

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Vladislav Lahtarin 175003IDDR

Konverentside otseülekande eessüsteemi arendus ettevõtte Videolevels näitel

Diplomitöö

Juhendaja: Joel Kivi

Bakalaurusekraad

Tallinn 2021

Autorideklaratsioon

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

Autor: Vladislav Lahtarin

15.05.2021

Annotatsioon

Käesoleva diplomitöö eesmärk on arendada Elisa ettevõtte poolt tellitud eessüsteem konverentside otseülekannete vaatamiseks. Eessüsteemi loomise eesmärgiks on võimaldada vaatajatel suhelda ning koos toimetada esinejatega ja teiste konverentsil osalejatega.

Analüüsi käigus autor kirjeldab rakenduse funktsionaalseid ja mittefunktsionaalseid nõudeid, analüüsib kliendilt saadud kasutajaliidest, võrdleb võimalikke tehnoloogiaid eesmärkide saavutamiseks ja põhjendab nende valikut.

Töö praktilises osas kirjeldab autor kogu rakenduse arhitektuuri, eessüsteemi struktuuri, andmetüüpide struktuuri ja vaadete arendamist. Seejärel tuuakse välja rakenduse edasiarendamise võimalused.

Diplomitöö tulemuseks on rakendus, mille kaudu Elisa ettevõtte on juba edukalt otseülekandes konverentsi läbi viinud.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 25 leheküljel, 5 peatükki, 16 joonist.

Abstract

Development of the Conference Live Streaming Front-end on the Example of Videolevels

The goal of the thesis is to develop a front-end system for watching live broadcasts of conferences ordered by Elisa. The purpose of creating this front-end is to allow viewers to interact with speakers and other conference participants.

In the course of the analysis, the author names the functional and non-functional requirements of the application, analyzes the user interface received from the customer, compares possible technologies to achieve the goals and justifies their choice.

In the practical part of the work, the author describes the architecture of the whole application, the structure of the front-end system, the structure of data types and the development of views. The author then mentions the possibilities for further development of the application.

The result of the diploma thesis is an application through which Elisa company has already successfully conducted a live conference.

The thesis is in Estonian and contains 25 pages of text, 5 chapters, 16 figures.

Lühendite ja mõistete sõnastik

AJAX	<i>Asynchronous JavaScript and XML</i>
Amazon S3	<i>Amazon Simple Storage Service</i> , Salvestusruum
Ant Design	React'i kasutajaliidese teek
CLI	<i>Command line interface</i> , käsurea liides
CSS	<i>Cascading Style Sheets</i> , stiililehe keel, mis kirjutatud veebilehe kujundamiseks ja esitamiseks
DOM	<i>Document Object Model</i> , dokumendi objektimudel
DRY	<i>Don't repeat yourself</i> , ära korda iseennast
Figma	Kasutajaliidese ja kasutajakogemuse prototüüpimise rakendus
getServerSideProps	Next.js'i funktsioon
HTML	<i>Hypertext Markup Language</i> , hüperteksti märkimise keel
HTTPS	<i>HyperText Transfer Protocol Secure</i> , turvaline hüperteksti edastusprotokoll
ID	Identifikaator
IP	Internetiprotokoll
JavaScript	Objektorienteeritud programmeerimiskeel
JSX	<i>JavaScript XML</i> , JavaScripti süntaksi laiendus
JWT	<i>JSON Web Token</i> , autentimise lubade standard
<i>localhost</i>	Kohalik host
MVC	<i>Model-View-Controller</i> , Mudel-Vaade-Kontroller – rakendustes kasutatav arhitektuur
SEO	<i>Search engine optimization</i> , otsingumootoritele optimeerimine
SPA	<i>Single Page Application</i> , üheleherakendus
SSR	<i>Server-side rendering</i> , serveripoolne renderdamine
Styled-components	Stiiliteek React komponentide kujundamiseks
SVG	<i>Scalable Vector Graphics</i> , skaleeruv vektorgraafika
Timeline	Stiiliteegi Ant Design'i komponent
Transpiler	Tööriist, mis loeb ühes programmeerimiskeeles kirjutatud lähtekoodi ja toodab samaväärse koodi teises keeles, millel on sarnane abstraktsioonitase

Sisukord

1 Sissejuhatus	8
1.1 Metoodika.....	8
1.2 Aktuaalsus	9
2 Analüüs.....	10
2.1 Eessüsteemi nõuded.....	10
2.2 Raamistiku valik.....	11
2.3 Kasutatud tööriistad ja tehnoloogiad	12
2.3.1 Visual Studio Code.....	12
2.3.2 Next.js.....	13
2.3.3 TypeScript	13
2.3.4 Ant Design.....	14
2.3.5 Styled-components	14
2.4 Kasutajaliidese disain ja vaated.....	14
3 Valminud eessüsteemi kirjeldus	17
3.1 Rakenduse arhitektuur	17
3.2 Eessüsteemi struktuur	18
3.3 Andmetüüpide struktuur	20
3.4 Autoriseerimata kasutaja vaade.....	21
3.4.1 Autoriseerimine	21
3.4.2 Konverentsi kava	23
3.4.3 Esinejate ja sponsorite loetelud	24
3.5 Autoriseeritud kasutaja vaated.....	25
3.5.1 Videomängija	25
3.5.2 Külgriba menüü	26
3.5.3 Sponsorite detailvaade.....	29
4 Edasiarendamise võimalused.....	31
5 Kokkuvõte	32
Kasutatud kirjandus	33
Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks	35

Jooniste loetelu

Joonis 1. Rakenduse arhitektuur.....	18
Joonis 2. Eessüsteemi rakenduse struktuur.	19
Joonis 3. Konverentsi andmeobjekti tüüpide struktuur.	20
Joonis 4. Osalejate andmeobjekti tüüpide struktuur.....	21
Joonis 5. Sisselogimis vaade.	22
Joonis 6. Veateade vale pileti koodiga autoriseerimisel nutitelefoni- st.....	22
Joonis 7. Konverentsi kava autoriseerimata kasutaja vaates.	23
Joonis 8. Esinejate info modaalses aknas.	24
Joonis 9. Esinejate loetelu.	25
Joonis 10. Sponsorite loetelu.	25
Joonis 11. Videomängija vaade koos avatud külgriba menüüga.....	26
Joonis 12. Videomängija vaade koos suletud külgriba menüüga.....	27
Joonis 13. Külgriba menüüs olev konverentsi osalejate aken.	28
Joonis 14. Külgriba menüüs olev küsitlus nutitelefoni- st.....	28
Joonis 15. Külgriba menüüs olev küsimuste jaoks jututuba.....	29
Joonis 16. Sponsorid detailvaade.	30

1 Sissejuhatus

Igal aastal tekivad uued ideed, kuidas mitmekesistada konverentside traditsioonilist korraldust. See on suurepärane võimalus teha konverents meeldejäävaks mitte ainult esinejatega, vaid ka pakutavate võimalustega.

Käesoleva lõputöö eesmärk on luua konverentside otseülekannete eessüsteem, mis oleks mugav kasutada nii arvuti kui ka nutitelefoniga. Rakendus arendatakse Elisa ettevõtte jaoks ning peab võimaldama konverentse ja muid üritusi jälgida otse eetris.

Probleemiks osutus kasvav vajadus veebikonverentside korraldamiseks ja nende edastamiseks. Antud probleemi lahendamiseks arendatakse eessüsteemi, mis vastab kliendi vajadustele. Rakenduse kaudu saab jälgida üritust otseülekandes erinevates keeltes, osaleda küsitlustes ja suhelda esinejaga ning teiste vaatajatega.

1.1 Metoodika

Antud lõputöö analüüsi osa alguses selgitatakse välja arendatava eessüsteemi funktsionaalsed ja mittefunktsionaalsed nõuded. Seejärel analüüsitakse võimalikke raamistikke ja tehnoloogiaid lahenduse elluviimiseks ning põhjendatakse tehtud valikuid. Enne arenduse alustamist analüüsitakse kasutajaliidese disain Figma keskkonnas, mis oli kliendi poolt etteantud. See aitab ehitada eessüsteemi stiili teemat, rakenduse andmeobjekti ning planeerida komponendi põhiseaduse läbi.

Töö praktilise osa alustatakse rakenduse arhitektuuri kirjeldamisega, mis annab ülevaate kuidas andmed liiguvad väljaspool eessüsteemi ja kuidas muud eessüsteemiga seotud teenused omavahel suhtlevad. Tehakse ülevaade eessüsteemi ja andmetüüpide struktuurist. Seejärel kirjeldatakse valminud eessüsteemi. Lõputöö viimases osas tuuakse välja eessüsteemi edasiarendamise võimalused.

1.2 Aktuaalsus

Lõputöö teema on aktuaalne, eriti arvestades kuivõrd on inimeste igapäevane elu muutunud. Pandeemiast tulenevate piirangute tõttu ei saa konverentsi korraldajad üritust mitu kuud ette planeerida, kuna pole võimalik ette teada, millal ja kui kauaks piiranguid kehtestatakse. Seetõttu on kasvanud vajadus ürituste otseülekannete järele.

Pandeemia piirangute ja ohtude vähendamise korral ei kao konverentside otseülekannete tähtsus, kuna see võimaldab üritusel füüsiliselt mitte viibida, kuid sellegipoolest selles osaleda. Kui potentsiaalne konverentsi osaleja elab ürituse läbiviimise asukohast kaugel ja soovib kuulata konkreetset esinejat, siis pole tal vaja palju aega kulutada kohale jõudmiseks.

Käesoleva lõputöö rakendus annab esinejatele võimaluse suhelda ja koos toimetada vaatajatega vaatamata sellele, kas publik on saalis või vaatab otseülekannet. Samuti ürituse edastamine erinevates keeltes võimaldab meelitada laiema publiku tähelepanu.

2 Analüüs

Arenduse lihtsustamiseks püstitas autor eesmärgi analüüsida võimalikke kasutatavaid tehnoloogiaid vastavalt kliendi poolt seatud rakenduse nõuetele. Kuna tegemist on arenduse esimese etapiga, lihtsustab metoodika ja tehnoloogiate õige valik tulevikus rakenduse laiendamist ja edasiarendamist. Õigete tehnoloogiate valimine on iga rakenduse algetapi arendamises oluline osa, kuna nende muutmine võib tulevikus olla väga kulukas.

2.1 Eessüsteemi nõuded

Eessüsteemi nõuete kogumiseks toimus pidev koostöö kliendiga. Lisaks algsetele püstitatud eessüsteemi nõuetele täiendati arendusprotsessi käigus nõuete loetelu kliendi soovi muutumisel või mitmesuguste piirangute ilmnemisel. Järgnevalt on väljatoodud kliendi poolt seatud funktsionaalsete ja mittefunktsionaalsete nõuete loetelu.

Eessüsteemi funktsionaalsed nõuded:

- Autoriseerimine pileti koodiga.
- Autoriseerimata kasutajale on näha ainult informatiivne lehekülg koos esinejatega, sponsoritega ja ürituse kavaga.
- Autoriseeritud kasutajale on kättesaadav otseülekande videomängija koos kohandatud funktsionaalsusega.
- Konverentsi otseülekanne ei tohiks videomängijalt sponsori detailvaate lehele üleminekul katkeda.
- Sponsori detailvaates on võimalik konkreetse sponsoriga alustada privaatset vestlust.
- Esinejal on võimalik suhelda ja koos toimetada kõigi pealtvaatajatega korraga.
- Küsitluste läbiviimise ja nende statistika kogumise võimalus.
- Võimalus jälgida konverentsi erinevates keeltes.
- Konverentsi progress ja esineja on jälgitav kava järgi.

Eessüsteemi mittefunktsionaalsed nõuded:

- Arendus peab olema teostatud inglise keeles.
- Mugav kasutada arvuti kui ka nutitelefoniga.
- Võimaldada kodulehe optimeerimine otsingumootorite jaoks.
- Lihte edasi arendada.
- Kasutajaliides on ehitatud vastavalt kliendi poolt esitatud prototüübi järgi.
- Peab töötama järgnevates veebilehitsetes:
 - Google Chrome
 - Mozilla Firefox
 - Safari
 - Opera

2.2 Raamistiku valik

Lähtudes 2020. aasta lõpu JavaScript'i raamistike statistikast on React enim kasutatav eessüsteemide tehnoloogia [1]. Vaatamata sellele, et React'i võrreldakse alati teiste raamistikega, on see teek siiski väga korraliku dokumentatsiooniga. Arvestades, et React'il on suur kogukond ja populaarsus, on tal ka rohkem materjale harjutamiseks ja õppimiseks. Suure koormusega eessüsteemi laadimine on tänu React'i virtuaalse DOM'i juurutusele ja erinevatele renderdamise optimeerimistele väga kiire, mis teeb kasutajakogemust paremaks. Selle asemel, et otse DOM'iga suhelda, tehakse sellest koopia – virtuaalne DOM. Kõik muudatused tehakse koopias, seejärel võrreldakse erinevusi tegeliku DOM puuga ja joonistatakse ümber ainult need kohad, kus erinevused on leitud [2]. Samuti rakendab React funktsionaalse programmeerimise kontseptsioone, luues koodi, mida on lihtne testida ja uuesti kasutada [3]. Suure valiku kergesti kättesaadavate, avatud lähtekoodiga pistikprogrammide ja laienduste abil saab välja töötada peaaegu igat tüüpi eessüsteeme ja veebilahendusi. Komponendimallid on üles ehitatud JSX abil, mis võib mõne arendaja jaoks segadust tekitada, kui nad esimest korda React'iga tutvuvad. React'i on võimalik kasutada koos TypeScript'iga kuna, TypeScript toetab JSX süntaksi.

Angular on kolme tehnoloogia seast kõige nõudlikuma olemusega raamistik. Üksikasjalik dokumentatsioon võimaldab arendajal saada kogu vajaliku informatsiooni raamistiku õppimisel. Loetletud kolme raamistiku seast pakub Angular parimat CLI [4]. Angular kasutab kahepoolset andmete sidumist, mis on sarnane MVC arhitektuuriga, kus mudel

ja vaade on sünkroniseeritud [5]. Raamistiku struktuur ja arhitektuur on spetsiaalselt kohandatud suurte projektide jaoks, mistõttu Angular'i õppimine võtab tavaliselt rohkem aega võrreldes React'iga ja Vue'ga. Angular on mitmekülgne raamistik, millel on palju võimalusi ühe probleemi lahendamiseks. Sellel on keeruline komponentide haldamine, mis nõuab palju korduvaid toiminguid [5]. Vaatamata 2020. aasta JavaScript'i raamistike andmetele, mis näitavad, et Angular'i kasutajate rahulolu on langenud, jääb antud raamistik populaarsuselt teisel kohal [1].

Vue on loetletud tehnoloogiatest kõige uuem kasvava kasutajaskonnaga raamistik. See on algajate jaoks populaarne valik, sest kui arendajal pole varasemat kogemust sarnaste raamistikega, siis võrreldes React'iga ja Angular'iga on Vue'ga kõige lihtsam alustada [4]. Vue'l pole nii palju sisseehitatud funktsioone kui Angular'il, kuid rohkem kui React'il. Antud raamistikku saab kasutada nii SPA kui ka keerukamate veebirakenduste loomiseks. Miinuseks on see, et teiste kahe tehnoloogiaga võrreldes puudub Vue'l nii lai auditoorium. Teadmiste jagamine selle tehnoloogias on alles algstaadiumis [3]. Seetõttu võib suurtesse projektidesse antud raamistiku integreerimine osutada keeruliseks ülesandeks.

Lõputöö autoril on pooleteiseaastane arenduskogemus kasutades React teeki. Analüüsi eesmärk oli välja selgitada, kas on midagi, mis takistaks React'i valimist ja soosiks mõne muu raamistiku valimist. Autor kavatseb püstitatud ülesannete lahendamiseks kasutada ka TypeScript'i, mida saab kasutada koos React teegiga. Võttes arvesse autori varajast kogemust React teegiga ja selle sobivust antud rakenduse nõuete elluviimiseks, on eelistatav kasutada tehnoloogiat, millega autor on tuttav.

2.3 Kasutatud tööriistad ja tehnoloogiad

2.3.1 Visual Studio Code

Eessüsteemi arenduskeskkonnaks oli valitud Microsoft Visual Studio Code, kuna autoril on antud arenduskeskkonna kasutamisel positiivne kogemus. Visual Studio Code on tasuta arenduskeskkond, mille kasutamine on võimalik Windows'il, OS X'il ja Linux'il. Selle pakutavad võimalused sobivad kõigi käesolevas lõputöös püstitatud ülesannete lahendamiseks. Samuti on Visual Studio Code'il lai valik pistikprogramme ja laiendusi, mis muudavad arendamise mugavamaks.

2.3.2 Next.js

Mõnede funktsionaalsete nõuete saavutamiseks peab eessüsteem olema üheleherakendus. Kuna üheleherakendused laadivad sisu AJAX'i kaudu, saab otsingumootor sisuliselt ainult eessüsteemi struktuuri ehk DOM elemendid ilma lehe sisuta [6]. See tähendab, et otsingumootorid ei saa piisavalt infot veebilehelt, et seda õigesti otsingutulemuste loendisse paigutada.

Sobiva tehnoloogia valimisel võttis autor arvesse ka üht kliendi nõuet, milles eessüsteem peab toetama kodulehe otsingumootori optimeerimist. Üheleherakenduste otsingumootori optimeerimise probleemi on võimalik lahendada serveripoolse renderdamisega, mida saab saavutada Next.js'i raamistiku abil. Next.js on React'i peale ehitatud avatud lähtekoodiga JavaScript'i raamistik. Nimelt peab Next.js serveri poolel panema osa HTML'i kokku, tänu millele saab otsingumootor juurdepääsu rakenduse lehe sisule ja metaandmetele.

2.3.3 TypeScript

TypeScript on avatud lähtekoodiga programmeerimiskeel, mis põhineb JavaScript'il. Antud tehnoloogia võimaldab JavaScript'ile staatiliste tüüpide definitsioonide lisamist. See on maailma üks enim kasutatud tehnoloogiaid staatiliste tüüpide lisamiseks. TypeScript'is arendamine võib võtta rohkem aega, kuid tulevikus on lihtsam rakendust edasi arendada ja vältida vigade tekkimist. Koodi käivitamata teavitab TypeScript, kui arendaja on kusagil tüübivea teinud.

TypeScript võimaldab määrata tüüpe, liideseid, loendeid, funktsioonide parameetreid ja nende tagastusväärtuse tüüpe ning märkida millised väljad või parameetrid on mittekohustuslikud [7]. TypeScript pakub sisestamise ajal võimalikke valikuid, mida antud objekt, funktsioon või komponent võib sisaldada. See kiirendab arendamise protsessi ja aitab arendajatel üksteise koodist hõlpsasti aru saada.

Lõputöö autor on rakendanud TypeScript'i, kuna kõik eessüsteemis kuvatud andmed on võetud Amazon S3'st. Kui kliendil tekib soov muuta eessüsteemi sisu, siis saab neid muudatusi teha andmefailis, koodi puutumata. Arvestades sellega, et eessüsteem hakkab kasutama palju andmebaasist pärinevaid andmeid, siis autori arvates TypeScript aitab vältida vigade tekkimist ja kiirendab andmete kasutamist arenduse käigus.

Kuna TypeScript ei ole loetav veebibrauserite poolt, teisendab *transpiler* TypeScript'i koodi JavaScript'iks [8]. Raamistiku analüüsi käigus valiti React teeki. React on JavaScripti teek kasutajaliideste loomiseks, samas kui TypeScript on JavaScript'i täiendatud versioon koos staatiliste tüüpidega, mille teisendab *transpiler* tavaliseks JavaScript'iks. Neid koos kasutades luuakse kasutajaliides staatiliselt tüübitud JavaScript'i abil [9].

2.3.4 Ant Design

Kasutajaliidese arenduse lihtsustamiseks oli kasutusele võetud Ant Design, mis on React'i kasutajaliidese teek ning on ehitatud kasutades TypeScript'i. Ant Design sisaldab kvaliteetseid komponente rikkalike ja interaktiivsete kasutajaliideste loomiseks [10]. Ant Design komponente on lihte kohandada, kuna see on piisavalt populaarne ning omab head kogukonna tuge ja korralikku dokumentatsiooni. Autor kasutas Ant Design'i, kuna see teek sisaldab komponente, mis aitavad realiseerida mõned püstitatud funktsionaalsed ja visuaalsed nõuded.

2.3.5 Styled-components

Styled-components on stiiliteek, mis võimaldab luua uusi React komponentide stiile ja muuta olemasolevaid [11]. React komponentidele, mis on stiliseeritud Styled-components'i abil võib edastada parameetreid. Neid parameetreid saab kasutada kas stiiliväärtuste määramiseks või tingimusefunktsioonide jaoks. Analüüsi käigus sai stiiliteek Styled-components valitud kuna see aitab autoril kasutada stiili teemat. Samuti Styled-components'i on võimalik kasutada koos TypeScript'iga. Kõigile komponendile edastatud stiili parameetritele saab määrata tüüpe, mis aitab vältida vigade tekkimist.

2.4 Kasutajaliidese disain ja vaated

Kliendi poolt oli kasutajaliidese disain ette antud Figma keskkonnas. Figma on tasuta brauseripõhine kasutajaliidese ja kasutajakogemuse kujundusrakendus, mis võimaldab samade prototüüpidega teha koostööd terve meeskonnaga [12]. See ühendab disainereid, projektijuhte, tootejuhte ja arendajaid, et teha koostööd ühes rakenduses. Antud töö skoobis aitas Figma hõlpsasti juurde pääseda kõigile kujundusfailidele, lihtsustades prototüübi ja sellega seotud informatsiooni autorile üleandmise protsessi.

Autori arvates oli mõistlik analüüsida disaini enne arendamise alustamist. Võttes arvesse, et eessüsteemi arendamiseks kasutatakse React teeki, aitab disainianalüüs komponentide struktuuri läbi mõelda. Sellest tuleb juttu eessüsteemi struktuuri käsitlevas peatükis.

Arenduse lihtsustamiseks ja projekti stiili muutmiseks tulevikus, pidas autor vajalikuks salvestada sageli esinevad stiiliväärtused objektina eraldi faili, millele on juurdepääs projekti igast kohast. See kehtib peamiselt värvide, kirjastiilide ja elementide suuruste kohta. Näiteks värvidest oli konstantidena salvestatud viis enimkasutatavat teksti- ja taustavärvi (valge, must, helehall, tumehall ja viridiaanroheline [13]). Selle stiili teemale pääseb juurde eessüsteemi iga komponent, kasutades stiiliteeki Styled-components. Antud lähenemine võimaldab järgida DRY printsiipi, mille põhimõte on vältida korduva koodi kirjutamist.

Eessüsteemil on kolm peamist vaadet. Üks nendest on tehtud autoriseerimata kasutajatele ja kaks on edukalt pileti koodiga sisse loginud kasutajatele.

Autoriseerimata kasutajale on nähtav ainult informatiivne lehekülg koos pileti koodi sisestamise aknaga. Pileti koodi aken asub otse konverentsi plakatil lehe ülaosas. Autor on nõus kliendi valikuga autoriseerimisakna asukoha suhtes. See on intuitiivne ja hõlpsasti ligipääsetav neile, kes on juba konverentsiga tutvunud ja soovivad lihtsalt sisestada pileti koodi ning vaatamist alustada. Peale seda järgneb konverentsi kirjeldus, kava, esinejate ja sponsorite loetelu ning alumine menüü. Alumises menüüs on järgmised valikud: kontaktid, tehniline tugi, korduvad küsimused, kasutustingimused ja privaatsuspoliitika. Igale menüü valikule vajutades avatakse kasutajale vastava sisuga modaalaken.

Autoriseeritud kasutajale on näha täisekraanil ja ekraanisuurusele kohanduv videomängija, mille all on konverentsi sponsorid. Videomängijast paremal asub külgriba menüü, mille avamisel muudetakse videomängija suurust nii, et kogu video oleks nähtav. Külgriba menüüs on kättesaadavad järgmised valikud: vaatajate vestlusring, esinejale küsimuste esitamiseks vestlusring, küsitlused, konverentsi kava ning sponsorite ja esinejate loetelu. Sponsorid saab avada, valides kindla sponsori külgriba menüü aknast või videomängija all olevast sponsorite loendist. Antud vaade sisaldab sponsori infot, tutvustavat videot ja temaga privaatset vestlust, kuhu iga vaataja saab oma Facebooki kontot kasutades sisse logida.

Käesolevas peatükis analüüsis autor, milliseid tehnoloogiaid oleks vaja määratud ülesannete lahendamiseks, ning analüüsis kliendi poolt saadud kasutajaliidese disaini Figma keskkonas. Järgmises peatükis kirjeldab autor eessüsteemi arendust ja saadud tulemusi.

3 Valminud eessüsteemi kirjeldus

Käesolevas peatükis kirjeldatakse eessüsteemi tehnilist teostust. Autor tutvustab rakenduse arhitektuuri, eessüsteemi failide ja andmete struktuuri, eessüsteemi vaadete arendust ja kolmandate osapoolte lahenduste integreerimist.

React kasutab komponendipõhist lähenemist, mille üheks eeliseks on korduvkasutatavad komponendid. Korduvkasutatavate komponentide korralikuks toimimiseks peavad need olema piisavalt üldised ja vältima keerukat ärioloogikat. Kui komponent sisaldab enda sees keerulist ärioloogikat, ei raskenda see mitte ainult taaskasutamist, vaid komponent muutub ka vähem hooldatavaks [14].

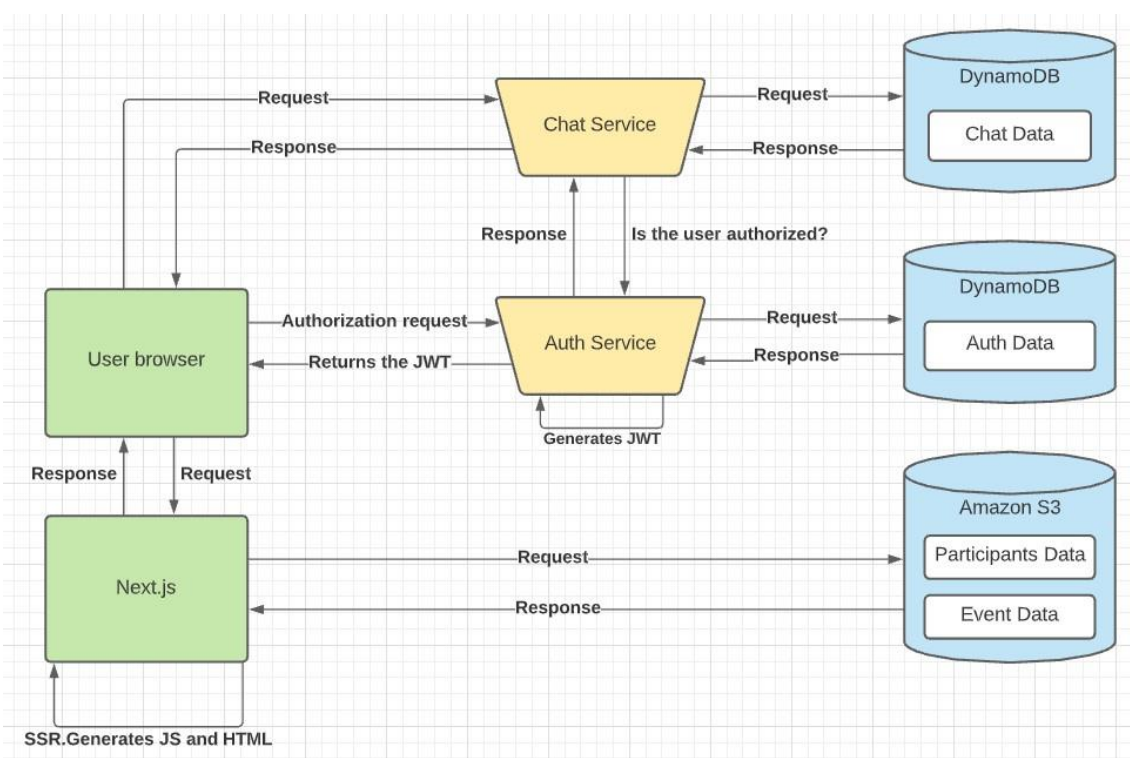
Kasutades kliendi poolt esitatud kasutajaliidese disaini, otsis autor korduvaid elemente, millest saab arendusprotsessi kiirendamiseks luua korduvkasutatavaid komponente. Järgmisi elemente kasutati mitmetes vaadetes: kava, sponsorite loetelu, modaalne aken ja videomängija. Autor pidi selliste komponentide arendust hoolikalt läbi mõtlema, kuna igal korduvkasutataval komponendil olid erinevad stiiliväärtused ja sisendparameetrid olenevalt kasutaja vaatest.

3.1 Rakenduse arhitektuur

Selleks, et paremini mõista, kuidas andmed hakkavad väljaspool eessüsteemi liikuma ja kuidas muud eessüsteemiga seotud teenused omavahel suhtlevad, otsustas autor joonistada terve rakenduse arhitektuuri mudeli. Autor pole tegelenud seotud teenusete ja andmebaaside arendusega. Rakenduse arhitektuur koosneb järgnevatest osadest:

- Brauser (*User browser*) - kuvab konverentsi rakendust.
- Next.js - tänu SSR'ile on rakendus SEO sõbralik. Kui brauserist tuleb päring, koostab Next.js HTML'i ja JavaScript'i ning tagastab selle brauserisse.
- Vestlusteenus (*Chat Service*) - teenust kasutatakse siis, kui kasutaja avab vestluse külgriba menüüs. Samuti käesolev teenus suhtleb ka autentimisteenusega, et kontrollida, kas kasutajal on õigust vestlusse kirjutamiseks.

- Autentimisteenus (*Auth Service*) - teenust kasutatakse siis, kui kasutaja üritab enda pileti koodiga autoriseerida ja vestluse kirjutada.
- Vestlusteenuse andmebaas (*Chat Data*) - hoiab jututooga seotud andmeid.
- Autentimisteenuse andmebaas (*Auth Data*) - hoiab kasutajate andmeid koos pileti koodiga.
- Andmeobjektid (*Amazon S3*) - selles andmeobjektis on salvestatud konverentsi ja selle osalejate andmed. Peaaegu kõik konverentsi eessüsteemi sisu võetakse sellest andmeobjektist. Osalejate andmetest on salvestatud ainult avalikud andmed, mida soovitakse jagada teiste osalejatega.



Joonis 1. Rakenduse arhitektuur.

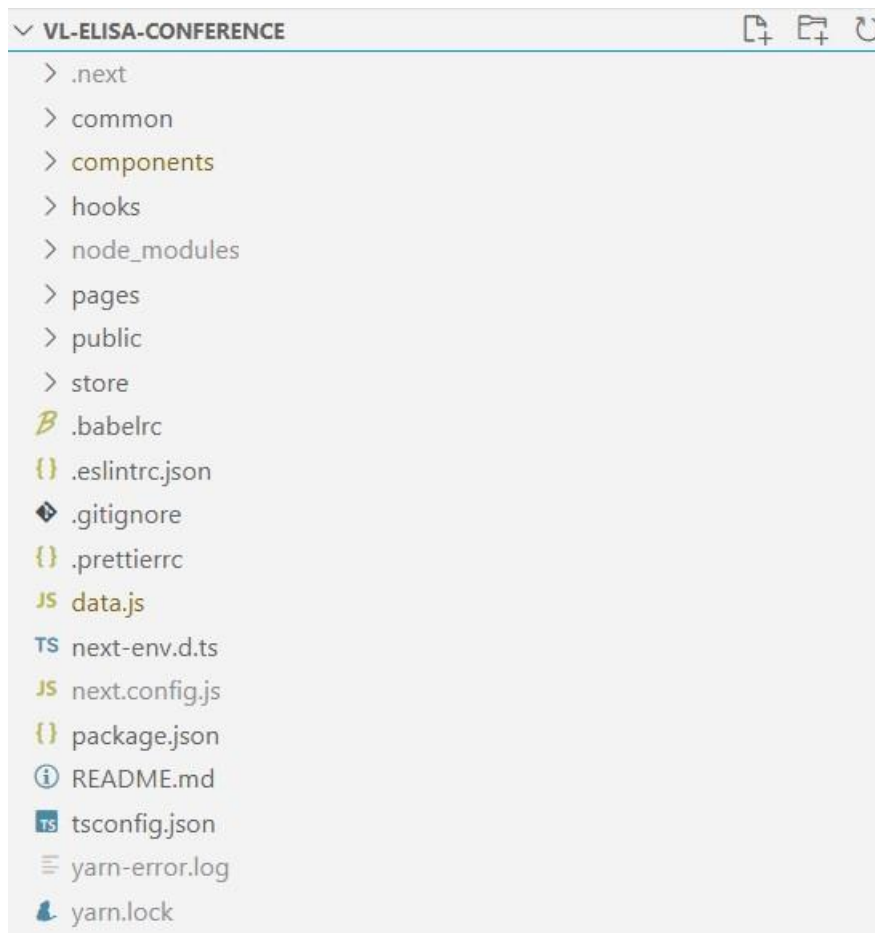
3.2 Eessüsteemi struktuur

Käesoleva eessüsteemi struktuur on ehitatud vastavalt Next.js'i kaustade struktureerimise parimate tavadele [15],[16],[17]. Kaustade loetelu ja nende sisu:

- *Common* – TypeScript'i tüübid ja rakenduse stiili teema fail. Antud failid aitavad vältida koodi ümberkirjutamist ja järgida DRY printsiipi.
- *Components* – selles kaustas asuvad kõik komponendid, kus need jagunevad ka eraldi alamkaustadeks. Komponentide stiilid on kirjutatud kasutades stiiliteeki Styled-components. Iga komponendi stiilid kirjutatakse samasse faili, kuid

väljaspool komponendi renderdamise meetodit, kuna vastasel juhul luuakse need iga renderdamise käigus uuesti. Stiliseeritud komponendi määratlemine renderdamise meetodis pärsib vahemällu salvestamist ja aeglustab renderdamise kiirust [18].

- *Hooks* – mõni loogika, mis on vajalik komponentide jaoks toimingute ja andmete edastamiseks.
- *Pages* – Next.js rakenduse struktuuri 3 peamist komponenti: *app*, *document* ja *index*. Next.js kasutab lehtede initsialiseerimiseks ja globaalse oleku salvestamiseks *App* komponenti. *Document* komponent vastutab HTML'i dokumendi ehitamise ja täiendamise eest. Ja *index* komponent on kogu rakenduse alguspunkt.
- *Public* – hoiab kõik eessüsteemiga seotud SVG'd ja pildid.
- *Store* – globaalsete olekute failid.



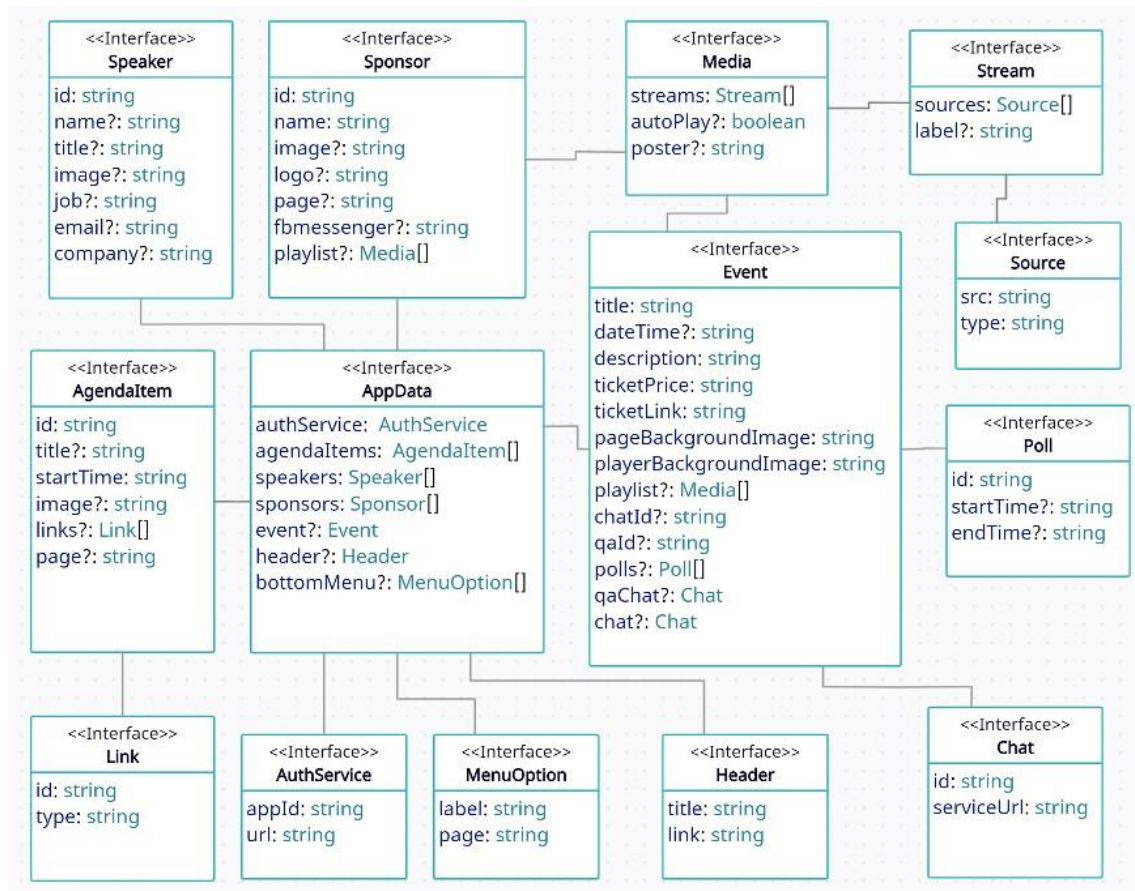
Joonis 2. Eessüsteemi rakenduse struktuur.

3.3 Andmetüüpide struktuur

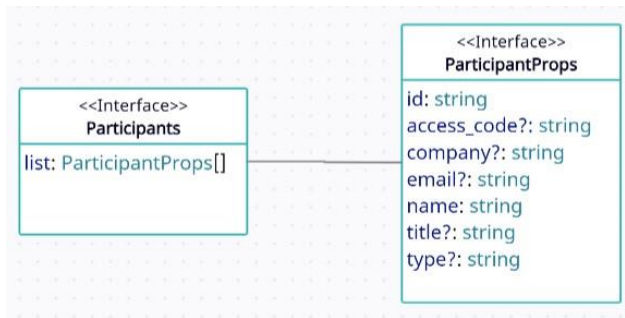
Välja arvatud autoriseerimisteenus ja vestlusteenus, pärinevad kõik eessüsteemiga seotud andmed Amazon S3'st. Amazon S3 on andmesalvestusteenus, mida saab kasutada andmete turvaliseks salvestamiseks ja hankimiseks [19]. Amazon S3'e salvestusruumides on salvestatud konverentsi ja osalejate andmeobjektid.

Tänu Next.js'i asünkroonsele funktsioonile *getServerSideProps* hangitakse mõlemaid andmeobjekte enne eessüsteemi renderdamist [20]. Arendus-protsessi kiirendamiseks ja vigade vältimiseks on konverentside ja osalejate andmeobjektidele lisatud TypeScript'i staatilisi tüüpe.

Järgmisena esitab autor andmetüüpide vaheliste seoste diagrammid. Konverentsi andmeobjekti peamine tüüp on *AppData* (Joonis 2) ja osalejate andmeobjekti peamine tüüp on *Participants* (Joonis 3). Tänu tüübi diagrammidele on näha mõlema andmeobjekti struktuuri.



Joonis 3. Konverentsi andmeobjekti tüüpide struktuur.



Joonis 4. Osalejate andmeobjekti tüüpide struktuur.

3.4 Autoriseerimata kasutaja vaade

Eessüsteemi arendus algas autoriseerimata kasutaja vaate loomisest. Tegu on rohkem informatiivse ja konverentsi tutvustava vaatega. Kasutades kliendi poolt esitatud kasutajaliidese disaini, jagas autor autoriseerimata kasutaja vaate loogilisteks komponendi osadeks. Nendeks osadeks on: konverentsi pealkiri, sisselogimise vorm, kirjeldus, kava, esinejate ja sponsorite loetelu ning alumine menüü. Järgnevalt autor tutvustab mõningaid komponente.

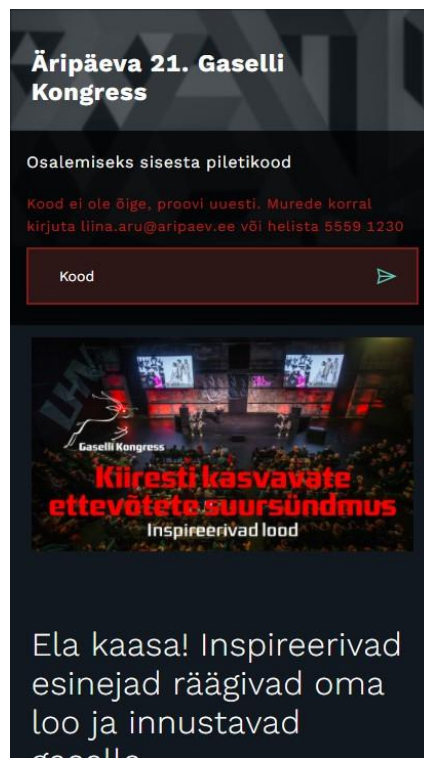
3.4.1 Autoriseerimine

Kasutajad on eelnevalt ostnud kliendilt konverentsipileti koos ainulaadse pileti koodiga. Kui kasutaja on konverentsi rakenduse avanud, siis autoriseerimis akna koodi sisestusväli on fookuses, mis vähendab kasutaja poolt klikkide arvu. Autoriseerimise vormi eesmärk on kontrollida kas sisestatud kasutaja poolt pileti kood on olemas vastavas autentimisteenuse andmebaasis. Sisestatud pileti kood saadetakse serveripoolsesse autentimisteenusesse, mis kasutab JWT'd. Autentimisteenuse andmebaasis on kirjas kõik pileti koodid, mis on olemas käesoleva konverentsi jaoks. Autentimisteenus teeb päringu andmebaasi koos kasutaja poolt sisestatud koodiga. Kui vastav kood on andmebaasis olemas, tagastatakse sellele vastav kasutaja ID. Kasutaja ID't sisestatakse JWT'sse, mida allkirjastatakse ja tagastatakse brauserisse [21]. Seejärel suunatakse kasutaja videomängija vaatesse.



Joonis 5. Sisselogimis vaade.

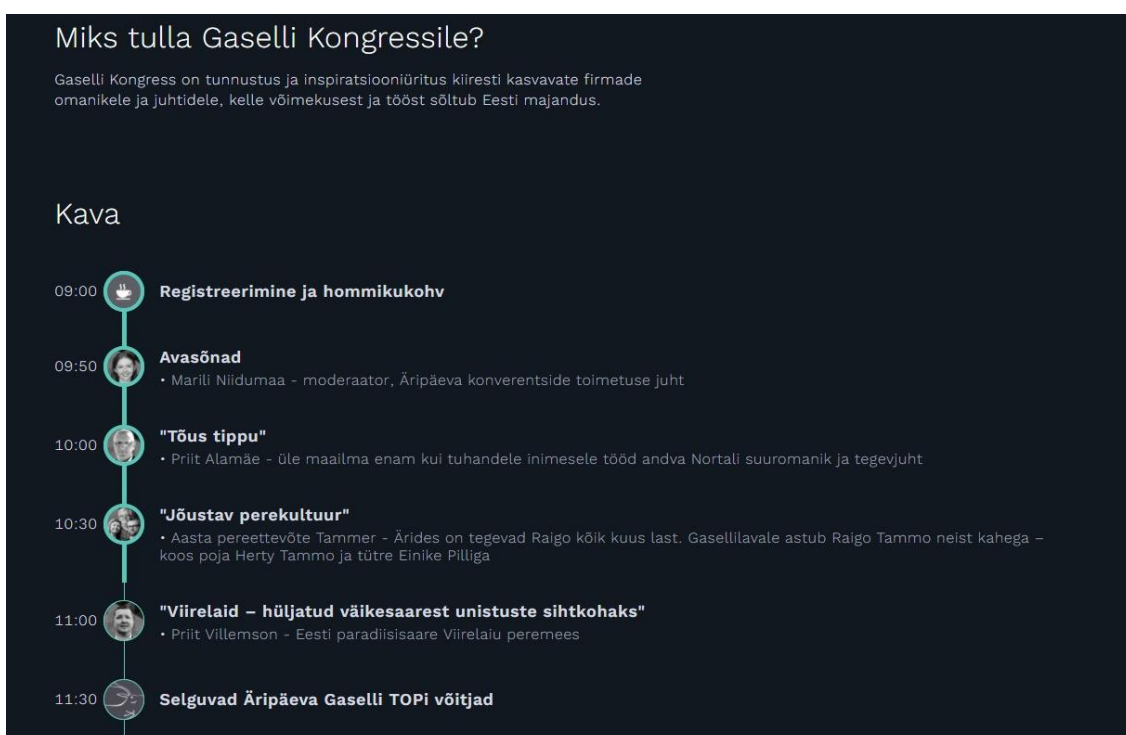
Kui sisestatud pileti kood puudub autentimisteenuse andmebaasis, tagastatakse brauserisse veateade. Kasutajale kuvatakse vastava välja juures veateadet ja kontaktandmeid, kuhu saab pöörduda autoriseerimisega seotud probleemide korral (Joonis 6). Autoriseerimise komponendile järgneb konverentsi kirjeldus.



Joonis 6. Veateade vale pileti koodiga autoriseerimisel nutitefonist.

3.4.2 Konverentsi kava

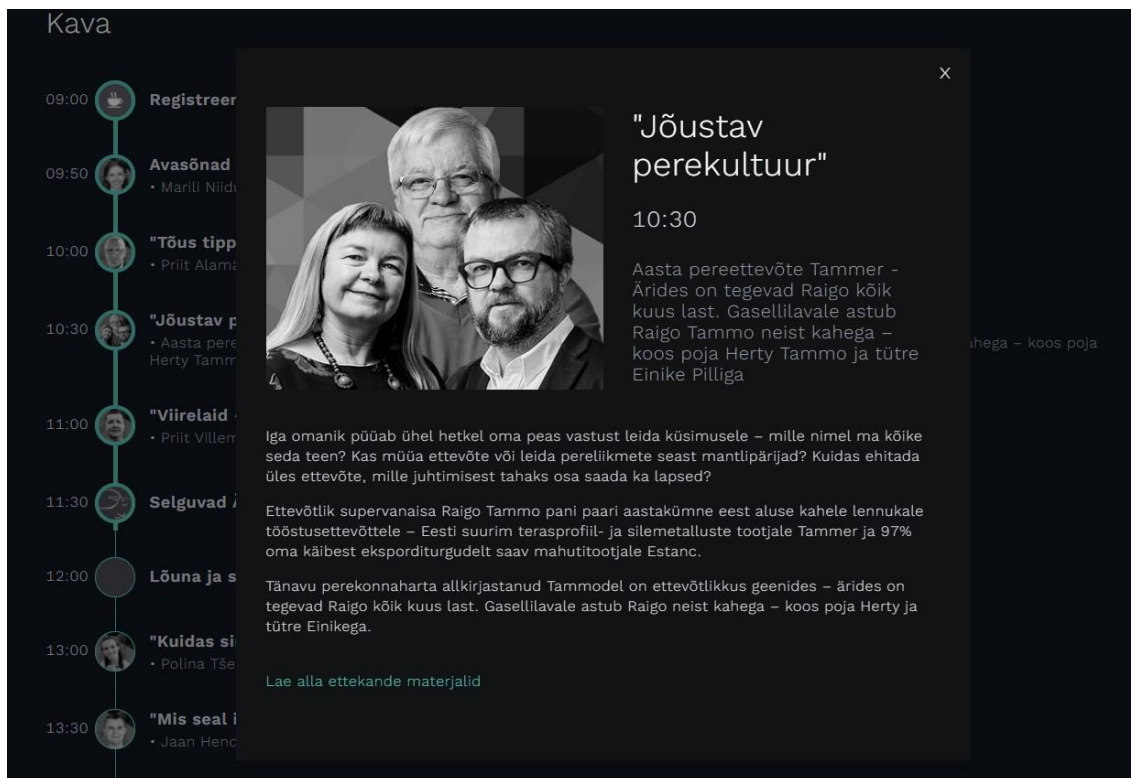
Pärast konverentsi kirjeldust järgneb selle kava. Kava komponenti kasutatakse autoriseerimata kasutaja vaates ja autoriseeritud kasutaja vaate külgriba menüüs. Käesoleva komponendi eesmärgiks on kasutajale teada anda, mis konverentsi etapp antud hetkel käimas on ja kui kaua see veel kestab. Joonisel 7 on välja toodud näide, kus on parasjagu käimas konverentsi neljas etapp. Kava komponendi sisendparameetriteks on konverentsi sündmuste massiiv ja tüübi nimi. Tüübi nime kasutatakse mingi kindla stiili kuvamiseks sõltuvalt sellest, mis vaates antud korduvkasutatavat komponenti kasutatakse.



Joonis 7. Konverentsi kava autoriseerimata kasutaja vaates.

Kava komponendi arendamise aluseks oli võetud Ant Design'i komponent *Timeline* [22]. *Timeline* komponent omas ülesande lahendamiseks sobilikku struktuuri. Iga *Timeline*'i rida oli seotud vertikaalse joonega koos ringi-kujulise sümboliga, millest paremal ja vasakul oli võimalik määrata väärtusi. Autori eesmärgiks oli rakendada kliendi poolt nõutud disaini *Timeline* komponendile ja täiendada seda lisafunktsionaalsustega. Kui sündmuse algusaeg on vastavalt kasutaja seadmes olevale ajale minevikus, muutuvad selle pildi ääred laiemaks. Samuti vertikaalse joone laius peaks muutuma sõltuvalt sellest, kui palju aega on sündmuse algusest selle lõpuni möödunud (Joonis 7). Vertikaalse joone ja pildi ääre laiused peavad reaalselt muutuma ilma lehte värskendamata.

Sündmuse nimel vajutades avaneb modaalne aken, mis sisaldab informatsiooni esineja või muu kavale vastava sündmuse kohta. See annab kasutajale eelnevalt teada, mis teemal iga esineja kavatseb rääkida. Samuti esinejate loo kirjelduse all on olemas link, tänu millele saab nende ettekande materjale alla laadida (Joonis 8).



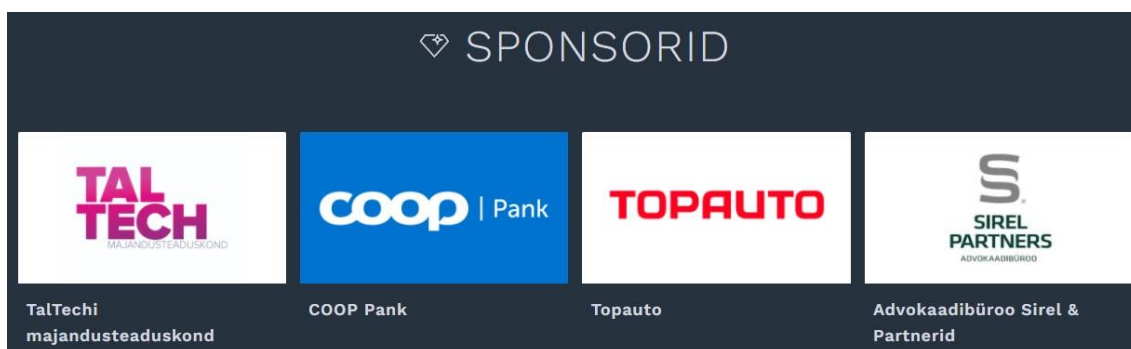
Joonis 8. Esinejate info modaalses aknas.

3.4.3 Esinejate ja sponsorite loetelud

Konverentsi kavale järgnevad esinejate ja sponsorite nimekirjad, mis on lisatud kasutajale kiire ülevaate saamiseks kõigist esinejatest ja sponsoritest konverentsil. Esinejate komponenti kasutatakse ainult selles vaates. Sponsorite komponenti kasutatakse ka autoriseeritud kasutajate vaates, kus ta omab teistsuguseid stiile. Seda silmas pidades on sponsori komponendile sisendparameetrina lisatud tüübi nimi, mille abil komponent kasutab õigeid stiile.



Joonis 9. Esinejate loetelu.



Joonis 10. Sponsorite loetelu.

3.5 Autoriseeritud kasutaja vaated

Peale kasutaja edukat autoriseerimist, suunatakse teda autoriseeritud kasutaja põhivaatesse. Antud vaade koosneb peamiselt otseülekande videomängijast ja külgriba menüüst. Külgriba menüü kaudu saab üle minna sponsori detailvaatesse ja osalejate nimekirja vaatesse. Järgnevalt kirjeldab autor mainitud vaateid ja komponente.

3.5.1 Videomängija

Antud eessüsteem on arendatud niimoodi, et vaataja saaks kasutada kogu rakenduse funktsionaalsust ilma otseülekande vaatamist katkestamata. Eessüsteemi videomängijaks oli valitud THEOplayer, millel on asjakohane dokumentatsioon, mis aitab koodi kirjutades vajalikku funktsionaalsust lisada [23]. Videomängija kasutajaliides oli muudetud vastavalt kliendi kasutajaliidese nõutele.

Kliendi üheks funktsionaalseks nõudeks oli võimalus vaadata konverentsi ülekannet mitmes keeles. Tavaliselt on igas videomängijas olemas konfiguratsiooni võimalused, kus saab video kvaliteeti muuta või kontrollida subtiitrite kuvamist. Arenduse käigus oli

lisatud konfiguratsiooni menüüsse võimalus valida videot teises keeles. Igale keele valikule oli vastav otseülekanne video link. Keele muutmisel käivitab videomängija ülekanne määratud lingilt.

Videomängija all on loetelu konverentside sponsoritest, mille peale vajutades kuvatakse kindla sponsori detailset vaadet. Vaate muutmisel ei tohiks otseülekanne video katkeda, vaid see liigub lehe ülaossa.



Joonis 11. Videomängija vaade koos avatud külgriba menüüga.

3.5.2 Külgriba menüü

Pärast kasutaja autoriseerimist, kuvatakse talle videomängijat koos avatud külgriba menüüga. Kui kasutaja ei vaja külgriba menüüd ega menüü valikuid, saab neid ükshaaval peita. Juhul, kui kasutaja sulgeb kogemata külgriba menüü ja ei tea, kuhu see kadus, aitab intuitiivne kasutajaliides tal sellest aru saada. Esimene vihje on paremas ülemises nurgas olev nupp koos vastava SVG ikooniga, millel vajutades menüü avaneb või sulgub. Teine visuaalne vihje on lehe parema piiri laius. Kui menüü on suletud, muutub see laiemaks, nii et kasutaja teaks, kuhu menüü on peidetud (Joonis 12).

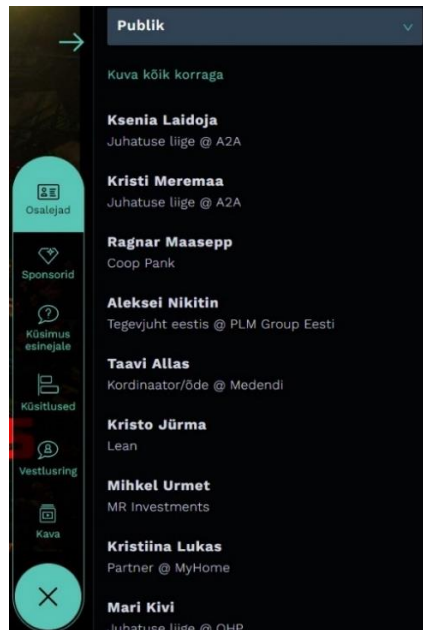


Joonis 12. Videomängija vaade koos suletud külgriba menüüga.

Navigeerides külgriba menüüs, salvestatakse viimane valitud kategooria globaalse olekusse. Kui kasutaja tuleb tagasi sponsori detailvaatest või osalejate nimekirja vaatest, kuvatakse talle viimati valitud külgriba menüü kategooria. Antud funktsionaalsus vähendab kasutaja klikkide arvu ja aitab kasutajakogemust paremaks teha.

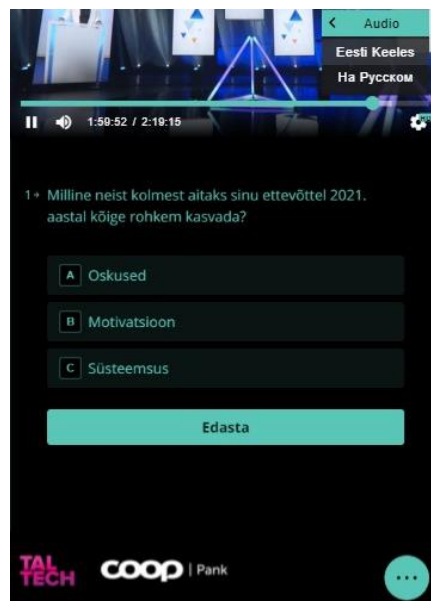
Külgriba menüüs saab avada järgmiseid vaateid: osalejad, sponsorid, küsimus esinejale, küsitlused, vestlusring ja kava. Kava ja sponsorite sisu kuvamiseks kasutatakse samasuguseid komponente, mis olid kirjeldatud peatükkides 3.4.2 ja 3.4.3. Selles vaates oli sponsorite komponendile lisatud funktsionaalsus, mis võimaldab kuvada detailset infot kindla sponsori kohta. Antud funktsionaalsust tutvustatakse peatükis 3.5.3.

Osalejate aknas kuvatakse käesoleva konverentsi vaatajaid, esinejaid ja korraldajaid. Kindla kategooria kuvamiseks kasutajal on võimalik valida filtreerimis aknast vastava osalejate tüübi. Iga osaleja kohta kuvatakse tema nimi ja info, mille ta sisestas pileti ostmisel, teiste konverentsi osalejatega jagamiseks (Joonis 13).



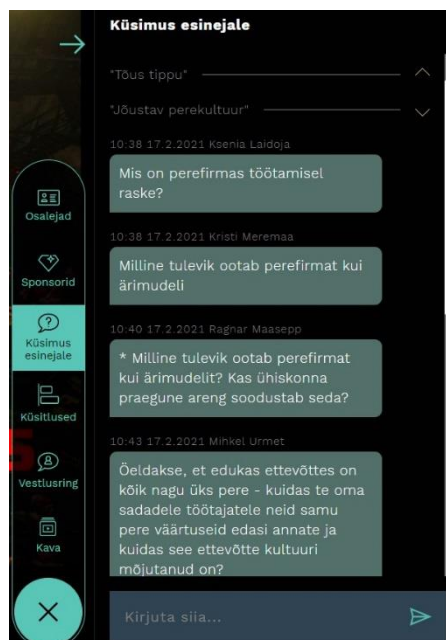
Joonis 13. Külgriba menüüs olev konverentsi osalejate aken.

Külgriba menüü küsitluste aknas saavad osalejad vastata konverentsi korraldajate poolt koostatud küsitlustele. Küsitlused on olulised kasutajate suhtlemis ja tagasiside elemendid, et pakkuda paremat kasutajakogemust. Küsitluste läbiviimiseks oli kasutatud Typeform. Typeform on suurepärase kolmanda osapoole tööriist, mis võimaldab hõlpsasti luua kena välimusega küsitlusi. See omab põhjalikku dokumentatsiooni, mis aitab integreerida ja konfigureerida küsitluse vormi vastavalt kliendi nõutele ning salvestada vastuseid näiteks Excel faili [24].



Joonis 14. Külgriba menüüs olev küsitlus nutitefonist.

Külgriba menüü akendes nimetustega „küsimus esinejale“ ja „vestlusring“ kasutatakse sama jututoa komponendi. Antud komponenti arendusega pole autor tegelenud. Jututoa komponendi üheks eesmärgiks oli sõnumite sortimine vastavalt sellele, millise sündmuse jooksul kava järgi oli see postitatud. Kui jututuppa oli postitatud sõnum, mida kasutaja pole veel näinud, siis vastava menüü nuppu peale tekib punase värviga ring. Sõnumi postitamisel kasutab jututoa komponent kasutaja nime, mis oli sisestatud kasutaja poolt pileti ostmise ajal. Kui kasutaja on edukalt autoriseeritud, siis saadakse andmebaasist vastava pileti koodile kasutaja nime, mida edaspidi kasutab jututoa komponent.



Joonis 15. Külgriba menüüs olev küsimuste jaoks jututuba.

3.5.3 Sponsorite detailvaade

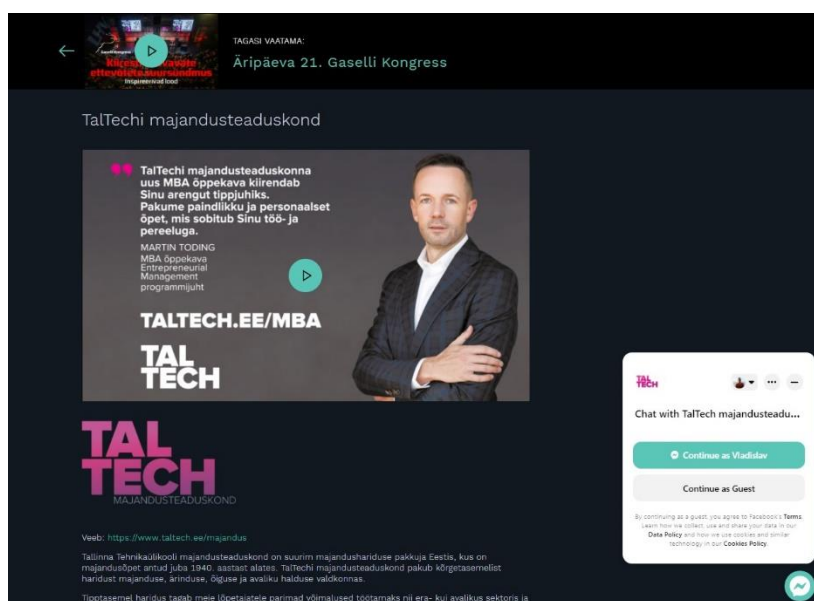
Vajutades sponsori logol külgriba menüü aknas või videomängija all olevas loendis, liigub katkestamata otseülekande video lehe ülaossa ja selle all kuvatakse kasutajale detailne ülevaade valitud sponsorist.

Vaate muutmiseks ilma videot katkestamata tuli autoril ehitada komponendid videomängija komponendi ümber ja kuvada neid vastavalt vaatele. Selleks oli rakenduses tehtud globaalne olek, mida muudetakse siis, kui kasutaja siseneb või lahkub sponsori vaatest. Valides sponsori, peidetakse kõik videomängija ümber olevad komponendid ja kuvatakse selle all sponsori vaade. Antud lähenemine aitab säilitada videomängija komponenti seda uuesti renderdamata, vastasel juhul otseülekande video oleks peatatud.

Konverentsi osalejate nimekirja vaade töötab sama funktsionaalsusega, mida saab avada külgriba menüü osalejate aknast, vajutades loendi kuvamise nuppu.

Käesolevas vaates kuvatakse sponsori info, tutvustav video ja privaatne jututuba sponsoriga. Sponsori jututoa tehnoloogiaks sai valitud Facebook'i pistikprogramm, kuhu iga vaataja saab enda Facebook'i kontot kasutades sisse logida. Samuti saab antud pistikprogrammi kasutada anonüümselt. Turvalisuse huvides renderdatakse pistikprogrammi ainult siis, kui see on laaditud domeenile, mis on konfiguratsiooni lubatud loendisse lisatud. Facebook'i jututoa integreerimiseks peab domeen toetama HTTPS-protokoll. Domeeninimi tuleb lisada tervenisti, näiteks: <https://www.messenger.com/>. IP-aadresse ja *localhost*'i ei saa lubatud loendisse lisada [25]. Võttes arvesse, et arendades kohaliku hosti peal polnud võimalik testida antud pistikprogrammi, siis lahenduse testimiseks juurutati eessüsteemi Vercel'i peale. Vercel võimaldab arendajatel üles laadida ning hostida eessüsteeme ja veebiteenuseid [26]. Eessüsteemi üleslaadimisel oli Vercel tagastanud domeeninime, mida saab kasutada hostitud eessüsteemi avamiseks. Selleks, et Facebook'i pistikprogrammi oleks võimalik testida tuli Vercel'i poolt tagastatud domeeninime lisada eelnevalt mainitud lubatud loendisse.

Kui kasutaja oli sisse loginud enda Facebook'i kontoga saab ta jätkata sponsoriga vestlemist oma tavalise Facebook Messenger'i abil ka pärast konverentsi rakendusest lahkumist, kuna kogu vestlusajalugu salvestatakse automaatselt.



Joonis 16. Sponsori detailvaade.

4 Edasiarendamise võimalused

Eessüsteemi edasiarenduseks võiks lisada järgmised funktsionaalsed nõuded:

- Lisada kasutajaliidese keele valimise võimalus. Antud lõputöö raames lisati videomängijale funktsionaalsus, mis võimaldab edastada video otseülekannet mitmes keeles. Seda arvestades oleks loogiline teha kasutajaliidese info kuvamist lisaks ka inglise ja vene keeles.
- Ühel konverentsil võib olla mitu lava korraga. Suurte ürituste ja konverentside korraldamisel võib tekkida olukord, kus otseülekandeid edastatakse korraga mitmelt lavalt. Seda silmas pidades võiks lisada eessüsteemi lavade valiku, nende olemasolul. Iga lava jaoks oleks eraldi kava, küsimuste jututuba ning video otseülekanne.
- Külgriba menüü võiks kuvada küsitluste statistikat pärast nende läbiviimist. Paljudel kasutajatel oleks huvitav näha statistikat küsitlustes, milles nad on osalenud.

5 Kokkuvõte

Käesoleva diplomitöö eesmärgiks oli luua konverentside otseülekannete vaatamiseks eessüsteem, mis võimaldaks vaatajatel suhelda ja koos toimetada esinejate ja teiste konverentsi osalejatega, sõltumata sellest, kas nad vaatavad konverentsi kohapealt või otseülekandes.

Analüüsi osa algas funktsionaalsete ja mittefunktsionaalsete nõuete väljaselgitamisega. Töö eesmärgi saavutamiseks määras autor kindlaks kasutatavaid tehnoloogiaid ja põhjendas nende valikut. Arenduse meetodikaks oli valitud komponendipõhine arendus kasutades React'i ja TypeScript'i.

Lõputöö tulemuseks sai arendatud rakenduse eessüsteem, mis vastab kõigile kliendi nõuetele. Selleks, et tulevikus lihtsustada rakenduse laiendamist ja edasiarendamist oli komponendipõhine arendus eelnevalt hästi läbimõeldud.

Antud lõputöö raames õppis autor palju uusi tehnoloogiaid ja sai eelnevalt omandatud teadmisi proovile panna. Töö kirjutamise hetkeks on klient juba edukalt otseülekandes konverentsi läbi viinud ja andnud positiivset tagasisidet. Seda silmas pidades võib öelda, et lõputöö eesmärgid on edukalt täidetud.

Kasutatud kirjandus

- [1] „Front-end Frameworks,“ [Võrgumaterjal]. Available: <https://2020.stateofjs.com/en-US/technologies/front-end-frameworks>. [Kasutatud 15.03.2021].
- [2] „Что такое Virtual DOM?,“ [Võrgumaterjal]. Available: <https://habr.com/ru/post/256965>. [Kasutatud 18.03.2021].
- [3] „React или Angular или Vue.js – что выбрать?,“ [Võrgumaterjal]. Available: <https://habr.com/ru/post/476312>. [Kasutatud 20.03.2021].
- [4] German Sharabok, „Vue vs React vs Angular,“ [Võrgumaterjal]. Available: <https://towardsdatascience.com/vue-vs-react-vs-angular-114718eac3c5>. [Kasutatud 20.03.2021].
- [5] „React vs Angular Compared: Which One Suits Your Project Better?,“ [Võrgumaterjal]. Available: <https://www.altexsoft.com/blog/engineering/react-vs-angular-compared-which-one-suits-your-project-better>. [Kasutatud 12.04.2021].
- [6] Svitlana Varaksina, „Single-Page-Applications vs Multi-Page Application,“ [Võrgumaterjal]. Available: <https://themindstudios.com/blog/spa-vs-mpa>. [Kasutatud 08.04.2021].
- [7] Mahesh Haldar, „Why and How to use TypeScript in your React App,“ [Võrgumaterjal]. Available: <https://blog.bitsrc.io/why-and-how-use-typescript-in-your-react-app-60e8987be8de#:~:text=The%20type%2C%20interfaces%2C%20enums%2C,understand%20each%20other's%20code%20easily>. [Kasutatud 08.04.2021].
- [8] „Transpiler vs Compiler,“ [Võrgumaterjal]. Available: <https://howtodoinjava.com/typescript/transpiler-vs-compiler/#:~:text=A%20good%20example%20of%20transpiler,very%20different%20in%20abstraction%20level>. [Kasutatud 14.04.2021].
- [9] Joe Previte, „React with TypeScript: Best Practices,“ [Võrgumaterjal]. Available: <https://www.sitepoint.com/react-with-typescript-best-practices/#:~:text=How%20React%20and%20TypeScript%20Work%20Together&text=React%20is%20a%20E2%80%9CJavaScript%20library,a%20typed%20version%20of%20JavaScript>. [Kasutatud 14.04.2021].
- [10] „Ant Design of React,“ [Võrgumaterjal]. Available: <https://ant.design/docs/react/introduce>. [Kasutatud 09.04.2021].
- [11] Nwose Lotanna, „8 reasons to use styled-components,“ [Võrgumaterjal]. Available: <https://blog.logrocket.com/8-reasons-to-use-styled-components-cf3788f0bb4d>. [Kasutatud 09.04.2021].
- [12] Ben Kopf, „The Power of Figma as a Design Tool,“ [Võrgumaterjal]. Available: <https://www.toptal.com/designers/ui/figma-design-tool>. [Kasutatud 27.03.2021].
- [13] „A nomenclature of colors for naturalists: and compendium of useful knowledge for ornithologists,“ [Võrgumaterjal]. Available: <https://archive.org/details/nomenclatureofco00ridg/page/n173/mode/2up>. [Kasutatud 05.04.2021].

- [14] Nirmalya Ghosh, „Building Reusable Components Using React“, [Vörgumaterjal]. Available: <https://buttercms.com/blog/building-reusable-components-using-react#:~:text=Conclusion-What%20are%20reusable%20components%3F,free%20from%20complex%20business%20logic>. [Kasutatud 15.04.2021].
- [15] Pablo A. Del Valle H., „An opinionated basic Next.js files and directories structure“, [Vörgumaterjal]. Available: <https://medium.com/@pablo.delvalle.cr/an-opinionated-basic-next-js-files-and-directories-structure-88fefa2aa759>. [Kasutatud 10.04.2021].
- [16] Sergio Xalambri, „Next.js File Structure“, [Vörgumaterjal]. Available: <https://sergiodxa.com/articles/next-file-structure>. [Kasutatud 10.04.2021].
- [17] „Next.js Documentation“, [Vörgumaterjal]. Available: <https://nextjs.org/docs/getting-started>. [Kasutatud 11.04.2021].
- [18] „Styled-components. Basics“, [Vörgumaterjal]. Available: <https://styled-components.com/docs/basics#passed-props>. [Kasutatud 11.04.2021].
- [19] „Amazon Simple Storage Service Documentation“, [Vörgumaterjal]. Available: <https://docs.aws.amazon.com/AmazonS3/latest/userguide/Welcome.html>. [Kasutatud 18.04.2021].
- [20] „Next.js Data Fetching“, [Vörgumaterjal]. Available: <https://nextjs.org/docs/basic-features/data-fetching>. [Kasutatud 18.04.2021].
- [21] „JSON Web Tokens“, [Vörgumaterjal]. Available: <https://auth0.com/docs/tokens/json-web-tokens>. [Kasutatud 24.04.2021].
- [22] „Ant Design. Timeline“, [Vörgumaterjal]. Available: <https://ant.design/components/timeline>. [Kasutatud 17.04.2021].
- [23] „THEOplayer Documentation“, [Vörgumaterjal]. Available: <https://docs.theoplayer.com>. [Kasutatud 25.04.2021].
- [24] „Typeform Documentation“, [Vörgumaterjal]. Available: <https://developer.typeform.com/get-started>. [Kasutatud 15.05.2021].
- [25] „Facebook Messenger Chat Plugin“, [Vörgumaterjal]. Available: <https://developers.facebook.com/docs/messenger-platform/discovery/facebook-chat-plugin/#login>. [Kasutatud 27.04.2021].
- [26] „Vercel“, [Vörgumaterjal]. Available: <https://vercel.com/about>. [Kasutatud 27.04.2021].

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks¹

Mina, Vladislav Lahtarin

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Konverentside otseülekande eessüsteemi arendus ettevõtte Videolevels näitel“, mille juhendaja on Joel Kivi.
 - 1.1. reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
 - 1.2. üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. Olen teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.
3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

15.05.2021

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 tähtjaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktile 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.