalike nõuete hulgast on kirjutamise hetkel veel puudulik dokumentatsioon, mis ei sisalda endas piisavalt infot taaskasutatavate komponentide kohta. Ajalise ressursi vähesuse tõttu otsustati selle arvelt rohkem tegeleda arendustööga, ning dokumentatsiooni täiendamine jääb edasiarenduste plaanidesse.

Üldtingimused raamistikule võib kokku võtta järgnevalt:

- Raamistik peab olema kergesti ülesseatav ja rakendatav enamustele plaanitavatele etendustele.
- Raamistikul peab olema põhjalik dokumentatsioon ning ülesseadmisjuhend.
- Raamistiku visuaalsed elemendid peavad olema lihtsa vaevaga seadistatavad, et kohanduda iga erineva etenduse disainiga.
- Raamistik peab võimaldama etenduse autorite ja publiku sisselogimist, etenduse instantside ja iga etenduse faaside haldust.
- Publikule suunatud osa peab olema eeskätt kasutatav mobiilseadetes.

MoSCoW järgi kindlasti vajalike nõuete täitmise tulemused on tabelis kujutatud tabelis 5, väiksema prioriteediga nõuete täitmise tulemused tabelis 6.

Tabel 5. Kindlasti vajalike nõuete täitmise staatused.

Nõue (üldine)	Staatus	Märkus
raamistik peab kasutama LTS versioone kõigist kasutusel olevatest keeltest ja tehnoloogiatest	valmis	
peab eksisteerima dokumentatsioon ja ülesseadmisjuhend kõigile PlayOni liikmetele	valmimisel	edasiarendusele
peavad olema olemas administraatoritele mõeldud liidesed konkreetsete toimingute haldamiseks ja vajalike andmete või konfiguratsioonide muutmiseks/värskendamiseks iga etenduse instantsi jaoks	valmis	
peab olema olemas turvaline andmelahendus tundlike andmete salvestamiseks	valmis	
peab olema olemas garderoobinumbritel ja valikulisel meiliväljal põhinev sisselogimissüsteem	valmis	
peab olema olemas etenduste haldussüsteem, millel on faasid esituse erinevate osade jaoks, et võimaldada mängu ajal paindlikkust	valmis	
peab olema olemas peamine API ja võimalus hõlpsasti mooduleid lisada uute poodide malli abil, et hõlbustada seadistamist	valmis	malli juhend puudu
peavad olema olemas lihtsasti seadistatavad keskkonnad ja visuaalsete elementide põhitunnused	valmis	
administraatorid peavad saama sisse logida emaili ja parooliga	valmis	
administraatorid peavad nägema publiku liikmetest ülevaadet	valmis	
administraatorid peavad saama käivitada ja peatada etenduse faase	valmis	
administraatorid peavad saama lisada etenduse instantsi (st kuupäevi, mil etendus toimub)	valmis	
Visuaalsed ja kasutajakogemusega seotud lisanõuded	Staatus	Märkus
peab olema piisav kontrast ja tekstisuurused	valmis	
peab olema igale ekraanisuurusele kohanduv disain	valmis	

rakenduse osad peavad töötama kiiresti	valmis	
peab eksisteerima selge <i>microcopy</i>	valmis	
peab toimuma taktiline tagasiside	valmis	
peab olema minimeeritud kasutajate eksimisvõimalus	valmis	

Tabel 6. Teisejärguliste nõuete täitmise staatused.

Peaks olema (<i>should have</i>)	staatus	märkus
arendusserveri püstipanek peab olema kerge ja kiire	valmis	
disain peab olema lihtsalt konfigureeritav (S)CSS-i muutujate kaudu, et see sobiks iga etenduse stiiliga	valmis	
olemas olema peavad mallid erinevate rakenduse osade kiireks seadistamiseks	valmis	
olemas peab olema konfigureeritud HTTPS-baasil kohaliku arenduse võimalus	valmis	
etendust alustades peab rakendus automaatselt looma uued mängude instantsid koos vajalike andmetega	valmis	
esimene faas peab käivituma automaatselt	valmis	
etenduse lõpetamine peab arhiveerima publiku ja nendega seonduva	valmis	
Võiks olla (<i>could have</i>)	staatus	märkus
peab olema lahendatud tõlgete lisamise süsteem	edasiarendusele	
publiku liikmed peavad saama tagantjärele oma emaili lisada	edasiarendusele	
administraatorid peavad saada publikule jooksvalt teavitusi saata	ei teostata	

5 Hinnang loodule ja võimalused edasi arenduseks

Eelmise peatüki lõpus vaadeldi raamistiku nõudmistele vastamise tulemusi. Kindlasti nõutud funktsionaalsustest rakendati 17/18 ehk 94%. Staatusega *should have* nõuete hulgast rakendati 100%, seega võib öelda, et raamistiku eesmärgid saavutati ning raamistik on kasutamiskõlbulik järgmise projekti jaoks. Dokumentatsiooni olulisust vähendati, kuna seda on päriselt vaja alles pilootprojekti õnnestumise järgselt.

Staatusega *could have* nõuded on planeeritud tulevikuarendusteks ning lisaks suurendaks veakindlust ja tõstaks töö kvaliteeti põhjalik automaattestide kogum. Järgnevalt on loetletud mõned võimalused edasiarendusteks.

5.1 Keeletoe lisamine

Algsesse skooopi planeeritud keeletoe lisamine liikus prioriteedilt madalamaks, sest tegemist ei ole hädavajaliku funktsionaalsusega esimese projekti jaoks. Küll aga on rahvusvaheliste projektide puhul tõlketoe lisamine hädavajalik ning üks saavutatavamaid plaane.

5.2 Jooksvad teavitused

Teavituste saatmine oli algses skooobis, kuid mittehädavajalik funktsionaalsus, seega eemaldati skoobist. Teatud lavastuste puhul võib kasulikuks osutada rakenduse võimekus saata publiku seadmetele teavitusi ning see on üks kõige suuremat väärtust pakkuvam edasiarenduse võimalus.

5.3 Etenduse kokkuvõtete saatmise süsteem

Esialgsete mõtete hulgas oli ka rakendusesisene süsteem, mis lubaks eeldefineeritud malli järgi etenduse lõppedes kõigile külastajatele saata emaili etenduse kokkuvõtte, publiku

panuse ja muu olulisega. Selline funktsionaalsus nõuaks aga suuremat lahendust, kui lahenduse skooopi mahub, ning prioriteeti samuti vähendati.

5.4 Automaattestid

Kogu lahenduse testimine on siiani toimunud käsitsi, kuid raamistiku põhiosade töökindluse tagamiseks oleks võimalik kirjutada automaattestid, mis aitaks paremini rakendust hallata ning ootamatuid probleeme ennetada.

6 Kokkuvõte

Käesoleva bakalaureusetöö käigus loodi raamistik interaktiivse teatri rakenduste arendamiseks. Rakenduse loomise ajendiks oli VAT Teatri vajadus leida interaktiivsetele produktsioonidele tehnoloogiline lahendus. Lisaks uuriti teisi interaktiivse teatri tehnoloogiaid ja nende kasutamise keerukust.

Tehnilise analüüsi tulemina arendati raamistik MEVN pinu (MongoDB, ExpressJS, VueJS, NodeJS) kasutades. Raamistik koosneb esi- ja tagarakenduse baasprojektist, andmebaasi seadistusest, konteineriseerimislahendusest ja dokumentatsioonist.

Valminud raamistik võimaldab kiiresti üles seada esmased arenduskeskkonnad, kõigi rakenduse osade konteineriseerimist ning serverisse püstitamise seadistust. Lisaks on kaetud olulisemad funktsionaalsused nagu kasutajate loomine ja sisselogimine ning etenduste käivitamine ja peatamine. Lisaks käsitleti bakalaureusetöö raames teatrikülastaja kasutajakogemusega seotud teemasid, mille tulemusena arendati publikupoolne esirakenduse osa külastajate ja keskkonna eripärasid arvestades.

Bakalaureusetöö võib lugeda õnnestunuks, sest raamistik täidab enamusi analüüsis kirjeldatud nõudeid ning esimene raamistiku kasutamiskord rakenduse loomiseks oli edukas. Loodud raamistikku saab edasi arendada täitmata nõuete ja eelmises peatükis välja toodud ideede baasil, mis on töös välja toodud.

Kasutatud kirjandus

- [1] P. N. S. w. I. Technologies, „Theatres,“ [Võrgumaterjal]. Available: <https://play-on.eu/theatres/>. [Kasutatud 28 november 2022].
- [2] A. B. Consortium, „Chapter 10: MoSCoW Prioritisation,“ [Võrgumaterjal]. Available: <https://www.agilebusiness.org/dsdm-project-framework/moscow-prioritisation.html>. [Kasutatud 30 november 2022].
- [3] U. o. Kansas, „Developing a Framework or Model of Change,“ [Võrgumaterjal]. Available: <https://ctb.ku.edu/en/4-developing-framework-or-model-change>. [Kasutatud 30 oktoober 2022].
- [4] A. B. Consortium, „Chapter 11: Iterative Development,“ [Võrgumaterjal]. Available: <https://www.agilebusiness.org/dsdm-project-framework/iterative-development.html>. [Kasutatud 29 november 2022].
- [5] C. Institute, „What is a Framework? Why We Use Software Frameworks,“ [Võrgumaterjal]. Available: <https://codeinstitute.net/global/blog/what-is-a-framework/>. [Kasutatud 30 november 2022].
- [6] O. Didenko, „How Long Does It Take to Develop a Web App: Answers to All Questions,“ 30 august 2022. [Võrgumaterjal]. Available: <https://www.altamira.ai/blog/how-long-does-it-take-to-develop-a-web-app/>. [Kasutatud 30 november 2022].
- [7] T. C. I. f. IT, „What makes software dependable?,“ 02 mai 2007. [Võrgumaterjal]. Available: <https://www.bcs.org/articles-opinion-and-research/what-makes-software-dependable/>. [Kasutatud 28 november 2022].
- [8] Vue, „VueJS,“ [Võrgumaterjal]. Available: <https://vuejs.org/>. [Kasutatud 30 oktoober 2022].
- [9] ExpressJS, „Fast, unopinionated, minimalist web framework for Node.js,“ [Võrgumaterjal]. Available: <https://expressjs.com/>. [Kasutatud 30 oktoober 2022].
- [10] MongoDB, Inc, „MongoDB Software Lifecycle Schedules,“ [Võrgumaterjal]. Available: <https://www.mongodb.com/support-policy/lifecycles>. [Kasutatud 03 jaanuar 2022].
- [11] R. Anderson, „Playing at Plays. What exactly is immersive, interactive, participatory or playing theatre, anyway?,“ 5 veebruar 2015. [Võrgumaterjal]. Available:

- <https://playingatplays.wordpress.com/2015/02/05/what-exactly-is-immersive-interactive-participatory-or-playing-theatre-anyway/>. [Kasutatud 28 november 2022].
- [12] C. R. Teresa Cerratto Pargman, „Understanding Audience Participation in an Interactive Theatre Performance,“ oktoober 2014. [Võrgumaterjal]. Available: https://www.researchgate.net/publication/264275142_Understanding_Audience_Participation_in_an_Interactive_Theater_Performance. [Kasutatud 28 november 2022].
- [13] Kahoot!, „Kahoot! Learning Games,“ [Võrgumaterjal]. Available: <https://kahoot.it/>. [Kasutatud 7 detsember 2022].
- [14] T. Ludowy, „1984: Ministry of Love,“ [Võrgumaterjal]. Available: <https://playon.eu/ludowy-immersive/>. [Kasutatud 29 november 2022].
- [15] Vene Teater, „TULEB / EI TULE. EESTI 100 AASTA PÄRAST,“ 2018. [Võrgumaterjal]. Available: <https://veneteater.ee/et/show/budetne-budet-estoniia-chez-100-let.html>. [Kasutatud 05 jaanuar 2023].
- [16] OÜ Kultuurineta, „Otsuse anatoomia,“ 2015. [Võrgumaterjal]. Available: <https://www.kultuurikava.ee/event/otsuse-anatoomia-0>. [Kasutatud 05 jaanuar 2023].
- [17] D. C. Labs, „The Cuttin Room Presents: What Is Love?,“ [Võrgumaterjal]. Available: <https://yorkmediale.com/projects/the-cutting-room/>. [Kasutatud 29 november 2022].
- [18] Oomipood, „Oomipoe e-pood,“ [Võrgumaterjal]. Available: https://www.oomipood.ee/en/category/pir_liikumisandurid. [Kasutatud 29 november 2022].
- [19] BlackTrax, „Showcase,“ [Võrgumaterjal]. Available: <https://blacktrax.castsoft.com/showcase/>. [Kasutatud 29 november 2022].
- [20] P. H. c. David Gochfeld, „Presentation on XR technologies and recent projects,“ mai 2022. [Võrgumaterjal]. Available: <https://www.youtube.com/watch?v=AFFYQc8Yp9c>. [Kasutatud 29 november 2022].
- [21] Bluetooth, „Location Services | Bluetooth Technology Website,“ [Võrgumaterjal]. Available: <https://www.bluetooth.com/learn-about-bluetooth/solutions/location-services/>. [Kasutatud 29 november 2022].
- [22] AB&R, „What is RFID and How Does RFID Work?,“ [Võrgumaterjal]. Available: <https://www.abr.com/what-is-rfid-how-does-rfid-work/>. [Kasutatud 29 november 2022].
- [23] N. Forum, „What NFC does,“ [Võrgumaterjal]. Available: <https://nfc-forum.org/learn/what-nfc-does/>. [Kasutatud 29 november 2022].

- [24] QRcode.com, „QR Code Standardization,“ [Võrgumaterjal]. Available: <https://www.qrcode.com/en/about/standards.html>. [Kasutatud 29 november 2022].
- [25] Oomipood, „Oomipoe e-pood, RFID kategooria,“ [Võrgumaterjal]. Available: https://www.oomipood.ee/category/rfid_rf_1_wire_moodulid. [Kasutatud 29 november 2022].
- [26] N. Pood, „NFC komplektid,“ [Võrgumaterjal]. Available: <https://www.nfcpood.ee/6-nfc-komplektid>. [Kasutatud 29 november 2022].
- [27] S. Lyonnais, „Accessibility Design 101: Color Contrast Considerations for UX Designers,“ 22 aprill 2020. [Võrgumaterjal]. Available: <https://xd.adobe.com/ideas/principles/web-design/color-contrast-considerations-accessibility-design/>. [Kasutatud 29 november 2022].
- [28] U. Pickle, „What is the smallest screen size you should design for?,“ 23 detsember 2021. [Võrgumaterjal]. Available: <https://uxpickle.com/what-is-the-smallest-screen-size-you-should-design-for/>. [Kasutatud 29 november 2022].
- [29] B. Chung, „Designing for the appearance of speed,“ 23 märts 2017. [Võrgumaterjal]. Available: <https://juneuprising.medium.com/designing-for-the-appearance-of-speed-aaabc7f568c2>. [Kasutatud 29 november 2022].
- [30] Design Minds, „Disainmõtlemisest,“ 2022. [Võrgumaterjal]. Available: <https://designminds.ee/disainmotlemisest>. [Kasutatud 05 jaanuar 2023].
- [31] Interaction Design Foundation, „Design Thinking,“ [Võrgumaterjal]. Available: <https://www.interaction-design.org/literature/topics/design-thinking>. [Kasutatud 29 november 2022].
- [32] N. Raval, „Top 10 JavaScript Usage Statistics to Watch Out for in 2022,“ 20 september 2022. [Võrgumaterjal]. Available: <https://radixweb.com/blog/top-javascript-usage-statistics>. [Kasutatud 30 november 2022].
- [33] Mongoose, „Elegant MongoDB object modeling for NodeJS,“ [Võrgumaterjal]. Available: <https://mongoosejs.com/>. [Kasutatud 29 november 2022].
- [34] BuiltWith, „Websites using Vue including Historical,“ [Võrgumaterjal]. Available: <https://trends.builtwith.com/websitelist/Vue/Historical>. [Kasutatud 30 november 2022].
- [35] E. Answers, „What is MEVN stack?,“ [Võrgumaterjal]. Available: <https://www.educative.io/answers/what-is-mevn-stack>. [Kasutatud 30 november 2022].
- [36] OpenJS Foundation, „About NodeJS,“ [Võrgumaterjal]. Available: <https://nodejs.org/en/about/>. [Kasutatud 30 november 2022].

- [37] npm Inc., „npm,“ [Võrgumaterjal]. Available: <https://www.npmjs.com/>. [Kasutatud 30 november 2022].
- [38] E. You, „Vite - Next Generation Frontend Tooling,“ 2019. [Võrgumaterjal]. Available: <https://vitejs.dev/>. [Kasutatud 30 november 2022].
- [39] M. Donath, „Material for MkDocs,“ [Võrgumaterjal]. Available: <https://squidfunk.github.io/mkdocs-material/>. [Kasutatud 30 november 2022].
- [40] Google LLC, „Material Design,“ [Võrgumaterjal]. Available: <https://m3.material.io/>. [Kasutatud 30 november 2022].
- [41] Git, „Git,“ [Võrgumaterjal]. Available: <https://git-scm.com/>. [Kasutatud 30 november 2022].
- [42] J. Miller, „Which Version Control System Should Your Developers Use?,“ 10 juuni 2021. [Võrgumaterjal]. Available: <https://www.bairesdev.com/blog/which-version-control-system-developers-use/>. [Kasutatud 30 november 2022].
- [43] GitHub, Inc., „GitHub: Let's build from here,“ [Võrgumaterjal]. Available: <https://github.com/>. [Kasutatud 30 november 2022].
- [44] JetBrains s.r.o, „WebStorm - The smartest JavaScript IDE,“ [Võrgumaterjal]. Available: <https://www.jetbrains.com/webstorm/>. [Kasutatud 30 november 2022].
- [45] 3T Software Labs Ltd., „Studio 3T, the professional GUI for MongoDB,“ [Võrgumaterjal]. Available: <https://studio3t.com/>. [Kasutatud 30 november 2022].
- [46] Postman, Inc., „Postman API Platform,“ [Võrgumaterjal]. Available: <https://www.postman.com/>. [Kasutatud 30 november 2022].
- [47] VueSchool, „How to Structure a Large Scale VueJS Application,“ 6 juuli 2021. [Võrgumaterjal]. Available: <https://vueschool.io/articles/vuejs-tutorials/how-to-structure-a-large-scale-vue-js-application/>. [Kasutatud 3 jaanuar 2023].

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks¹

Mina, Helena Väinmaa

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose "Raamistik interaktiivse teatri rakenduste arendamiseks", mille juhendaja on Kristiina Hakk, PhD
 - 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.

03.01.2023

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.