

TALLINNA TEHNIKAÜLIKOOL  
Infotehnoloogia teaduskond

Kristina Revizor 205986IADB

# **Fännikirjanduse platvormi klientrakenduse arendus**

Bakalaureusetöö

Juhendaja: Meelis Antoi

Magistrikraad

Tallinn 2023

## **Autorideklaratsioon**

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

Autor: Kristina Revizor

15.05.2023

## **Annotatsioon**

Käesoleva bakalaureusetöö eesmärgiks on luua fännikirjanduse platvormi kliendirakendus, mis on suunatud massikultuuri edendamiseks eelkõige eesti keelt kõnelevatele inimestele. Fännikirjandusega tutvumine ja selle jagamine aitab tekitada huvi lugemise vastu, parandada kirjutamisoskust, arendada kujutlusvõimet ja leida mõttekaaslasi liitudes fännikogukondadesse. Veebirakenduse lõppkasutajaks võib saada iga huviline, kuna vanuse- ega muid piiranguid ei ole.

Lõputöö raames luuakse kliendirakendus võimalikult lihtsa kasutajaliidesega, serverrakendus valmib kaastudeng Vladislav Stepanenko bakalaureusetöona “Fännikirjanduse platvormi serverrakenduse arendus”. Analüüsitakse ka renderdamise protsessi lähtudes kasutaja soovist ja serverrakendusest saadud andmetest.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 36 leheküljel, 5 peatükki, 7 joonist, 1 tabelit

## **Abstract**

### **Development of the Front-end Application for Fan Fiction Platform**

The aim of the current bachelor's thesis is to create a front-end application for a fanfiction platform that would help aimed at promoting mass culture, especially among Estonian speakers. Exploring and sharing fan fiction helps to generate interest in reading, improve writing skills, develop imagination, and find like-minded individuals by joining fan communities. Anyone interested can become an end-user of the web application, as there are no age or other restrictions.

As part of the thesis, a customer application with a simple UI will be created, and the server application will be developed by fellow student Vladislav Stepanenko's bachelor's thesis "Development of the Backend Application for Fan Fiction Platform". The rendering process will also be analyzed based on user preferences and data obtained from the server application.

## Lühendite ja mõistete sõnastik

API	Application Programming Interface
CRUD	Create, Read, Update, Delete
CSS	Cascading Style Sheets
DOM	Document Object Model
MPA	Multi Page Application
SPA	Single Page Application

# Sisukord

1 Sissejuhatus .....	11
1.1 Metoodika.....	11
2 Probleemi püstitamine .....	13
2.1 Sihtgruppi määramine.....	14
2.1.1 Uuringu läbiviimine.....	14
2.1.2 Olemasolevad lahendused .....	14
2.1.3 Pakutav lahendus .....	16
3 Arendatava veebirakenduse analüüs.....	17
3.1 Uuringu tulemused .....	17
3.2 Nõuete määramine .....	17
3.2.1 Funktsionaalsed nõuded .....	18
3.2.2 Mittefunktsionaalsed nõuded.....	19
3.2.3 Tulevikus loodavad nõuded.....	19
3.3 Tehnoloogia valik.....	20
3.3.1 Klientrakenduse tehnoloogiate valik .....	20
3.3.2 Serverrakenduse tehnoloogiate valik.....	21
3.4 Arenduskeskkonna valik.....	21
3.4.1 Koodihalduse keskkonna valik.....	21
3.4.2 Integreeritud arenduskeskkonna valik .....	22
4 Kliendirakenduse arendus.....	23

4.1 Veebirakenduse disain .....	23
4.2 Veebilehekülje peamised komponendid.....	24
4.2.1 Navigeerimispaneel .....	25
4.2.2 Peamise lehekülje sisu konteiner.....	25
4.2.3 Praeguse lehekülje SPA konteiner.....	25
4.2.4 Hüplikaken.....	26
4.3 Veebirakenduse ülesehitus .....	28
4.4 Rakenduse struktuur .....	28
4.5 Lexical .....	29
4.6 Kliendi- ja serverrakenduse sidustamine.....	30
5 Hinnang loodud klientrakendusele .....	33
5.1 Nõuete täitmine ja kasutatavus .....	33
5.2 Kasutatud tehnoloogiate uudsus ja sobivus .....	34
5.3 Kasutatavuse testimine .....	35
5.4 Võimalused edasiseks arenduseks .....	36
Kokkuvõte .....	37
Kasutatud kirjandus .....	38
Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks .....	40
Lisa 2 – Uuringu Tulemused .....	41
Lisa 3 - Näide CSS adapteeritud ekraani suuruse kohta, failist Navigation.css.....	43
Lisa 4 - Infosüsteemi skeem .....	45

Lisa 5 - Tuleviku fännikirjanduse platvormi andmebaas .....	46
Lisa 6 - Kasutatavuse testülesanded .....	47
Lisa 7 – Loodud prototüübi struktuur.....	49
Lisa 8 – Klientrakendusele link.....	50



## Jooniste loetelu

Joonis 1 .Veebirakenduse komponendid. ....	24
Joonis 2. Fännikirjanduse pealeht koodis. ....	26
Joonis 3. Hüpikaken fännikirjanduse lisamiseks loendisse pärast klõpsamist "Loendisse" fännikirjanduse lehel.....	27
Joonis 4. Teede deklaratsioon MPA töötamiseks.....	27
Joonis 5 Redaktori väljanägemine lehel. ....	30
Joonis 6 Asünkroonne funktsioon BackendUtils.getAllStatuses() tagastab IStatus liidesele kuuluvate objektide nimekirja lubadusena.....	31
Joonis 7 IStatus liides defineerib fännilugude staatuse objektides sisalduvate parameetrite omadused.....	32

## **Tabelite loetelu**

Tabel 1. Ülesannete lahendamisele kulutatud aeg .....	35
---	----

# 1 Sissejuhatus

Raamatute lugemine, videomängude mängimine, filmi või seriaali vaatamine on igapäevased tegevused ilma milleta suur hulk inimesi ei kujuta enda elu ette. Kõik hea peab aga ükskord lõppema. Suur osa fännidest ei suuda lihtsalt istuda ja oodata, kui ilmub raamatu järgmine osa või telesarja uus hooaeg, vaid levitavad fännikunsti, riietuvad lemmiktegelasteks, kirjutavad fännikirjandust, viies massikultuuri populaarsuse väga suurele kasvule.

Fännikirjandus on kirjandus, mis on kirjutatud tuntud kirjandustegelastest, kuid teiste autorite poolt. Tavaliselt kirjutavad seda mingisuguse raamatu või filmi huvilised teistele fännidele, kes juba teavad neid tegelasi ja tegevuspaiku. Fännikirjandust võib kirjutada ka laiemalt tuntud inimestest või nendest, keda tunnevad vaid loo kirjutaja ja lugejad. [1]

Tänapäeval on kättesaadav laialdane valik veebiplatvorme, mis sobivad fännikirjanduse jagamiseks ja kirjutamiseks. Olemasolevate rakenduste põhipuuduseks on nende keerukus algkasutajatele, ebaloogiline või ülekoormatud veebilehtede disain. Kahjuks keskenduvad need veebilehed ühekeelsele fännikogukonnale või on mitmekeelne valik olemas, kuid seal puudub eesti keel. Huvilised süvenevad maailma mõeldes alternatiivseid lõppe välja, viivad tegelasi kõige ootamatutesse kohtadesse või täidavad süžeelisi lünki. See võimaldab süveneda lemmikteoste maailma ja ennast loominguliselt väljendada ning arendada kirjutamisoskust.

Käesoleva lõputöö raames luuakse eestikeelne intuitiivse kasutajaliidesega kliendirakendus avatud lähtekoodiga. Kliendirakenduse kasutatav serverrakendus valmib kaastudeng Vladislav Stepanenko bakalaureusetöona “Fännikirjanduse platvormi serverrakenduse arendus”. Valmiva veebirakenduse abil saavad kasutajad fännikirjandusega tutvudes ja enda poolt loodud teoseid jagades arendada fantaasiat, kasvatada armastust lugemise ja kirjanduse vastu ning rikastada sõnavara.

## 1.1 Metoodika

Enne sobiva IT-lahenduse arendusprotsessiga alustamist, formuleeritakse eelkõige käsitlevat probleemi, seejärel jätkatakse eksisteeriva alternatiiv veebirakenduste avastamise ning potentsiaalse kasutajate seas uuringu läbiviimisega.

Kogutud andmete põhjal määratletakse täpsemaid funktsionaalsed ja mittefunktsionaalsed nõuded, et tagada loodetaval prototüübil vajalikku funktsionaalsust. Parima tulemuse saavutamise eesmärgil analüüsitakse ka kaasaegseid tehnoloogilisi vahendeid valides sobilikuma. Käesoleva lõputöös tegeldakse klientrakenduse loomisega, seega tehnoloogiate analüüsimisel enam tähelepanu keskendutakse kliendipoolsetele tehnoloogiate valikule.

Arendamise käigu kirjeldamisel rõhutatakse tööprotsessi ülevaatest klientrakenduse kõige tähtsamatele punktidele: ühemõtteline ja töökindel funktsionaalsus ning algkasutajale lihtsasti mõistetav ja visuaalselt atraktiivne kasutajaliides. Järgmiseks etapiks antakse täidetud nõuete järgi loodud prototüübile hinnangut testimise käigus saadud tulemuste põhjal. Antud töö lõppfaasis kirjeldatakse valminud klientrakenduse lahenduse tulemusi ning edaspidiseid arendusvõimalusi serverrakendusega ühendamisel.

## 2 Probleemi püstitamine

Suurt emotsionaalset pinget põhjustavad elusündmused, nagu näiteks töökohast ilmajäämine või lähedase inimese kaotus, tõstavad oluliselt riski haigestuda kehalistesse haigustesse ning kutsuvad sageli esile hingeelu häireid. [2] Eksisteerivad erinevad viisid stressi maandamiseks, kuid parimaks ravimiks peetakse loovust. [3] Enda loomingulisust saab väljendada kirjutades lühikest teksti fokuseerimata peategelaste iseloomule või süvenemata detailidesse, kuna lugejad on juba sellest teadlikud. [4]

Lausest „lapsed ei loe” on saanud aksioom, väide, mis ei vaja tõestamist. Millegipärast on see postulaat just nüüdisaegses kooliarhitektuuris ajanud võrseid. Kui lapsed raamatuid ei loe ja raamatukogus ei käi, siis tuleb viia raamatud sinna, kus lapsed käivad, et nad hakkaksid lugema. [6] Tänapäeva lapsed on imikust peale tuttavad pigem arvuti- ja videomaailmaga, kui raamatutega. [7] Noortele sisse kasvatama armastust lugemise ja kirjanduse vastu saab juba väiksest peale, lihtsate ja huvipakkuvate juttude kaudu veebipõhisel platvormil. See võimaldab mitte ainult süveneda lemmikteoste maailma ning ennast loominguliselt väljendada, kuid ka parandada kirjutamisoskust viies seda kõrgemale tasemele, mis omakorda aitab oluliselt arendada lugemiskiirust ja rikastada kasutatavat sõnavara.

Nüüdisajal on massikultuuri populaarsus maailmas plahvatuslikult kasvanud. Massikultuur on kõige laiemale tarbijaskonnale kättesaadav, kuna see on peamiselt tehniliste vahendite varal paljundatud kultuurivorm. Massikultuuri olulised vahendajad on ajaviiteline massimeedia, kino, levimuusika, ajaviitekirjandus ning muud meelelahutuslikke või hariva ja teatuid hoiakuid kujundava sisuga harrastuslikud tegevusvaldkonnad. [5] Fännikirjandus kuulub samuti nende vahendajate hulka. Tuhanded fännikirjanduse lood on raskusteta leitavad erinevatel platvormidel, kuid põhipuuduseks on suunatus ühekeelsele fännikogukonnale või mitme keele valik välja arvatud eesti keel. Lähtudes kirjeldatud probleemist ja eesmärgist kaasata ka Eestis elavaid inimesi massikultuuri levikule, tekkis vajadus luua eestikeelne platvorm.

Eestikeelset veebipõhist fännikirjanduse platvormi, kus kasutaja saab ennast väljendada, ei leidu. Eestlased kasutavad tihti fännikirjandusega tutvumiseks võõrkeelseid platvorme. Võõrkeelte valdamine madalal tasemel piirab saada täielikku kasutajakogemust rakenduse funktsionaalsusest. Lisaks lugedes võõrkeeles võivad olulisemad süžee

pöörded jääda vahele. Pakutavaks lahenduseks on eestikeelne fännikirjanduse platvorm, mis aitab säilitada keelt ja laiendada antud huviala Eestis. Fännikirjanduse arendusprotsessis plaanitakse võtta arvesse Eestis elavaid inimesi ning luua funktsionaalsust lähtudes nende soovidest.

## **2.1 Sihtgruppi määramine**

Eestikeelse platvormi arendamine on mõeldud eelkõige eesti keelt kõnelevatele inimestele, kellel on sarnased raamatute, filmide ja muusika eelistused. Vanuselisi ega muid piiranguid ei ole, seega arendatav kliendirakendus peab olema algkasutajale intuiitiivne kasutajaliidese poolest ning olema loogiliselt struktureeritud ilma ülekoormava sisuta.

### **2.1.1 Uuringu läbiviimine**

Uuringu eesmärk on välja selgitada, kas Eestis elavad inimesed tunnevad, mis on fännikirjandus, mis funktsionaalsust nad peavad tähtsamaks ja millest tunnevad puudust. Lähtudes nende vajadustest ja soovidest määratletakse täpsemaid nõudeid arendatavale platvormile. Planeeritava veebipõhise platvormi populariseerimiseks reklaami kaudu selgitatakse välja ka potentsiaalsete kasutajate vanusegrupp, et kasutada tulevikus õigeid sotsiaalmeedia kanaleid.

### **2.1.2 Olemasolevad lahendused**

Antud peatükis kirjeldab autor planeeritava arendatava platvormi võrdlemist sarnaste funktsionaalsustega veebiplatvormidel. Võrdlemiseks võetakse kaks kõige populaarsemat fännikirjanduse platvormi jagamiseks ja tutvumiseks järgmiste aspektide poolest: sisu kvaliteet, kasutusmugavus ning sisu kättesaadavus [3]. Analüüsi käigus soovitakse välja selgitada nende funktsionaalsused ning kasutajaliidese leitud eelised ja puudused. Avastatud nõrgemad küljed võetakse arvesse protüübi arendusprotsessis ja parandatakse neid eesmärgiga tagada kasutajasõbralikkust kõikidele kasutajatele. Tugevamad küljed võetakse edasiarenduse aluseks.

Analüüsimist alustatakse kõige populaarsemast veebirakendusest - Wattpad. Wattpad on lugemisplatvorm, mis kaotab vaheseina lugejate ja kirjanike vahel. See julgustab kasutajaid looma ja jagama oma lugusid kõikides žanrites, alates noortekirjandusest kuni

luule, seikluskirjanduse ja paljude teiste teosteni. [17] Antud veebirakendus on mitmekeelne, mis suurendab kasutajate arvu, kuid eesti keelt ei ole.

Sisselogimata kasutaja jaoks on Wappadi veebilehe mõned pakutavad teenused piiratud, kuid registreerimine ei võta palju aega ning sisse logimine on võimalik erinevate kontode kaudu, näiteks: Google, Facebook. Ainult sisselogitud kasutaja saab autorit jälgida, et olla kursis värskemate lisamistega. Platvormil on registreeritud üle 90 miljoni kasutaja, seega lisatavad teosed on alati kontrollimata ja võivad sisaldada näiteks diskrimineerivaid väljendeid.

Lugejate mugavuse ja kiire otsimise jaoks on sisseehitatud otsingumootoris erinevad parameetrid: žanr, pealkiri, peatükkide arv. Rakendada saab ka filtreid ning sorteerida uudsuse või hinna järgi (tasuta või tasuline). Tulemuste seast saab samuti näha fännikirjanduse lühidat kirjeldust koos vaatamiste numbriga. Enne kindla teosega tutvumist peab vaatama segavat videoreklaami, mille vältimine on võimalik ainult Premium konto ostmisel. Pärast videoreklaami vaatamist suunatakse kasutaja detailsemale fännikirjanduse veebilehele, kus lisaks eelnevale kirjeldusele on määratud meeldimiste arv, viimase värskenduse kuupäev, staatus, võimalus lisada teos isiklikku või avalikku lugemisloendisse. Samuti saab näha ka sisukorda ja valida peatükki, et suunduda juba selle lugemisele. Autoriseeritud kasutaja annab soovi korral tagasisidet, kas peatükki all kommenteerides või kirjutades otse autorile.

Teiseks analüüsitavaks platvormiks on AO3 või Archive Of Our Own, mille eesmärgiks on fännide huvi laiendada, pakkudes juurdepääsu ning säilitades fännitööde ja fännikultuuri ajalugu nende erinevates vormides. Veebirakendus pakub mittetulunduslikku kesket kohta fännitöödele, kasutades avatud lähtekoodiga arhiveerimistarkvara. [8]

Nimetatud veebileht ei oma kaasaegset disaini: navigeerimispaneel ja mõned sildid on eraldatud ainult punase värviga ja taust on valge. Algakasutajatele võib kasutajaliides olla keerukas. Sisselogimata kasutaja saab tutvuda fännikirjandusega ning kasutada navigeerimispaneelis asuvaid erinevate sisseehitatud otsingumootoritega.

Otsingumootoreid on mitu, saab valida, kas otsida tööde, järjehoidjate, märgendite või autorite järgi. Valides kindla kriteeriumi saab edasi sisestada ka detailsemalt otsingu parameetreid. Liiga lai otsingumootorite valik võib raskendada kasutajal otsuse tegemist.

Sisselogimise võimalus on olemas, kuid selleks peab saama kutse. Kutsesüsteem on mõeldud selleks, et vältida rämpsposti ning hoida fännikogukond kontrolli all. Seega algajale kirjanikule võib olla probleemiks enda teoseid teistega jagada.

### **2.1.3 Pakutav lahendus**

Eksisteerivate veebipõhiste platvormide ühiseks puuduseks on eesti keele puudumine. Kui kasutaja võõrkeele oskused on madalal tasemel võivad tekkida probleemid veebilehe funktsionaalsuse mõistmisega. Keelebarjääri tõttu võib olla veebilehe funktsionaalne kasutamine piiratud või tekib ajakulu selle funktsionaalsusega tutvumiseks. Suurenenud ajakulu võib tekitada ka liiga lai erinevate otsingumootorite valik, suure parameetrite hulgaga ning liiga kitsendanud parameetrite valikuga. Kasutajakogemust mõjutab ka veebirakenduste disain, olemasolevad lahendused on minimalistlikud ja liiga lihtsa kujundusega, mis ei kaasa uusi kasutajaid ega ei jää meelde.

Lahendusena pakutakse mugav ja intuitiivne kasutajaliides sõltumata arvutioskusest. Planeeritav prototüüp aitab mitte ainult eestikeelse IT-lahenduste laiendamisele, kuid annab ka platvormi eneseväljendamiseks emakeeles. Olemasolevate teoste põhjal loodud uued teosed aitavad arendada kujutlusvõimet. Arendatava platvormi kasutajad saavad süveneda alternatiivmaailma mõeldes oma lõppu välja, viies teatud tegelasi kõige ootamatutesse kohtadesse või täites süželisi lünki.

Planeeritava tulemusena on fännikirjanduse platvormi loomine ja selle arendamine. Potentsiaalseteks kasutajateks saavad eesti keelt kõnelevad inimesed. Arendataval platvormil tagatakse vajalikke funktsioone, mis saavad selgeks uuringu läbiviimisel ning on autori poolt täiustamisel.



### **3 Arendatava veebirakenduse analüüs**

Analüüsi protsessis uuritakse välja kindlad nõuded ja seejärel valitakse tehnoloogiad lahenduse elluviimiseks ning viimasena kujundatakse ka planeeritav kasutajakogemus.

#### **3.1 Uuringu tulemused**

Eesmärgiga tagada arendataval prototüübil kõiki vajalikke funktsionaalsusi, viis autor läbi uuringu eestlaste seas, kuna loodav lahendus on mõeldud neile. Uuringu käigus küsitles autor 116 inimest, vanusegrupis 13-19 aastat (23%), 20-30 aastat (64,4%), üle 30 aasta (12,6%). Osalejatest enamus ehk 62% ei loe fännikirjandust. Ülejäänud 38% loevad inglise keeles ning eelistavad võrdselt selleks kas Wattpadi või Archive on Our Own platvormi. Kõige populaarsemaks osalejate poolt sisestatud vastuseks sai venekeelne veebipõhine fännikirjanduse platvorm Fanfiction.net. Uuringu koostaja oletas madalat teadlikkust fännikirjanduse kohta. Seetõttu täpsustas küsimustikus, et vastates järgnevatele küsimustele osalejad lähtuksid soovidest, mida nad eeldaksid näha antud platvormil. Kolm neljandiku vastanutest peab sisseehitatud otsingumootorit vajalikuks ning sooviksid otsida järgmiste parameetrite järgi: fännikogukond, märgendid, lehekülgede arv, meeldimiste arv, pealkiri, loomise kuupäev, sõnade arv. Samuti vastasid osalejad, et sooviksid kommentaare jätta ja suhelda teiste lugejate või kirjanikega kirj vahetuse teel, tagades eesmärgi anda tagasisidet, näha mitmekülgsid arvamusi, arutleda teiste kasutajatega, vahetada ideid ja saada inspiratsiooni. Enamus (82,9%) sooviks näha enne lugemist lühikirjeldust. Olemasolevate platvormide miinuseks peeti otsingumootorite ebamugavust ja kasutatavust ning märkisid veebilehe sisu ebastruktureeritaks.

Lisas 2 on uuringu tulemused.

#### **3.2 Nõuete määramine**

Nõuete määramisel võeti arvesse eksisteerivate platvormide analüüsi tulemusi ja läbiviidud uuringu kogutud andmeid. Nõuded olid täiendatud autori poolt.

### 3.2.1 Funktsionaalsed nõuded

Arendataval platvormil on kaks kasutaja rolli: autoriseerimata (tavakasutaja) ja autoriseeritud kasutaja (kas lugeja või autor). Kõik funktsionaalsed nõuded, mis on saadaval tavakasutajale, on saadaval ka lugejale või autorile.

Tavakasutaja funktsionaalsed nõuded:

- Tavakasutajana soovin luua kontot;
- Tavakasutajana soovin otsida huvipakkuvaid fännikirjandusi rakendades erinevaid parameetreid (autor, pealkiri, fännikogukond, lehtede ja meeldimiste arv);
- Tavakasutajana soovin lugeda fännikirjandust ja suunduda järgmistele lehekülgedele;
- Tavakasutajana soovin näha fännikirjanduse lühikirjeldust;
- Tavakasutajana soovin valida fännikirjanduse peatükki;
- Tavakasutajana soovin näha meeldimiste ja jälgimiste arvu;

Lugeja funktsionaalsed nõuded:

- Lugejana soovin sisse ja välja logida;
- Lugejana soovin otsida huvipakkuvaid fännikirjandusi rakendades erinevaid parameetreid (autor, pealkiri, fännikogukond, lehtede ja meeldimiste arv);
- Lugejana soovin lugeda fännikirjandust jasuunduda järgmistele lehekülgedele;
- Lugejana soovin grupeerida fännikirjandusi kaustadesse vastavalt mugavusele;
- Lugejana soovin näha fännikirjanduse lühikirjeldust;
- Lugejana soovin valida peatükki;
- Lugejana soovin märkida lemmikuks ja jälgida fännikirjandust;
- Lugejana soovin näha meeldimiste ja jälgimiste arvu;

Autori funktsionaalsed nõuded:

- Autorina soovin sisse ja välja logida;
- Autorina soovin lisada fännikirjandust koos pealkirjaga;
- Autorina soovin fännikirjanduse trükkimisel muuta fondi suurust või muuta teksti paksemaks, kursiiviks või allakriipsutuseks;
- Autorina soovin näha meeldimiste arvu ja jälgimiste arvuga;

- Autorina soovin muuta fännikirjanduse pealkirja ja sisu;
- Autorina soovin kustutada fännikirjandust;
- Autorina soovin kustutada kindla fännikirjanduse peatükki;
- Autorina soovin kustutada kindla fännikirjanduse lühikirjeldust (pealkiri, staatus, fännikogukonnad, märgendid, kirjeldus).

### **3.2.2 Mittefunktsionaalsed nõuded**

Tavakasutaja mittefunktsionaalsed nõuded:

- Tavakasutajana soovin veebirakendust kasutada vabalt valitud veebibrauseris;
- Tavakasutajana soovin meeldejäätavat ja intuitiivset kasutajaliidest.

Autori mittefunktsionaalsed nõuded:

- Autorina soovin kasutada veebirakendust kasutada vabalt valitud veebibrauseris;
- Autorina soovin meeldejäätavat ja intuitiivset kasutajaliidest.

Lugeja mittefunktsionaalsed nõuded:

- Lugejana soovin kasutada veebirakendust kasutada vabalt valitud veebibrauseris;
- Lugejana soovin meeldejäätavat ja intuitiivset kasutajaliidest.

### **3.2.3 Tulevikus loodavad nõuded**

Ajalise piirangute tõttu jäid mõned planeeritavad funktsionaalsed ja mittefunktsionaalsed nõuded katmata. Antud bakalaaurusetöös jäi ka administraatori rolli lisamine skoobist välja.

Lugeja funktsionaalsed nõuded:

- Lugejana soovin jätta oma arvamust fännikirjanduse kohta kommentaarides;
- Lugejana soovin lugeda teiste kasutajate kommentaare;
- Lugejana soovin liituda autori kirjandusloominguga;
- Lugejana soovin tellida teavitusi uue fännikirjanduse kohta;
- Lugejana soovin alla laadida fännikirjandust.

Autori funktsionaalsed nõuded:

- Autorina soovin näha vaatamiste arvu;
- Autorina soovin lisada oma fännikirjandusele vanusepiiranguid;
- Autorina soovin esitada kaebuse lugeja kommentaari kotha.

Administraatori funktsionaalsed nõuded:

- Administraatorina soovin tegeleda kaebuste juhtimisega;
- Administraatorina soovin kustutada fännikirjandusi, mis solvavad teisi lugejaid;
- Administraatorina soovin hallata platvormi, et tagada vastavust platvormi reeglitele.

Lugeja mittefunktsionaalsed nõuded:

- Lugejana soovin ligipääsu personaalsete andmete turvalisusele.

Autori mittefunktsionaalsed nõuded:

- Autorina soovin ligipääsu personaalsete andmete turvalisusele.

Administraatori mittefunktsionaalsed nõuded:

- Administraatorina soovin ligipääsu personaalsete andmete turvalisusele.

### **3.3 Tehnoloogia valik**

Sobivate tehnoloogiate valik on keeruline, kuna maailmas on kasutusele võetud suur valik programmeerimiskeeli ja raamistikke. [9]

#### **3.3.1 Klientrakenduse tehnoloogiate valik**

Puhtas JavaScriptis, raamistike kasutamiseta on võimalik kirjutada klientrakenduse, kuid see võib tuua raskusi veebirakenduse arendamisel, näiteks: tekitada brauserite vahelise ühilduvuse tõrkeid, arengutsükli pikendamine. Raamistike abil saab hallata rakenduse arhitektuuri, disainimalli ja koodi struktuuri. Raamistike valimisel autor tõi esile kaht kõige parima klientrakenduse arendamiseks 2023 aasta seisuga: Reacti ja Angulari.

Need on mõlemad populaarsed ja laialdaselt kasutatavad JavaScripti raamistikud veebirakenduste loomiseks.

Reactiga on lihtne alustada ja see muudab selle sobivaks valikuks projektidele, millel napib ressursse keeruka arhitektuuri nullist väljatöötamiseks. React võimaldab alustada lihtsate kasutajaliidestega ning suurendada järk-järgult projekti kasvades uusi komponente ja funktsioone lisades. Reacti modulaarne arhitektuur võimaldab kasutajaliidese jagamist väiksemateks, taaskasutatavateks komponentideks. See muudab kasutajaliidese hooldamise ja ajakohastamise lihtsamaks aja jooksul, kuna muudatusi saab teha üksikute komponentide kaupa ilma kogu rakendust mõjutamata.

### **3.3.2 Serverrakenduse tehnoloogiate valik**

Serverrakendus valmib klientrakendusest hiljem, aga programmeerimiskeeleks planeeritakse valida C# koos ASP.NET Core raamistikuga ja Entity Framework Core PostgresSQL andmebaasiga suhtlemiseks. Serverrakenduse arenduskeskkonnaks kavatakse valida Rider IDE ja versioonihalduskeskkonnaks Bitbucket. Serverrakendus valmib kaastudeng Vladislav Stepanenko bakalaureusetöona “Fännikirjanduse platvormi serverrakenduse arendus”

## **3.4 Arenduskeskkonna valik**

Analüüsi käigus vaadeldakse kõige tuntumaid versioonihalduskeskkondi ja integreeritud arenduskeskkondi, et võrrelda neid ja valida arendatavaks prototüübiks.

### **3.4.1 Koodihalduse keskkonna valik**

Versioonihalduskeskkonna valimine on oluline arendusprotsessi osa, kuna selles mängib suurt rolli tootlikkus. Seetõttu otsuse tegemisel peab veenduma, et valitud platvorm vastab arendaja ja tema meeskonna vajadustele ning eesmärkide saavutamisele. Seega versioonihalduskeskkonnade võrdlemise käigus võetakse kõige sagedamini kasutatavaid: GitHub ja Bitbucket. [10] Mõlemad platvormid pakuvad funktsioone nagu versioonihaldus, probleemide jälgimine, tõmbepäringud ja koodi ülevaated, kuid nende kahe platvormi vahel on mõned erinevused.

GitHub on veebipõhine tarkvaraarenduse platvorm. Seda kasutatakse tarkvaraprojektide salvestamiseks, jälgimiseks ja koostööks. See võimaldab arendajatel jagada lihtsalt koodifaile ja teha koostööd teiste arendajatega avatud lähtekoodiga projektides. [13] GitHub kuulub Microsoftile, mis on tekitanud mõningaid muresid selle tuleviku osas, kuid see jääb siiski kõrgelt usaldusväärseks ja kindlaks platvormiks.

Bitbucket omakorda kuulub Atlassianile, ettevõttele, mis on spetsialiseerunud tarkvaraarendusvahenditele. Bitbucket on vähem tuntud kui GitHub, kuid sellel on lojaalne kasutajaskond ja see pakub mõningaid funktsioone.

Projektil on potentsiaali edasi areneda, mis tähendab, et sellesse võib kaasata rohkem osalejaid. Valiti just Bitbucket, kuna tal on parem integratsioon teiste Atlassiani tööriistadega, nagu Jira ja Confluence, mis võivad teha kogu arendusvoogu ühes kohas hõlpsamini hallatavaks. [12]

### **3.4.2 Integreeritud arenduskeskkonna valik**

Integreeritud arenduskeskkondade hulgast valis autor kahe kõige populaarsema vahel: WebStorm ja VSCode.[14]

Visual Studio Code on tasuta koodiredaktor, mis aitab kasutajatel kiiresti programmeerimisega alustada. Seda saab kasutada mistahes programmeerimiskeele koodi kirjutamiseks, sealhulgas JavaScripti jaoks. [15] Selle lõi Microsoft ja see on kiiresti muutunud üheks populaarseimaks koodiredaktoriks arendajate seas.

Autor valis WebStormi järgmistel põhjustel:

- WebStorm on spetsiaalselt mõeldud JavaScripti arendamiseks, seega pakub see palju funktsioone, mis on kohandatud just JavaScripti arendajatele.
- WebStorm on spetsialiseerunud ainult JavaScripti arendamisele ning see suudab pakkuda selle konkreetse keele jaoks rohkem arendatud funktsioone ja tööriistu, kui VSCode.
- WebStormi peetakse üldiselt kiiremaks ja reageerivamaks, kui VSCode.
- WebStormil on arenenud koodianalüüsi tööriistad, mis aitavad tuvastada potentsiaalseid vigu ja halbu programmeerimise harjumusi enne, kui need muutuvad probleemiks. See aitab kirjutada puhtamat ja tõhusamat koodi.

## 4 Kliendirakenduse arendus

Kliendirakenduse arendamisel võeti arvesse analüüsi käigus kindlaks tehtud nõudeid ning tehniliste vahendite valikut. Veebiteenusel on see arendatud täielikult eraldiseisvalt ning kasutusel on serveripoolse rakendusega suhtlemiseks püsivad ühenduspunktid, mille kaudu vahetatavad andmekujud sobituvad nii kliendi- kui ka serverirakenduse vajadustele.

Veebilehekülge on konstrueeritud eraldiseisvaid elemente kombineerides, mida on võimalik edaspidi arendada. Antud arendusraamistiku tugevaimad küljed on rakenduse lihtne hallatavus ning optimaalne efektiivsus. Viimane põhjustab Reacti olekust sõltuvat arengutsüklit. See tähendab, et elementides on võimalik omistada muutujad, mille seis jälgitakse ning nende värskendamisel ei laeta uuesti tervet lehekülge, vaid üksnes uusi väärtusi sisaldavaid alamelemente.

Antud lõputöö prototüüp on loodud Create React App lahenduse toel, mis võimaldab kliendirakenduse baasprojekti luua mugavalt ja kiiresti käsurea kaudu, nõudmata üleaarust seadistamist. Lõputööd arendati Typescriptis, mitte JavaScriptis, seetõttu lisati projekti loomise käsule oodatud tagajärje saamiseks vastav näitaja. Käsu käivitamiseks ning edaspidiseks arenduseks on vajalik arvutis npm-i ehk Node Package Manageri eksisteerimine, mis haldab projektis kasutatud teke ning käivitab serverit rakenduse arendamiseks.

Kliendirakenduse kirjutamise hetkel ei olnud serverrakendus veel valmis, seega arendamise etapil kasutati abiklassi, mille meetodid tagastasid varasemalt ettevalmistatud andmeid. Serverrakenduse arendamise käigus oli planeeritud järk-järgult kirjutada üle klassi meetodi sisu, et nad töötaksid koostöös serveriga api kaudu. Abiklass pidi saama täisväärtuslikuks puhvriks suhtlemiseks kliendi- ja serverrakenduse vahel. Andmete formaat abiklassi tagastamisel vastab andmete formaadile, mis on kokku lepitud serverrakenduse arendajaga.

### 4.1 Veebirakenduse disain

Seoses sellega, et antud lõputöö autoril puudub täielikult veebilehtede planeerimise ja kujundamise kogemus, otsustas autor otsida inspiratsiooni olemasolevatel

fännikirjanduse veebipõhistel platvormidel. Kõige enam tõmbas autori tähelepanu uuringu käigus populaarseim osalejate poolt sisestatud veebirakendus Fanfiction.net. Autorile imponeeris idee, et veebilehekülje kujundus meenutab raamatut, kuna lugemise ajal tekkis tunne nagu kasutaja hoiaks käes päris raamatut.

## 4.2 Veebilehekülje peamised komponendid

Arendatava prototüübi leheküljed on visuaalselt ja struktureeritult jagatud mitmeks komponendiks. Joonisel 1 on kujutatud pealeht, mis esimesena kuvatakse külastajale veebirakendusele jõudmisel.

Pealehe komponendid:

- Valge riskülik tähendab konteinerit kogu lehekülje sisule;
- Roheline riskülik tähendab navigeerimispaneeli;
- Kollane riskülik tähendab pealmist lehekülje sisu;
- Sinine riskülik tähendab tähtsaima konteiner-komponenti



Joonis 1 .Veebirakenduse komponendid.



### 4.2.1 Navigeerimispaneel

Lehtede vahel liikudes paikneb navigeerimispaneel alati ühes kohas ning selle struktuur ei muutu, mis mõjutab positiivselt kasutajakogemust. Muutuda võivad ainult parempoolses nurgas asuvad nupud sõltuvalt sellest, kas kasutaja on autoriseeritud või mitte. Autoriseerimata külastaja näeb parempoolses nurgas järgmised valikud: “Koduleht“, “Otsi fännikirjandusi“, “Meist“, “Logi sisse“, “Registreeri”. Autoriseeritud kasutaja “Logi sisse“ ja “Registreeri” nuppude asemel näeb lauset “Tere tulemast,” ja lõpus tema kasutajatunnust, millele vajutades ilmub rullmenüü järgneva valikuga: “Profiil”, “Luu fännikirjandus” ja “Logi välja”.

Navigeerimispaneel on ekraanisuurusele reageeriv ehk *responsive*. Realiseeritud kohandatava ekraani suurus on CSS klasside abil ning Lisas 3 on näide failist Navigation.css.

### 4.2.2 Peamise lehekülje sisu konteiner

Konteiner, kus on peamine lehekülje sisu on märgistatud kollase värviga joonisel 2. See sisaldab enda keskkohas konteinerit SPA sisuga igal üksikul lehel. Lisaks on leheküljel mõeldud suure kasutajate hulga külastuseks, seega selle konteineri külgedel saab seadistada reklaami, et saada rahastust projekti üleval hoidmiseks ja edaspidiseks arendamiseks. Sellisel viisil paigutatud reklaam ei kattu lehekülje põhisisuga, mis paikneb konteineri keskel, kuid samal ajal on kasutajale märgatav.

### 4.2.3 Praeguse lehekülje SPA konteiner

Joonisel 1 kujundatud sinise värviga tähistatud konteineris asub praeguse lehe SPA poolt genereeritud sisu. Iga praeguse SPA alamkomponent kuvatakse selle konteineri piires, seega saab väita, et see on veebisaidi kõige olulisem konteinerikomponent.

```

class FanficMain extends React.Component<FanficMainProps> {

  render() {
    return (
      <div className="FanficMain">
        <div className="fm-container">
          <FanficMainInformation id={this.props.params.fanficId}/>
          <FanficMainChapters fanficId={this.props.params.fanficId}
chapterId={this.props.params.chapterId}/>
        </div>
      </div>
    );
  }
}

```

Joonis 2. Fännikirjanduse pealeht koodis.

Iga *routeris* määratud komponent Joonisel 2 renderdatakse selle komponendi sees vastava lingi klõpsamisel, näiteks fännikirjanduse pealeht lingil domeen/Fanfic/<id>.

#### 4.2.4 Hüpikaken

Loodud prototüübil ilmub hüpikaken antud juhtudel:

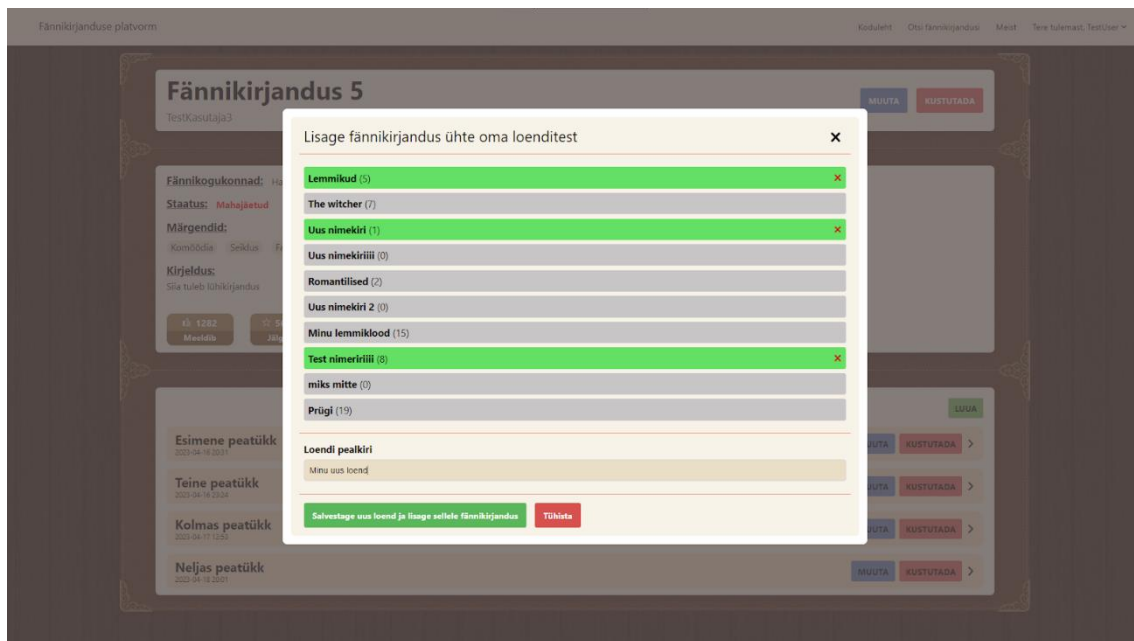
- Veateate kuvamine ja kasutaja suunamine teisele lehele. Näiteks näidates teadet, et fännikirjanduse linki /Fanfic/id-12 ei ole leitud.
- Kasutajalt kinnituse küsimine toimingu sooritamiseks, mida ei saa tagasi võtta. Näiteks kinnituse küsimine fännikirjanduse või peatüki kustutamise kohta.
- Täisfunktsionaalsuse esitamine. Näiteks fännikirjanduse lisamine kasutaja lemmikute nimekirja, mis on rakendatud funktsionaalse pop-up komponendi abil (Joonis 3).

```

<Router>
  <Routes>
    <Route path="/" element={<Home/>}/>
    <Route path="/About" element={<About/>}/>
    <Route path="/Login" element={<Login/>}/>
    <Route path="/Register" element={<Register/>}/>
    <Route path="/Fanfic/Editor/:id?" element={<FanficEditor/>}/>
    <Route path="/Fanfic/:fanficId" element={<FanficMain/>}/>
    <Route
      path="/Fanfic/:fanficId/Chapter/:chapterId?"
      element={<FanficMain/>}/>
    <Route
      path="/Fanfic/:fanficId/Chapter/Editor/:chapterId?"
      element={<ChapterEditor/>}/>
    <Route path="/Search/Fanfic" element={<SearchFanfic/>}/>
    <Route path="/Tests/Lexical" element={<Lexical/>}/>
    <Route path="*" element={<p>There is nothing here: 404!</p>}/>
  </Routes>
</Router>

```

Joonis 3. Teede deklaratsioon MPA töötamiseks.



Joonis 4. Hüplikaken fännikirjanduse lisamiseks loendisse pärast klõpsamist "Loendisse" fännikirjanduse lehel.

### 4.3 Veebirakenduse ülesehitus

Klientrakenduse prototüüp on loodud SPA (Single Page Application) ja MPA (Multi Page Application) struktureerimise tüüpide kombineerimise teel. Veebirakenduse leheküljed, mis ei ole seotud ühise eesmärgiga on ühendatud MPA struktuuriga.

Iga lehekülg on eraldi laaditav ja salvestab state'is ainult vajalikud andmed oma tööks, näiteks fännikirjanduse otsinguleht, kodulehekülg, fännikirjanduse leht, loomise või redigeerimise lehekülg. Veebilehtede vahel liikumine toimub linkide kaudu, mis töötleb ja suunab edasi tänu React Routerile (Joonis 4).

Iga selline leht on omakorda eraldi SPA, mis koosneb veel väiksematest ja teineteisest sõltumatutest komponentidest. Serverrakendusest saadud andmetest lehekülg uueneb ning kirjutab ümber ainult need komponendid, mis sõltuvad saadud andmetest. Näiteks, kui kasutaja vajutab meeldimise nuppu fännikirjanduse leheküljel, ei ole vaja värskendada ja ümber kirjutada tervet lehekülge. Antud juhul toimub ainult "Meeldib" komponendi muutmine, viies selle aktiivsesse olekusse. Sarnase printsiibi järgi töötavad täitmised ja serveripoolne vormide valideerimine lehekülgedel, kommentaaride jätmise, serverrakendusest automaatselt laetavad variandid fännikogukonna või märgendi otsingul fännikirjanduse otsingu leheküljel.

SPA ja MPA kombineerimine lubab kasutada ära mõlema meetodi eeliseid. Tänu täpsele routingule on iga leht eraldiseisev ja täisväärtuslik, millel on parameetreid sisaldav link, mis laseb kasutajal sattuda alati konkreetsele leheküljele välerskendades veebibrauserit või avades linki. Iga lehe esitlemine eraldi SPA-na tagab kasutajale omakorda sujuvat kasutuskogemust, kui kogu lehte ei laadita uuesti, tehes toiminguid selle piires (vormide täitmine ja saatmine, kommentaaride jätmise).

### 4.4 Rakenduse struktuur

Create React App abil valminud prototüüp on edasiseks arendamiseks valmis, sealhulgas on olemas lehekülje mall index.html genereerivate html-komponentide näitamiseks ning index.tsx sobitub TypeScripti rakendamiseks. Reacti rakendused asetsevad ühes DOM komponendis, mille siseselt ei tehta uusi veebibrauseri DOM elemente, vaid Reacti omi, sellele vastavalt redutseerib React Virtual DOM veebilehe lõppsisu. Tänu Reacti

komponentide objekti-olemuse struktuurile on see lahendus märgatavalt sobivam jõudluse poolest.

Rakenduse arenduse käigus lisanduvad vajalikud komponendid, tüüpe määravad liidesed ning abiklassid projekti „src“ kausta. Käesoleva klientrakenduse struktuuris on komponendid alamkaustadesse jaotatud ning grupeeritud vastavalt nende funktsionaalsusele ja kuuluvusele lehekülgedel. Eraldi on kokku koondatud rakendusteüleselt vajaminevad abimeetodid (Utils), andmetüübid, kasutajaliides ja enumid.

Lisas 7 on loodud prototüübi struktuur.

## 4.5 Lexical

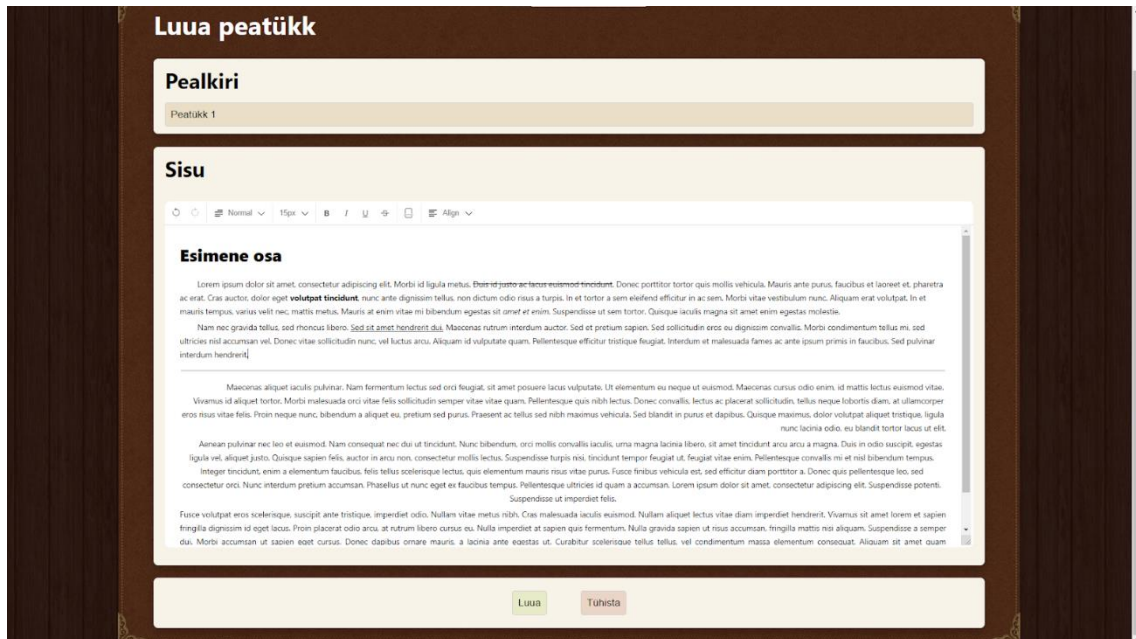
Lexical on mitmekülgne modulaarne tekstiredaktor React raamistiku jaoks. See võimaldab lisada veebilehele tekstiredaktorit koos teksti vormindamise võimalustega. Lexical abil saab lisada pealkirju ja loendeid, muuta teksti suurust, rakendada vormindamist (paks kiri, kaldkirjas, allakriipsutatud, läbikriipsutatud), lisada teksti pilte, tabeleid ja palju muud.

Lexical on tasuta, reklaamivaba tekstiredaktor, mis on loodud Reacti jaoks ning seda levitatakse MIT litsentsi alusel koos kõigi selle moodulite ja pistikprogrammidega.

Lexical tekstiredaktori integreerimise etapid:

1. Loodi eraldi Reacti komponent, mis hoiab endas redaktori olekut, kus seadistatakse redaktori mooduleid.
2. Kirjutati konfiguratsiooni objekt, mis hoiab muutujate kujul vajalikke redaktori seadeid.
3. Imporditi vajalikud moodulid ja plugineid, et mugavalt luua peakangelaste fännikirjandust veebiplatvormil: LexicalRichTextPlugin, LexicalHistoryPlugin, LexicalListPlugin, LexicalHorizontalRulePlugin.
4. Kirjutati menüü plugin, mis kuvab tekstiredaktori kohal võimalikuid vormindamise valikud, ToolbarPlugin.
5. Loodi CSS vormindatud plokkide korrektseks kuvamiseks tekstiredaktoris.

6. Koguti töötav redaktor (Joonisel 5) imporditud pluginatest, mida saab lisada vajalikele lehtedele Reacti komponendina.



Joonis 5 Redaktori väljanägemine lehel.

## 4.6 Kliendi- ja serverrakenduse sidustamine

Kliendirakenduse arendamise hetkel polnud serverit veel välja töötatud, seega kliendi andmete saamist ja saatmist on realiseeritud funktsioonide kaudu, mis tagastavad ettevalmistatud vastused (olgu need siis andmed või kinnitused toimingute sooritamise kohta). Need funktsioonid on ühendatud abistava TypeScript failiga BackendUtils. Tulevikus plaanitakse BackendUtils funktsioonide tööloogikat ümber kirjutada, et need suhtleksid API serveriga, kuid andmete tagastamise ja vastuvõtmise tüüpe ning vormingut ei muudetak. Seega vajab klientserveriga ühendamiseks vaid BackendUtils-i ajakohastamist, ilma et midagi peaks muutma React komponentides, mis kasutavad BackendUtils-i andmeid saamiseks ja saatmiseks. Selleks on kasutatud rangeid tüüpimisi BackendUtils-i funktsioonide parameetrite ja tagastatavate väärtuste osas, kasutades liideseid ja põhitüüpe (String, number jne). Samuti on reaktsioonikomponendid, mis

kasutavad BackendUtils-i funktsioone, kirjutatud arvestades võimalikku asünkroonsust, viivitusi ja vastuse ootamist serverilt.

```
export async function getAllStatuses(): Promise<IStatus[]> { return [  
  {  
    id: "dwdw22113-d1wd21-2331ddnvd1",  
    title: "Protsessis",  
    color: "#16e516"  
  },  
  {  
    id: "dwdw22113-ddawd21-2331ddnvd1",  
    title: "Mahajäetud",  
    color: "red"  
  },  
  {  
    id: "dwdw22113-ddawd21-21ddnvd1",  
    title: "Lõppenud",  
    color: "#c41be3"  
  },  
  {  
    id: "dwdw213-ddawd21-2331ddnvd1",  
    title: "Peatunud",  
    color: "blue"  
  }  
];  
}
```

Joonis 6 Asünkroonne funktsioon BackendUtils.getAllStatuses() tagastab IStatus liidesele kuuluvate objektide nimekirja lubadusena.

```
export default interface IStatus {  
  id:string;  
  title:string;  
  color:string;  
}
```

Joonis 7 IStatus liides defineerib fännilugude staatuse objektides sisalduvate parameetrite omadused.



## 5 Hinnang loodud klientrakendusele

Lõputöö raames valminud klientrakenduse prototüüp täidab kõike funktsionaalseid ja mittefunktsionaalseid nõudeid.

### 5.1 Nõuete täitmine ja kasutatavus

Pärast serverrakenduse ühildamist täidab valminud klientrakendus kõiki funktsionaalseid ja mittefunktsionaalseid nõudeid, mis kehtivad tavakasutajale ehk autorile ja lugejale.

Klientrakenduses on rakendatud loogika ning kõik meetodid ja funktsioonid, mis on vajalikud serverrakendusest saadud andmete töötlemiseks ja kasutajale kuvamiseks, samuti vajalikke andmete edastamiseks serverrakendusse, et kasutaja saaks süsteemiga edukalt töötada. Kliendirakendus võtab arvesse ja töötleb ka olukordi, kui serverrakendusest andmete vastuvõtmine on häiritud või võtab kaua aega.

Kliendirakenduse kood on üles ehitatud nii, et uute funktsioonide lisamiseks ei pea olemasolevaid funktsioone ümber kirjutama. Samuti on lehtedel ja abiklassidel koodis jäetud edaspidiseks arendamiseks ette nähtud funktsioonid, mis lihtsustavad uute plokkide ja moodulite integreerimist.

Kasutajaliides jäi oma ülesehituselt analüüsi käigus tuletatud prototüübile lähedaseks, kuid arendusprotsessi käigus langetati mitmeid kasutajakogemuse optimeerimiseks ja parandamiseks suunatud otsuseid. Näiteks mõned liidese funktsionaalsed elemendid, nagu fännikirjanduse lisamine kasutajaloenditesse, on liikunud üksikutelt lehtedelt funktsionaalsetele hüpikakendele, mis on vähendanud kasutajate üleminekute arvu erinevate lehtede vahel.

Seoses tavakasutajate funktsionaalsete nõuete täitmise keerukusele, püüdes keskenduda detailide ideaalse oleku viimistlemiseni, jäid ajalise resurssi tõttu administraatori kasutajatega seotud nõuded täitmata. Administraatori funktsioonideta saab kliendirakenduse kasutusele võtta ainult tingimusel, et administraatorid teevad kõik vajalikud toimingud serverrakenduse tasandil läbi otseste andmebaasi päringute. Selline lähenemine on vastuvõetav, kuid see ei ole jätkusuutlik ning sobib ainult tehniliselt pädevate administraatorite juuresolekul.

## 5.2 Kasutatud tehnoloogiate uudsus ja sobivus

Tehnoloogiate valimisel võeti arvesse nende pakutud funktsionaalsust, populaarsust, dokumentatsiooni olemasolu, arendamise keerukust, integreerimisvõimalust teiste kasutatavate tehnoloogiatega ja autori kogemusi. Rakenduse tuumaks on arendusraamistik React, mille puhul kasutati arendamise lõpetamise hetkeks kõige uuemat ning stabiilsemat versiooni 18.2.0, koos programmeerimiskeelega TypeScript. Vaatamata sellele, et kõige uuem TypeScripti keele versioon arenduse lõpuks on 5.0.4, arenduse jaoks valiti 4.9.5 versiooni, kuna Reacti raamistik lõputöö kirjutamise hetkel ei toetanud veel TypeScripti viiendat versiooni.

Rikkalik tekstiredaktori loomise teek Lexical koos lisandmoodulitega pakkus võimalust luua mugavat tekstiredaktorit fännikirjanduse kirjutamiseks, mis on klientrakenduse lahutamatu osa. Lexical teek on endiselt arendamisel ja uusi versioone avalikustatakse sageli, kuid klientrakenduse arendamise lõpetamise hetkeks on teegi kõige uuem versioon 0.10.0, mis on ka klientrakenduses kasutatud.

Teiste arenduseks valitud tehnoloogiate jaoks valiti samuti aktuaalseid ja stabiilseid versioone. Tehnoloogiad vastasid kliendirakenduse nõuetele ja vajadustele, tehtud otsustega jäädi rahule.

### 5.3 Kasutatavuse testimine

Loodud klientrakenduse prototüübi kasutatavuse testimise põhieesmärgiks on kontrollida määratud nõuete täitmist, et tõestada kasutajaliidese lihtsust uute külastajate jaoks. Kasutatavuse testimiseks on kolm viisi: kasutatavuse testimine kohapeal, testimine küsitlustega ning kaugtestimine. Kaks viimast meetodit jäid vahele, sest lõputöö autor soovis jälgida prototüübi testijate toimetulekut ülesannetega, mõõta nende täitmiseks kulutatud aega ja murekohtade tekkimisel operatiivselt neid registreerida. Kasutatavuse testimine kohapeal toimus prototüübi arendaja arvutist ning testijateks valiti potentsiaalseid fännikirjandusi veebiplatvormi kasutajaid ehk erineva vanuse ja arvutioskusega eestlasi. Osalejateks said viis inimest sealhulgas ka prototüüplahenduse arendaja ise, kelle ülesandeks oli kasutades lahenduskäigu samme jõuda oodatava tulemuseni. Lisas 6 on välja toodud testülesanded koos sammudega.

Iga testimiskatset läbi viimisel mõõdeti aega ja sisestati saadud andmed tabelisse (Tabel 1). Esimeseks testijaks oli prototüübi arendaja ehk inimene, kes on kõige paremini tuttav testitava funktsionaalsusega. Tema saadud tulemusi võetakse aluseks võrdlemisel teise testijatega, kellele loodud tarkvaralahenduse funktsionaalsus on tundmatu.

	<b>Esimene ülesanne</b>	<b>Teine ülesanne</b>	<b>Kolmas ülesanne</b>	<b>Neljas ülesanne</b>	<b>Viies ülesanne</b>
<b>Esimene testija</b>	6 sekundit	19 sekundit	20 sekundit	32 sekundit	47 sekundit
<b>Teine testija</b>	15 sekundit	26 sekundit	23 sekundit	43 sekundit	53 sekundit
<b>Kolmas ülesanne</b>	12 sekundit	27 sekundit	29 sekundit	49 sekundit	53 sekundit
<b>Neljas testija</b>	10 sekundit	25 sekundit	24 sekundit	37 sekundit	54 sekundit
<b>Viies testija</b>	20 sekundit	29 sekundit	33 sekundit	50 sekundit	58 sekundit

Tabel 1. Ülesannete lahendamisele kulutatud aeg.

Tabelis 1 esitatud andmetest nähtub, et testijad said kõigi ülesannetega oma tempos edukalt hakkama.

## 5.4 Võimalused edasiseks arenduseks

Esiteks on edasiseks arendamiseks vaja ühendada olemasoleva kliendirakendust serverrakendusega ja viia läbi täieliku testimist nende interaktsiooni õigsusele. Serverrakendus valmib käesoleva aasta sügisel, seejärel planeeritakse valmiva fännikirjanduse platvormi avalikustada.

Kliendirakenduse ülesehitusest tulenevalt on selleks vaja muuta üks abiklass, asendades staatilised andmed serverist andmete vastuvõtmise funktsiooniga.

Samuti on vaja võimalikult kiiresti lisada kliendirakendusele administraatorite funktsionaalsuse kasutajaliides. Vajalikud lehed on fännikogukonnad märkide lisamiseks ja haldamiseks, funktsionaalsust kasutajakaebuste vastuvõtmiseks ja töötlemiseks ning liidest kasutajakontode haldamiseks, näiteks võimalus reegleid rikkuvaid kasutajaid kustutada või blokeerida.

Pikaajalise toe ja arendusega saab lisada kasutajaliideses piltide kasutamise võimalus, näiteks võimalus lisada fännikirjanduse tekstidesse illustratsioone, võimalus lisada fännikirjandusele kaaneid ja võimalus lisada kasutajaprofiilile pilte. Samuti saab arendada kliendirakenduse potentsiaali, lisades kasutajaliidesesse keelte vahetamise funktsiooni, lisades vähemalt inglise keele toe rakendusele rahvusvahelise mastaabi andmiseks. Lisaks kasutajaliidese keele vahetamisele, nähakse võimalust lisada tuge fännikirjanduse tõlkimisele erinevatesse keeltesse.

## Kokkuvõte

Lõputöö käigus valmis eestikeelne fännikirjanduse prototüüp. Veebirakenduse arendusprotsess oli jagatud kahe tudengi vahel. Käesoleva lõputöö autor tegeles klientrakenduse loomisega ja kaastudeng Vladislav Stepanenko serverrakenduse loomisega. Paraku ei valminud serverrakendus õigeaegselt.

Alternatiivsete veebirakenduste ja läbiviidud uuringu põhjal said määratud andmed ja nõuded eestlaste seas. Lähtudes nendest lõi autor funktsionaalsust arendataval platvormil. Valmis sai järgmine funktsionaalsus: fännikirjanduse lugemine, lisamine, muutmine ning registreerimisvõimalus. Arendusprotsessi käigus kasutas autor kaasaegsemaid ja kõige sobivamaid tehnoloogilisi vahendeid, mis selgusid nende võrdlemise käigus.

Vaatamata sellele, et fännikirjanduse platvorm ei saanud valmis, hinnatakse käesolevat lõputööd õnnestunuks. Esmaversioon arendatud klientrakendusest on töötav ja valmis sidustamiseks serverrakendusega, mis peab valmis saama käesoleva aasta sügisel. Arendatavale prototüübile määratleti täpsemaid nõuded ning autor kontrollis nõuete täitmist kasutatavuse testimise läbiviimisega. Edasiarenduses planeeritakse lisada funktsionaalsust ja huvipakkuvaid fännikirjandusi, et köita tähelepanu ja aidata massikultuuri levikule.

## Kasutatud kirjandus

- [1] jutulabor, „Sõnaseletused,“ jutulabor, [Võrgumaterjal]. Available: <http://jutulabor.blogspot.com/p/sonaseletused.html>. [Kasutatud 4 Märts 2023].
- [2] Roo, „4 Ways Fanfiction Writing Can Boost Your Creativity,“ medium, 25 Mai 2020. [Võrgumaterjal]. Available: <https://medium.com/@nomadicblog2019/4-ways-fanfiction-writing-can-boost-your-creativity-ed7069234644>. [Kasutatud 16 Märts 2023].
- [3] H. Lopez, „Best 13 Fanfiction Sites of All Time,“ epubor, 12 Märts 2023. [Võrgumaterjal]. Available: <https://www.epubor.com/best-fanfiction-sites-of-all-time.html>. [Kasutatud 22 Aprill 2023].
- [4] raulpage, „STRESSI TEEJUHT - Kuidas saada lahti liigsest pingest?,“ raulpage, [Võrgumaterjal]. Available: <http://raulpage.org/stress/teke110.html>. [Kasutatud 9 Märts 2023].
- [5] Lokyatha, „How Stress is Eating Into Your Creativity and How to Beat It?,“ Lokyatha, [Võrgumaterjal]. Available: <https://www.lokyatha.com/library/stress-creativity-relation>. [Kasutatud 9 Märts 2023].
- [6] Lääne Elu, „Juhtkiri: kas lapsed ei loe?,“ Lääne Elu, 1 Juuni 2022. [Võrgumaterjal]. Available: <https://online.le.ee/2022/06/01/juhtkiri-kas-lapsed-ei-loe/>. [Kasutatud 20 Märts 2023].
- [7] „massikultuur,“ 2006. [Võrgumaterjal]. Available: <http://entsyklopeedia.ee/artikkel/massikultuur2>. [Kasutatud 2 Aprill 2023].
- [8] archiveofourown, „About the OTW,“ archiveofourown, [Võrgumaterjal]. Available: <https://archiveofourown.org/about>. [Kasutatud 12 Aprill 2023].
- [9] codecrux, „What is the best platform to develop a web application?,“ codecrux, [Võrgumaterjal]. Available: <https://codecrux.com/blog/what-is-the-best-platform-to-develop-a-web-application.html>. [Kasutatud 13 Märts 2023].
- [10] B. Li, „Bitbucket vs GitHub: Which Code Repository Is Better for Your Development Projects?,“ kinsta, 17 Aprill 2023. [Võrgumaterjal]. Available: <https://kinsta.com/blog/bitbucket-vs-github/>. [Kasutatud 4 Mai 2023].
- [11] K. Klaus, „Lapsed eelistavad raamatutele multimeediat,“ postimees, 10 Veebruar 2007. [Võrgumaterjal]. Available: <https://parnu.postimees.ee/2146729/lapsed-eelistavad-raamatutele-multimeediat>. [Kasutatud 22 Märts 2023].
- [12] A. Kisliak, „Why Bitbucket repository is a great tool for professional dev teams,“ polontech, 10 Juuli 2019. [Võrgumaterjal]. Available: <https://polontech.com/blog/bitbucket-team/>. [Kasutatud 14 Märts 2023].
- [13] J. Juviler, „What is GitHub?,“ hubspot, Oktoober 31 2022. [Võrgumaterjal]. Available: <https://blog.hubspot.com/website/what-is-github-used-for#:~:text=It%27s%20used%20for%20storing%2C%20tracking,collaborate%2C%20and%20pitch%20their%20work>. [Kasutatud 7 Märts 2023].
- [14] J. Inouye, „WebStorm vs VS Code: Compare top IDEs,“ techrepublic, 9 Juuni 2022. [Võrgumaterjal]. Available: <https://www.techrepublic.com/article/webstorm-vs-vscode/>. [Kasutatud 5 Mai 2023].

- [15] visualstudio, „Why VS Code?“, visualstudio, [Võrgumaterjal]. Available: <https://code.visualstudio.com/learn#:~:text=Visual%20Studio%20Code%20is%20a,%2B%2B%2C%20JavaScript%2C%20and%20more>. [Kasutatud 2 Mai 2023].
- [16] visualstudio, „Why did we build Visual Studio Code?“, visualstudio, [Võrgumaterjal]. Available: <https://code.visualstudio.com/docs/editor/whyvscode#:~:text=Edit%2C%20build%2C%20and%20debug%20with,selection%2C%20snippets%2C%20and%20more>. [Kasutatud 5 Mai 2023].
- [17] Internet Matters Team, „What is Wattpad? A breakdown for parents“, internetmatters, 12 Juuli 2022. [Võrgumaterjal]. Available: <https://www.internetmatters.org/hub/news-blogs/what-is-wattpad-a-breakdown-for-parents/>. [Kasutatud 12 Aprill 2023].
- [18] jutulabor, „Sõnaseletused“, jutulabor, [Võrgumaterjal]. Available: <http://jutulabor.blogspot.com/p/sonaseletused.html>. [Kasutatud 4 Märts 2023].
- [19] Põllumajanduse Registrate ja Informatsiooni Amet, „Klaviatuuriga navigeerimine“, Põllumajanduse Registrate ja Informatsiooni Amet, 2 Veebruar 2020. [Võrgumaterjal]. Available: <https://www.pria.ee/juurdepaasetavus>. [Kasutatud 7 Aprill 2023].
- [20] idalko, „Bitbucket vs. GitHub: 7 Reasons Why Bitbucket Wins“, idalko, [Võrgumaterjal]. Available: <https://www.idalko.com/bitbucket-vs-github/>. [Kasutatud 4 Mai 2023].
- [21] Interaction Design: Beyond Human- Computer Interaction. Yvonne Rogers, Helen Sharp, Jenny Preece The Design of Everyday Things -Donald A. Norman Emotional Design: Why We Love (or Hate) Everyday Things -Donald A. Norman

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

Mina, Kristina Revizor

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

15.05.2023

---

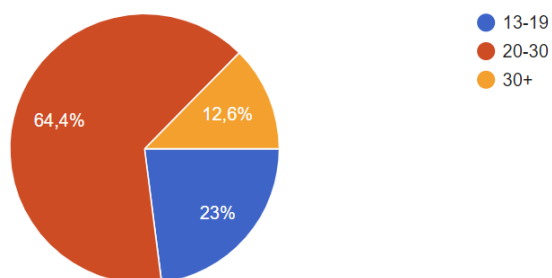
<sup>1</sup> Lihtlitsents ei kehti juurdepääsupiirangu kehtivuse ajal vastavalt üliõpilase taotlusele lõputööle juurdepääsupiirangu kehtestamiseks, mis on allkirjastatud teaduskonna dekaani poolt, välja arvatud ülikooli õigus lõputööd reprodutseerida üksnes säilitamise eesmärgil. Kui lõputöö on loonud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



## Lisa 2 – Uuringu Tulemused

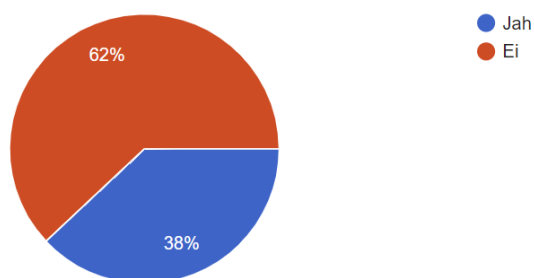
Kui vana te olete?

116 vastust



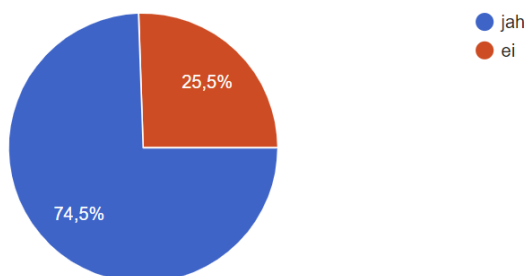
Kas te loete fännikirjandust? (Kui vastuseks on ei, palun mõlege, mis funktsionaalsust sooviksite näha fännikirjanduse platvormil)

116 vastust



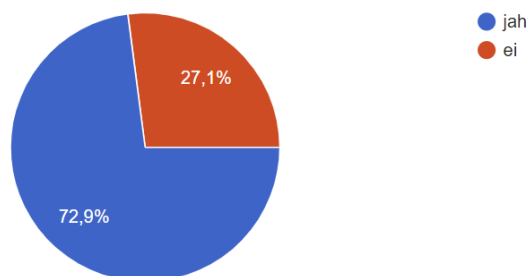
Kas Teile on tähtis sisseehitatud otsingumootor, mis lihtsustaks fännikirjanduse otsimist?

116 vastust



Kas fännikirjanduse valimise käigus on Teie jaoks oluline näha selle lühikirjeldust?

116 vastust

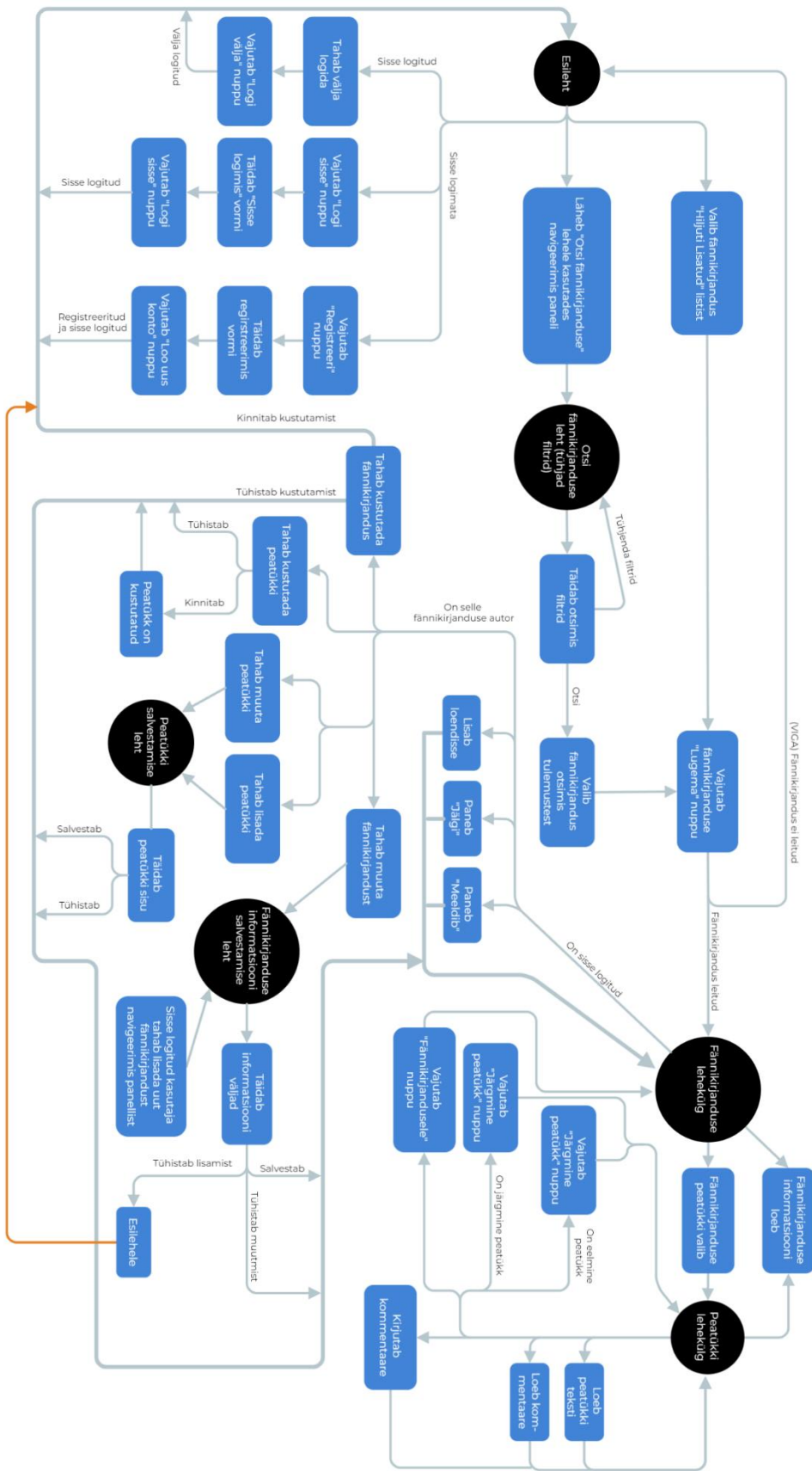


## Lisa 3 - Näide CSS adapteeritud ekraani suuruse kohta, failist Navigation.css.

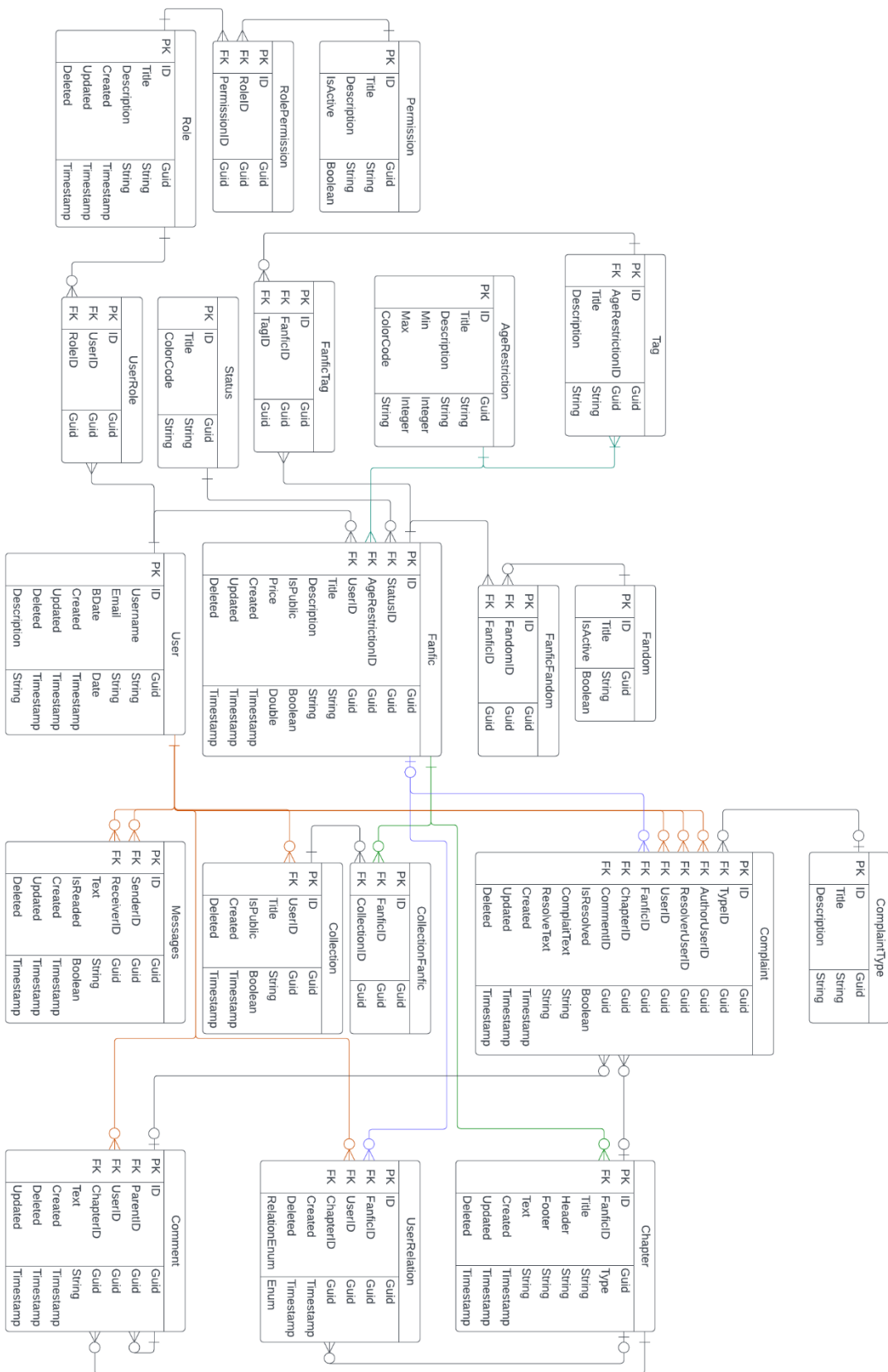
```
/* Mobile menu */
.nav-menu {
  display: flex;
  flex-wrap: wrap;
  justify-content: space-between; align-items: center;
  margin-top: 0;
}
.logo {
  font-size: 20px;
  padding: 7.5px 10px 7.5px 0; }
.logo a {
  width: max-content;
}
.nav-item {
  padding: 10px;
  order: 3;
  width: 100%;
  text-align: center;
  display: none;
}
/* Desktop menu */
@media all and (min-width: 850px) {
  .nav-menu {
    justify-content: center; align-items: flex-start; flex-wrap: nowrap;
    background: none;
  }
  .logo {
    flex: 1;
    order: 0;
  }
  .nav-item {
    order: 1;
```

```
position: relative;
display: block;
width: auto;
}
}
```

# Lisa 4 - Infosüsteemi skeem



# Lisa 5 - Tuleviku fännikirjanduse platvormi andmebaas



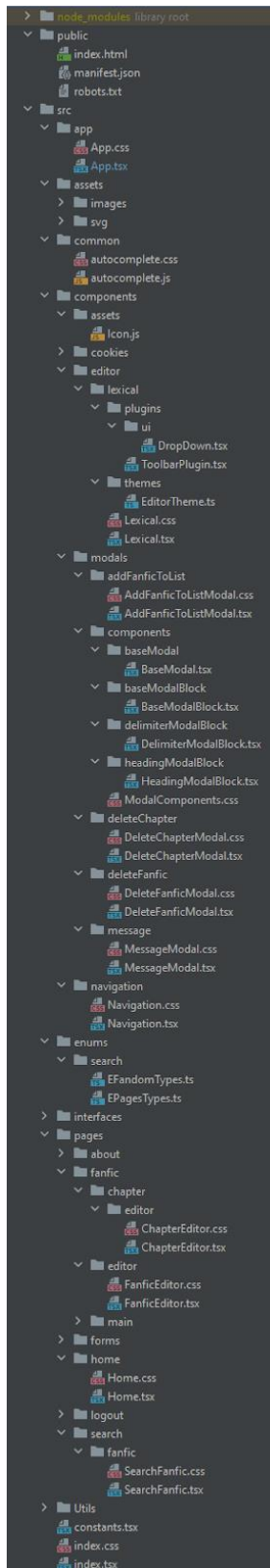
## Lisa 6 - Kasutatavuse testülesanded

No.	Sammud	Oodatav tulemus
1	<ol style="list-style-type: none"> <li>1. Pealehel vajutada "Lugema" nuppu peale viienda fännikirjanduse juures.</li> <li>2. Olles viienda fännikirjanduse lehel vajutada "Esimene peatükk".</li> </ol>	<ol style="list-style-type: none"> <li>3. Avamine viienda fännikirjanduse esimese peatükki.</li> </ol>
2	<ol style="list-style-type: none"> <li>1. Pealehel vajutada "Registreeri" nuppu peale.</li> <li>2. Täida registreerimise vormi väljad: kasutajatunnus, e-post, salasõna.</li> <li>3. Vajutada "Loo uus konto" nuppu peale.</li> <li>4. Pealehel vajutada "Lugema" nuppu peale viienda fännikirjanduse juures</li> <li>5. Olles viienda fännikirjanduse lehel vajutada "Meeldib".</li> </ol>	<ol style="list-style-type: none"> <li>6. Meeldida viienda peatüki, et nupp "Meeldib" muutus roheliseks.</li> </ol>
3	<ol style="list-style-type: none"> <li>1. Pealehel vajutada "Otsi fännikirjandusi" nuppu peale</li> <li>2. Parameetris "Märgendid" sisestada "komöödia" ja "draama".</li> <li>3. Vajutada "Otsi" nuppu peale</li> </ol>	<ol style="list-style-type: none"> <li>4. Rakendades üht parameetrit saada otsingutulemuses 4 fännikirjandust</li> <li>5. Leida fännikirjandust ühe parameetri järgi.</li> </ol>
4	<ol style="list-style-type: none"> <li>1. Pealehel vajutada "Otsi fännikirjandusi" nuppu peale</li> <li>2. Olles otsingumootori vahelehel leida parameeter Märgendid ja sisestada ajalugu</li> <li>3. Leia parameter Lehekülgede arv, vali valin ise ja sisesta vahemikus 2 kuni 19</li> <li>4. Vajutada "Otsi" nuppu peale</li> </ol>	<ol style="list-style-type: none"> <li>5. Leia fännikirjandust ühe parameetri järgi.</li> </ol>

5	<ol style="list-style-type: none"> <li>1. Pealehel vajutada "Logi sisse" nuppu peale</li> <li>2. Täida registreerimise vormi väljad: kasutajatunnus: Test1, parool: Test1234</li> <li>3. navigeerimispaneelis vajutada rippmenüü peale ja avanevast valikutest valida Luua fännikirjandusi</li> <li>4. Fännikirjanduse loomise lehel</li> </ol>	5. Leida, kuidas saab luua fännikirjandust
---	---	--



## Lisa 7 – Loodud prototüübi struktuur



## **Lisa 8 – Klientrakendusele link**

<https://bitbucket.org/krreworkspace/fanfiction/src/master/>