

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Morten Tšinakov 213114IAIB

**VEEBIRAKENDUS LÜHIANIMAFILMIDE SÜŽEE
ARENDAMISEKS**

Bakalaureusetöö

Juhendaja: Siim Rebane
BSc

Tallinn 2025

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: Morten Tšinakov

04.06.2025

Annotatsioon

Käesoleva bakalaureusetöö eesmärgiks on luua veebirakendus, mis võimaldaks animafilmitegijatel võimalikult mugavalt välja arendada filmisüžeesid. Lõputöö raames tehakse koostööd kliendiga ning antud vallas tegutsevate filmitegijatega, kes rakendust arendamise käigus katsetavad.

Lõpptulemusena valmib eesrakendusest, tagarakendusest ja andmebaasist koosnev prototüüp, mis võimaldab kasutajatel kirja panna ideid, neid mugavalt kohandada, ümberjärjestada ning filtreerida. Samuti on võimalik loodud projekte teiste kasutajatega jagada ning saada neilt tagasisidet. Rakendus pakub ka võimalust kirjutada loodud süžee ümber standarditele vastavaks stsenaariumiks ning see PDF-failina alla laadida.

Eesrakendus on kirjutatud Reactis, tagarakendus Spring Bootis ning andmete hoiustamiseks kasutatakse PostgreSQL andmebaasi. Kasutaja suhtleb eesrakendusega, mis omakorda saadab ja küsib andmeid tagarakenduselt. Tagarakendus suhtleb andmebaasiga.

Kasutajale kuvatakse andmeid viisil, mis võiks lühifilmisüžeede loomist hõlbustada: filmitegijal on andmeid võimalik näha väiksemate järjestatud tükkidena, tervikliku süžeenähtena ning ajajoonena.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 39 leheküljel, 6 peatükki ja 22 joonist.

Abstract

A Web Application for Plot Development in Short Animated Films

The goal of this bachelor's thesis is to create a web application that enables animation filmmakers develop storylines for their films. The application is developed in cooperation with a client and filmmakers active in the field, who will test the application during its development.

The result will be a prototype consisting of a frontend application, backend application and database, which allows users to write down ideas and conveniently modify, reorder and filter them. It's also possible to share created projects with other users and receive feedback. Finally, the application offers the ability to convert a created storyline into a standard-compliant screenplay and download it as a PDF file.

The frontend application is written in React, the backend in Spring Boot and PostgreSQL is used for data storage. The user interacts with the frontend, which in turn sends and requests data from the backend. The backend communicates with the database.

The data is presented to the user in a way that would make the creation of short film storylines as convenient as possible: the filmmaker can view the data as smaller ordered fragments, as a complete storyline, and as a timeline.

The thesis is written in Estonian and is 39 pages long, including 6 chapters and 22 figures.

Lühendite ja mõistete sõnastik

ACID	Omadused, mis tagavad andmebaasi terviklikkuse vigade esinemise korral (<i>Atomicity, consistency, isolation, durability</i>)
API	Rakendusliides (<i>Application Programming Interface</i>)
Boilerplate	Kood, mida kasutatakse muudatusteta erinevates rakenduse osades
CSS	Keel veebilehtede kujunduse märgendamiseks (<i>Cascading Style Sheets</i>)
DOM	HTML-i dokumentidega suhtlemise liides (<i>Document Object Model</i>)
HTML	Keel veebilehtede märgendamiseks (<i>HyperText Markup Language</i>)
HTTP	Protokoll teabe edastamiseks arvutivõrkudes (<i>Hypertext Transfer Protocol</i>)
HTTPS	Protokoll turvaliseks teabe edastamiseks arvutivõrkudes (<i>Hypertext Transfer Protocol Secure</i>)
Java	Programmeerimiskeel
JavaScript	Programmeerimiskeel
JSON	Andmevahetusformaad (<i>JavaScript Object Notation</i>)
JWT	Standard kasutaja turvaliseks autentimiseks (<i>JSON Web Token</i>)
HTTP-küpsis	HTTP päringuga kindlale domeenile saadetak tekstiplokk (<i>HTTP cookie</i>)
MIME tüüp	Interneti-standard, mida kasutatakse erinevate failide sisu kirjeldamiseks (<i>Multipurpose Internet Mail Extension type</i>)
PDF	Arvuti tark- ja riistvaraplatvormist sõltumatu elektrooniliste dokumentide vorming (<i>Portable Document Format</i>)
PostgreSQL	Relatsioonilise andmebaasi haldamise süsteem
React	JavaScripti raamistik kasutajaliideste loomiseks
Spring Boot	Java raamistik veebirakenduste arendamiseks
SQL	Relatsiooniliste andmebaaside jaoks loodud päringukeel (<i>Structured Query Language</i>)

SVG	Formaat vektorgraafika kirjeldamiseks (<i>Scalable Vector Graphics</i>)
XSS	Murdskriptimine, veebirakendustele suunatud rünnak (<i>Cross-site scripting</i>)

Sisukord

1	Sissejuhatus	10
2	Nõutud funktsionaalsused ja olemasolevad lahendused	11
2.1	Nõutud funktsionaalsused	11
2.1.1	Süžee arendamine ja stsenaariumi kirjutamine	11
2.1.2	Filmi kestuse arvutamine	11
2.1.3	Ideede filtreerimine	12
2.1.4	Stseenide agregatsioon	12
2.1.5	Süžee jagamine teiste kasutajatega	13
2.1.6	Piltide lisamine	13
2.2	Potentsiaalsed olemasolevad lahendused	13
2.2.1	Final Draft	14
2.2.2	Fade In	14
2.2.3	Celtx	15
2.2.4	WriterDuet	15
2.2.5	Arc Stutio Screenwriting	16
3	Tehnoloogiate valik	17
3.1	Eesrakenduse tehnoloogiad	17
3.2	Tagarakenduse tehnoloogiad	18
3.3	Andmebaasi tehnoloogiad	18
4	Implementeeritud funktsionaalsused	20
4.1	Autentimine	20
4.2	Projektid	23
4.3	Fragmendid	24
4.3.1	Fragmentide loomine	25
4.3.2	Fragmentide tirimine	27
4.3.3	Fragmentide muutmine ja kustutamine	29
4.3.4	Fragmentide detailne vaade	30
4.3.5	Fragmentide agregatsioon	30
4.4	Sildid ja fragmentide filtreerimine	31
4.4.1	Siltide lisamine	32
4.4.2	Ajajoon	33
4.5	Pildid	34

4.6	Projektide jagamine teiste kasutajatega	35
4.6.1	Kasutajate otsing	36
4.6.2	Teavitused	37
4.7	Stsenaariumi kirjutamine	38
4.7.1	Stsenaariumi formaat	39
4.7.2	Stsenaariumi kirjutamise vaade	40
4.7.3	Klahvikombinatsioonid	41
4.7.4	Stsenaariumi eksportimine	42
5	Tulemused	45
5.1	Loodud rakendus	45
5.2	Kasutajate tagasiside	46
5.3	Edasiarenduse võimalused	47
6	Kokkuvõte	49
	Kasutatud kirjandus	50
	Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks	52
	Lisa 2 – Andmebaasi tabelite struktuur	53

Jooniste loetelu

1	<i>Sisselogimine</i>	20
2	<i>Õnnestunud päring JWT-ga autentimisel</i>	21
3	<i>Aegunud JWT-ga päringu tegemine</i>	22
4	<i>Projektide vaade</i>	23
5	<i>Uue projekti loomisel kuvatav vaade</i>	24
6	<i>Fragmentide loend koos uue fragmendi loomisel avaneva külgpaneeliga</i>	26
7	<i>Fragmendi väljad külgpaneelis ning andmete põhjal genereeritud fragmendi komponent</i>	26
8	<i>Uue fragmendi tirimine fragmentide loendisse</i>	27
9	<i>Külgpaneel fragmendi info muutmiseks</i>	29
10	<i>Fragmentide detailne vaade</i>	30
11	<i>Agregeeritud fragmendid (vasakul paneelis)</i>	31
12	<i>Sildi loomise vaade</i>	32
13	<i>Fragmentide filtreerimine</i>	33
14	<i>Siltide põhjal loodud ajajoon</i>	33
15	<i>Fragmendile pildi joonistamine</i>	34
16	<i>Kasutajaga jagatud projekti vaade</i>	36
17	<i>Stsenaariumi kirjutamise vaade</i>	40
18	<i>Stsenaariumi eksportimisel avanev paneel</i>	43
19	<i>Andmebaasidiagramm</i>	53
20	<i>Kasutajaga seostuvad tabelid</i>	54
21	<i>Projektidega seostuvad tabelid</i>	55
22	<i>Fragmentidega seostuvad tabelid</i>	56

1. Sissejuhatus

Filmitegemise esimeseks osaks on filmi süžee loomine ning selle kirjutamine stsenaariumiks. Korralik stsenaarium on eelduseks filmile rahalise toetuse taotlemisel[1] ning vajalik filmi edukaks tootmiseks. Kuna stsenaariumile kehtivad üsna kindlad vormistatusnõuded[2], siis enne selle kirjutamist võiks filmi süžee juba suuresti paigas olla. Süžee arendamine ning ideedega katsetamine on stsenaariumi kirjutamise faasis tülikas ning aeganõudev.

Enne käesoleva lõputööga alustamist filmitegijatega rääkides selgus, et üldiselt kasutavad nad süžee arendamiseks üht programmi ja stsenaariumi kirjutamiseks teist. Lisaks sellele on kummaski faasis kasutataval tarkvaral olulisi puudusi, sest süžee arendamiseks puuduvad spetsiifiliselt filmitegijate vajadusi arvesse võtvad rakendused ning stsenaariumi kirjutamiseks loodud rakendused on tihti kas täielikult tasulised või pakuvad tasuta versiooni piiratud funktsionaalsustega.

Lõputöö eesmärgiks on luua prototüüp veebirakendusele, mis pakub nii süžee arendamiseks kasulikke funktsionaalsusi kui stsenaariumi vormistamise võimalust. Arvestades, et filmi planeerimisel kulub suur osa ajast just esimesele neist etappidest ning hetkel puuduvad taolised rakendused, mis peaks silmas filmitegijate soove ja vajadusi, siis keskendutakse lõputöö tegemisel eriti süžee loomiseks vajalike funktsionaalsuste arendamisele. Stsenaariumi kirjutamist käsitletakse kui süžee vormistamist lõplikuks dokumendiks ning lõputöö skooopi lisatakse vaid kõige tarvilikumad funktsionaalsused stsenaariumi kirjutamiseks.

Rakenduse arendamisel tehakse koostööd Horvaatia animafilmistuudio Adriatic Animationiga[3], kes pakkus välja algse idee. Algse idee ja nii Adriatic Animationis kui mujal väiksemates stuudiotest töötavate filmitegijate soovide ning tagasiside põhjal valmib rakendus, mis võtab arvesse lühianimafilme tegevate režissööride vajadusi filmide arendamisel.

Rakendus võimaldab kirja panna ideid või stseene, neid mugavalt ümber järjestada, filtreerida ja ühtseks tervikuks koondada. Tehtud tööd on võimalik teiste kasutajatega jagada ning saada neilt tagasisidet. Kuna rakendus on suunatud eelkõige animafilmitegijatele, siis võimaldab rakendus ka lihtsamate sketšide joonistamist. Lõpuks on võimalik süžee ümber kirjutada standarditele vastavaks stsenaariumiks ning see PDF dokumendina alla laadida.

2. Nõutud funktsionaalsused ja olemasolevad lahendused

Käesolevas peatükis uuritakse, kas on olemas rakendusi, mis kataks ära funktsionaalsused, mille järgi filmitegijatel loo arendamiseks vajadus on. Esimeses alapealtükis defineeritakse nõutud funktsionaalsused ning teises alapealtükis käsitletakse rakendusi, mida võiks potentsiaalselt probleemi lahendamiseks kasutada.

2.1 Nõutud funktsionaalsused

Filmitegijatega suheldes said defineeritud funktsionaalsused, mis rakendusel võiks olla, et loo arendamine toimuks võimalikult mugavalt. Selles alapeatükis tuuakse välja need funktsionaalsused ning seletatakse, miks need antud probleemi lahendamiseks vajalikud on.

2.1.1 Süžee arendamine ja stsenaariumi kirjutamine

Autorifilmid, sealhulgas lühianimafilmid, toodetakse tavaliselt riigi poolt pakutavate toetuste abil. Toetuse taotlemiseks on muuhulgas vajalik stsenaariumi olemasolu.

Kuna stsenaariumil on üsna kindlalt paika pandud vormistusnõuded, siis jaguneb stsenaariumi kirjutamine üldiselt kahe etappi: süžee arendamine ja stsenaariumi vormistamine. Seda sellepärast, et süžee arendamine on tihti üsna kaootiline protsess, kus filmitegija kogub ideid, kirjutab ja järjestab neid ümber ning eemaldab või asendab olemasolevaid uutega. Selle protsessiga üheaegselt stsenaariumi vormistusnõudeid jälgida on ebamugav ja tüütu.

Rakendus peakski koosnema kahest eraldi osast - vaatest, kus oleks võimalik mugavalt lugu välja arendada ning vaatest, kus saaks antud loo stsenaariumiks ümber vormistada.

2.1.2 Filmi kestuse arvutamine

Stsenaristidele on teada kirjutamata reegel, et üks stsenaariumi lehekülg võrdub ühe minuti filmiga. Animafilmirežissööride kogemuse põhjal võib öelda, et animafilmide puhul ei pea see reegel kahjuks paika ning ei ole seega väga usaldusväärne meetod filmi kestuse arvutamiseks.

Filmi kestuse hindamine on väga oluline, sest kestus mängib suurt rolli filmi eelarve koostamisel. Animaafilmide tootmine koosneb paljudest eri osadest nagu animeerimine, vahekaadrite joonistamine, ümberjoonistamine, värvimine jne. Mida pikem on film, seda rohkem tuleb kõiki neist tegevustest teha ning seda kallimaks film läheb.

Nagu mainitud, on üks võimalus filmi kestuse hindamiseks stenaariumi lehekülgede kokkulugemine, kuid animaafilmide puhul see meetod enamasti ei tööta. Teine probleem stsenaariumi põhjal filmi kestuse hindamisega on see, et filmi kestuse saab paika alles pärast stsenaariumi kirjutamist. Paremini oleks filmi eeldatavat kestust jälgida juba loo arendamise ajal, sest seda vähem on hiljem tööd loo ja stsenaariumi ümberkirjutamisega kui lugu peaks liiga pikaks venima.

Teine võimalus on lugu oma peas läbi mängida ning samal ajal aega mõõta. See lähenemine on aga tülikas juhul kui süžee peaks muutuma. Iga muudatuse korral tuleks kestus uuesti välja arvutada. Samuti peab aja mõõtmiseks terve lugu juba kirjas olema ning jooksvalt kestuse jälgimine pole võimalik.

Kõike seda arvesse võttes tundus nii stuudio Adriatic Animation produtsendile kui filmitegijatele, et rakenduses võiks olla võimalus igale ideele või stseenile määrata kestus ning rakendus võiks nende põhjal kokku arvutada kogukestuse. Nii saab kestust jooksvalt jälgida ning ideede eemaldamisel või ühe idee kestuse muutumisel ei pea filmitegija terve filmi kestust uuesti välja arvutama.

2.1.3 Ideede filtreerimine

Tavapäraseks võtteks loo arendamisel on keskendumine ühele tegelasele või süžeele ning unustada ülejäänud lugu hetkeks ära. Et see oleks mugav, peaks saama ideid millegi alusel filtreerida. Samas see, mille alusel stseene või ideid filtreerida võiks jääda filmitegija enda otsustada.

2.1.4 Stseenide agregatsioon

Süžee loomine on kõige lihtsam väikeste tükkide (ideede või stseenide) kaup. See võimaldab mugavat ideede lisamist, eemaldamist, muutmist, ümberjärjestamist ja filtreerimist. Samas on film siiski nendest tükkidest koosnev tervik.

Loo arendamise mugavdamiseks peaks filmitegijal olema võimalus loodud süžeed lugeda ühtse tervikuna. Rakendus peab olema võimeline kirjapandud ideed õiges järjekorras

kokku koguma ning need kasutajale kuvama võttes arvesse ka hetkel rakendatavaid filtreid.

2.1.5 Süžee jagamine teiste kasutajatega

Kuigi tavaliselt kirjutab süžee üks inimene - eriti autorifilmide puhul - on kasulik esitleda oma ideid vahepeal ka teistele. Nii on võimalik enne stsenaariumi kirjutamist saada tagasisidet ning uusi mõtteid.

Rakendus peaks võimaldama kasutajal oma projekte jagada teiste kasutajatega ning teistel kasutajatel peaks olema võimalik lisada kommentaare. Samas ei tohiks teine kasutaja saada süžeed ise muuta.

2.1.6 Piltide lisamine

Tihti on mõnda ideed lihtsam kirjeldada piltide kui sõnade abil, seda eriti animafilmi puhul. Filmi arendamise käigus on tavapärane koguda referents-pilte taustadest, stiilist, värvipalettidest jms.

Rakendus peaks võimaldama piltide lisamist stseenidele kahel viisil: arvutist üles laadides ning ise joonistades. Filmitegijad soovisid, et oleks võimalik joonistada lihtsamaid sketše ilma, et selleks peaks avama eraldi programmi, sealt joonistatud pilti eksportima ja seejärel üles laadima.

2.2 Potentsiaalsed olemasolevad lahendused

Selles alapealtükis analüüsitakse juba olemasolevaid potentsiaalseid lahendusi arvestades eelmises alapealtükis kirjeldatud soovitud funktsionaalsusi.

Filmitegijatega rääkides selgus, et suur osa neist kasutab üht rakendust süžee loomiseks ja teist rakendust stsenaariumi kirjutamiseks. Süžee loomisel kasutatakse erinevaid märkmete tegemiseks mõeldud rakendusi, Microsoft Wordi või kirjutatakse ideed üles märkmepaberitele, mida hiljem käsitsi ümberjärjestatakse. Stsenaariumi kirjutamiseks kasutatakse spetsiaalset tarkvara.

Selles peatükis jäetakse tähelepanuta nii märkmete tegemiseks mõeldud tarkvara kui Microsoft Word, sest oluline nõue oli, et rakendus võimaldaks nii süžee arendamise osa kui stsenaariumi kirjutamise osa. Need rakendused aga stsenaariumi kirjutamise võimalust ei paku. Lisaks sellele ei võimalda nad osasid teisi funktsionaalsusi nagu kestuse arvutamine,

stseenide agregatsioon või ümberjärjestamine, filtreerimine või sketšide joonistamine.

Potentsiaalseteks olemasolevateks lahendusteks on stsenaariumikirjutamise rakendused, mis pakuvad visandamise (inglise keeles *outlining*) võimalust. Visandamine tähendab töötamist väikeste tükkidega, millele on võimalik kirja panna ideid, mida on võimalik ümberjärjestada, muuta ja kustutada. Kuigi need rakendused keskenduvad esmajoonel stsenaariumite kirjutamise protsessi võimalikult mugavaks muutmisele, pakuvad osad neist ka visandamise võimalust ning on potentsiaalselt lahenduseks antud probleemile.

Järgnevates alapeatükkides analüüsitakse erinevate visandamise võimalust pakkuvate stsenaariumikirjutamise rakenduste sobivust antud probleemi lahendamiseks võttes arvesse eelmises peatükis kirjeldatud funktsionaalsusi. Kõik need rakendused pakuvad stsenaariumi automaatset vormindamist, koostöö võimalust ja palju muud. Stsenaariumi kirjutamise osast katavad nad ära kõik nõuded ja pakuvad lisaks veel palju võimalusi, seega keskendutakse neid analüüsides just süžee arendamise osale.

2.2.1 Final Draft

Final Draft[4] pakub loo planeerimise võimalust. Loo planeerimiseks luuakse tühi vaade, kuhu saab ükshaaval hakata lisama kaarte, mida nimetatakse inglise keeles *beat*-iks. Selline kaart koosneb pealkirjast ja kirjeldusest ning kaarte saab omavahel siduda. Samuti saab neid lisada ajajoonele ning määrata neile eeldatava pikkuse. Pikkust arvestatakse aga lehekülgedes mitte ajaühikutes.

Final Draft katab ära päris suure osa nõudmistest. Funktsionaalsused, mida Final Draft ei võimalda on piltide joonistamine ja kestuse arvutamine. Samuti ei leidnud autor infot kaartide filtreerimise kohta.

Kuigi Final Draft on väga võimekas rakendus, mis pakub mugavat stsenaariumi kirjutamise võimalust, ei sobi see antud probleemi lahendamise peamiselt selle tõttu, et Final Draft ei paku tasuta versiooni ning tasulised versioonid on üpris kallid. Samuti jääb osa soovitud funktsionaalsustest katmata.

2.2.2 Fade In

Fade In[5] pakub võimalust kuvada stsenaariumi osasid kaartidena, mida saab muuta, eemaldada ja ümber järjestada. Kaardid on üks-ühele seotud stsenaariumiga.

Fade In ei paku kestuse arvutamist, piltide joonistamist ega kaartide filtreerimist. Kaartidele saab määrata küll värve, kuid mitte silte ning värvide alusel otseselt midagi muuta ei saa. Lisaks ilmnesid programmi katsetamise käigus mõned vead. Näiteks ilmus parema hiireklõpsuga nähtavale tulnud menüü vales kohas ning kaarte ümber järjestades jooksis programm kokku.

Fade In on tasuline, kuid pakub demo-versiooni. Demo-versioon lisab stsenaariumile vesimärgi ning kui on kirjutatud üle 10 lehekülje hüppab teatud intervalli tagant lahti aken, mis meenutab, et tegu on tasuta versiooniga.

2.2.3 Celtx

Celtx[6] pakub võimalust lisada, muuta ja eemaldada kaarte. Kaarte saab lisada suvalisse kohta ekraanil ja Celtx järjestab need vasakult paremale ning ülalt alla. Kuna aga kaarte saab lisada suvalisse kohta ekraanil, peab ühe kaardi ümberjärjestamisel käsitisi muutma ka teiste kaartide asukohti, et need ei ilmuks üksteise peal.

Celtx ei paku võimalust lisada pilte, defineerida kestust ega kaartide filtreerimist. Tervikliku süžee lugemiseks peab iga kord kaarte stsenaariumiks eksportima ja siis neid stsenaariumi vaates lugema.

Celtx on tasuline tarkvara, kuid pakub tasuta versiooni. Tasuta versioon aga kaartide vaadet ei sisalda ning kasutaja saab luua ainult ühe projekti. Isegi tasulise versiooni kõige odavamas variandis on projektide arv piiratud.

2.2.4 WriterDuet

WriterDuet pakub võimalust lisada, muuta ja eemaldada kaarte. Samuti saab neid ümber järjestada ning grupeerida lisades neile silte. Siltide alusel saab kaarte ka filtreerida. Kaardid on otseselt seotud stsenaariumiga - iga kaart vastab ühele stseenile stsenaariumis. See tähendab, et üks kaart ei saa vastata osale stseenist või mitmele stseenile korraga.

Kuigi WriterDuet on tasuline, pakutakse tasuta versiooni, kus projektide arv on piiratud kolme projektini. WriterDuet ei paku võimalust piltide joonistamiseks või kestuse arvutamiseks. Kuna loodud kaardid on seotud stsenaariumi vaatega, siis stsenaariumi kirjutamise käigus tuleb need kas asendada või peab lugema stsenaariumit ja süžeed läbisegi.

2.2.5 Arc Studio Screenwriting

Nagu eelnevalt käsitletud stsenaariumi kirjutamise rakendused pakub ka Arc Studio[7] võimalust lisada, muuta, ümber järjestada ja eemaldada kaarte. Kaarte saab siltide alusel grupeerida ja filtreerida. Iga kaart on stsenaariumiga seotud, kuid võib sisaldada mitmeid ideid või stseene.

Arc Studio katab ära suure osa nõutud funktsionaalsustest, välja arvatud kestuse arvutamine ja piltide joonistamine.

Arc Studio on tasuline tarkvara, kuid pakutakse tasuta versiooni. Tasuta versioon visandamist ei sisalda, PDF failid eksporditakse vesimärgiga ning projektide arv on piiratud kahe projektini.

3. Tehnoloogiate valik

Võttes arvesse eelmises peatükis ära toodud nõudeid, valiti välja tehnoloogiad, mida rakenduse tegemiseks kasutati. Kuna üks rakenduse funktsionaalsustest on projektide jagamine teiste kasutajatega ning tähtis on, et rakendus oleks operatsioonisüsteemist sõltumatu, otsustati, et tegemist peaks olema veebirakendusega.

Rakendus koosneb kolmest osast: eesrakendus, tagarakendus ja andmebaas. Kasutaja suhtleb eesrakendusega ning eesrakendus suhtleb omakorda tagarakendusega. Vajalikud andmed salvestatakse andmebaasi. Eesrakenduse loomiseks kasutati JavaScripti ja React raamistikku, tagarakenduse jaoks Javat koos raamistikuga Spring Boot ning andmebaasisüsteemiks valiti PostgreSQL.

3.1 Eesrakenduse tehnoloogiad

Eesrakenduse tehnoloogiaks valiti React ning programmeerimiskeeleks JavaScript. React on vabavaraline teek kasutajaliideste arendamiseks.

React loodi Facebooki poolt ning sellest on saanud üks kõige populaarsemaid teeke kasutajaliideste loomiseks. Teegil on korralik dokumentatsioon[8] ning kuna ta on väga populaarne, leidub suurel hulgal materjale, mis rakenduse arendamisel abiks on. Samuti on Reacti jaoks loodud väga palju teeke, mis rakenduse arendamise lihtsamaks teevad.

React võimaldab luua korduvkasutatavaid komponente ning komponente uuendatakse ning joonistatakse uuesti ekraanile ainult siis kui komponendi olek muutub[9]. See tähendab, et rakendus on kiire, sest iga muudatuse järel ei uuendata kogu ekraanil olevat infot.

Komponentide disainimiseks kasutati CSS märgistuskeelt.

Eesrakenduses kasutati ka järgnevaid teeke:

Axios[10]	HTTP päringute tegemiseks
Dnd-kit[11]	Komponentide liigutamiseks hiirega tirimise teel
Lodash[12]	Päringute ärajätmiseks kui nende vaheline intervall on liiga väike (inglise keeles <i>debounce</i>)
Perfect-freehand[13]	Digitaalseks joonistamiseks
React-error-boundary[14]	Koodivigade püüdmiseks

React-icons[15]	Ikoonide lisamiseks
Slate-react[16]	Stsenaariumi kirjutamiseks
Zustand[17]	Rakenduse oleku hoiustamiseks

3.2 Tagarakenduse tehnoloogiad

Tagarakenduse loomiseks valiti Java programmeerimiskeel ning raamistikuks Spring Boot. Spring Boot on väga populaarne raamistik serveripoolsete rakenduste loomiseks. See võimaldab rakenduse automaatset seadistamist vastavalt teekidele, mida rakenduses kasutatakse hoides nii kokku arendaja aega. Iga Spring Booti rakendus sisaldab eelseadistatud serverit, mis võimaldab rakendust käivitada ilma välise serveri seadistamiseta.

Spring Bootil on väga korralik dokumentatsioon[18], mis sisaldab näiteid, saadaolevate teekide loetelu ja palju muud. Tänu tema populaarsusele on saadaval suur hulk kogukonna poolt loodud materjale, mis aitavad programmeerijat Spring Booti rakenduse arendamisel. Samuti on olemas arvukalt tööriistu tagarakenduse loomiseks.

Tagarakenduses kasutati ka järgnevaid teeke:

Apache PDFBox[19]	PDF failide loomiseks
Apache Tika[20]	Piltide salvestamisel MIME tüüpide valideerimiseks
Flyway[21]	andmebaasi skeemi lisamiseks ja haldamiseks
ImgScalr[22]	Piltide mõõtmete muutmiseks
JWT[23]	JWT loomiseks ja valideerimiseks
JUnit[24]	Rakenduse testimiseks
Lombok[25]	<i>Boilerplate</i> koodi automaatseks genereerimiseks
MapStruct[26]	Java objektide konverteerimiseks
Mockito[27]	Koodi testimiseks
OpenAPI[28]	API dokumentatsiooni automaatseks genereerimiseks

3.3 Andmebaasi tehnoloogiad

Andmebaasisüsteemiks valiti PostgreSQL. PostgreSQL on vabavaraline andmebaasisüsteem, millel on spetsiaalne PostgreSQL litsents, mis on üsna sarnane Apache 2.0 litsentsiga ning seab seega väga vähe piiranguid süsteemi kasutajale.

PostgreSQL on võimas ja stabiilne andmebaasisüsteem, millel on väga põhjalik dokumentatsioon ning mis vastab SQL standarditele ning ACID põhimõtetele.

PostgreSQLi jaoks on loodud suurel hulgal laiendusi ja rakendusi, mis täiustavad selle funktsionaalsusi. Ühte neist - pg_trgm - on kasutatud ka antud rakenduses kasutajate otsimiseks ligilähedase nime järgi.

Samuti saab PostgreSQLis luua erinevaid indekseid[29], näiteks GiST (*Generalized Search Tree*), BRIN (*Block Range Indexes*) või GIN (*Generalized Inverted Index*), millest viimast on kasutatud ka antud rakenduses, kombineerides seda juba mainitud pg_trgm laiendusega.

4. Implementeeritud funktsionaalsused

Selles peatükis räägitakse lähemalt implementeeritud funktsionaalsustest. Selguse huvides püüab autor funktsionaalsustest rääkida sellises järjekorras nagu rakenduse kasutaja nendega kokku puutub alustades sisselogimisest ja lõpetades stsenaariumi eksportimisega. Kuna kasutajaliides on antud projektis tähtsal kohal, lisatakse alapeatükkide juurde ka kuvatõmmiseid, et lugeja saaks parema arusaama rakenduse toimimisest.

4.1 Autentimine

Rakenduse kasutamiseks peab looma kasutajakonto. Kasutajakonto tegemiseks peab sisestama e-maili, ees- ja perekonnanime ning parooli. Sisestatud andmed saadetakse tagarakendusele ning kui antud meiliaadressiga kasutajat juba ei eksisteeri, siis salvestatakse info andmebaasi ning kasutaja logitakse sisse.

Kui kasutajal on juba konto, siis saab sisse logida sisestades meiliaadressi ja parooli.

Nii kasutajakonto loomisel kui juba eksisteeriva kontoga sisselogimisel genereeritakse tagarakenduse poolt JWT[30] ehk *JSON Web Token* ja *refresh token*. JWT on JSON vormingus dokument, mis sisaldab vajalikke kasutaja andmeid, kehtivusaega ja infot krüpteerimisalgoritmi kohta ning mis krüpteeritakse salajase võtmega. Salajase võtme valedesse kättesse sattumise vältimiseks genereeritakse see tagarakenduses automaatselt rakenduse käivitamisel ning seda hoitakse tagarakendust jooksvatava masina mälus. Joonisel 1 kujutatakse sisselogimisel tehtavaid samme ees- ja tagarakenduses.

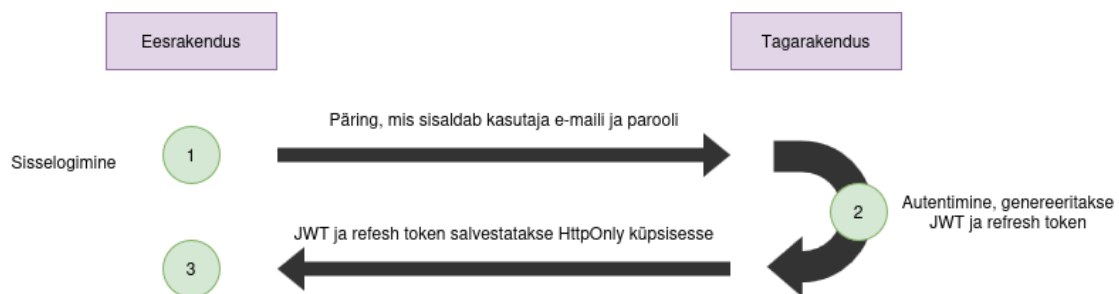


Figure 1. *Sisselogimine*

Pärast parooliga sisselogimist toimub järgnevate päringute tegemisel kasutaja autentimine JWT kaudu, et minimeerida pöördumisi andmebaasi poole. Joonisel 2 kujutatakse õnnestunud päringut tagarakendusele.

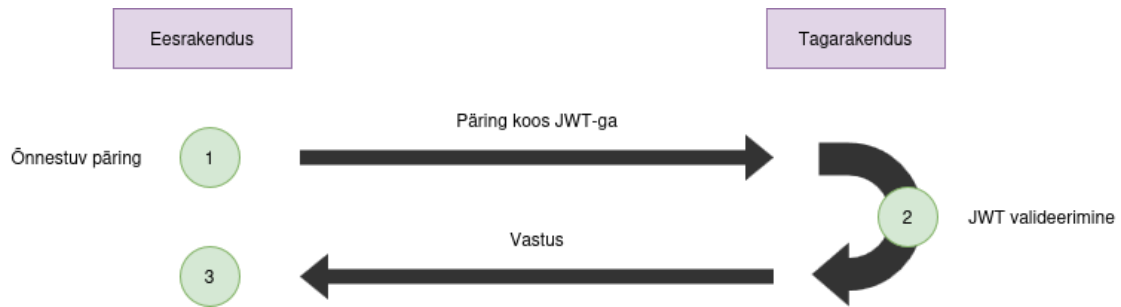


Figure 2. *Õnnestunud päring JWT-ga autentimisel*

Kuna tagarakenduses kontrollitakse ainult seda, kas JWT on krüpteeritud õige võtmega ja on endiselt kehtiv, siis autentimiseks piisab sellest, et kasutajal on valideerne JWT. See aga tähendab ka seda, et kui keegi teine peaks saama ligipääsu kasutaja JWT-le, siis ei takista miski teda autentimast antud kasutajana. JWT-i lekkimise ja sellest tulenevate kahjude vastu on rakenduses kasutusele võetud kaks võtet: JWT-i hoiustamine *HttpOnly*[31] HTTP-küpsises ja JWT-i lühike kehtivusaeg.

JWT-i kehtivusajaks on antud rakenduses 15 minutit - kui keegi teine peaks saama enda käsutusse kasutaja JWT-i, siis saab ta kahju teha maksimaalselt 15 minutit. *HttpOnly* küpsis on HTTP-küpsis, millele lisatakse päringu pealdisesse spetsiaalne väli, mis keelab ligipääsu küpsisele kliendirakenduse kaudu. See aitab XSS- ehk *Cross-site scripting* rünnakute vastu.

JWT-i lühikese kehtivusaja tõttu tekib kasutaja jaoks ebamugavus: kui JWT-i kehtivus läbi saab, peab kasutaja uuesti sisse logima. Selle vältimiseks on kasutusele võetud *refresh token*. *Refresh token* on juhuslikult genereeritud unikaalne märkide jada, mis salvestatakse andmebaasi ning mida kasutatakse uue JWT-i genereerimiseks. Selle asemel, et kasutaja peaks iga 15 minuti tagant uuesti parooliga sisse logima, saadetakse ebaõnnestunud päringu korral kliendirakendusest automaatselt spetsiaalne päring JWT-i uuendamiseks. Tagarakendus võrdleb eesrakenduse saadetud *refresh token*-it andmebaasis olevaga ning kui need klappivad, siis genereeritakse uus 15 minutit kehtiv JWT. Nii JWT kui *refresh token* salvestatakse *HttpOnly* küpsisesse. Pärast JWT-i edukat uuendamist, tehakse algne päring automaatselt uuesti. Joonisel 3 kujutatakse samme, mis tehakse ees- ja tagarakenduses aegunud JWT-ga päringu tegemisel.

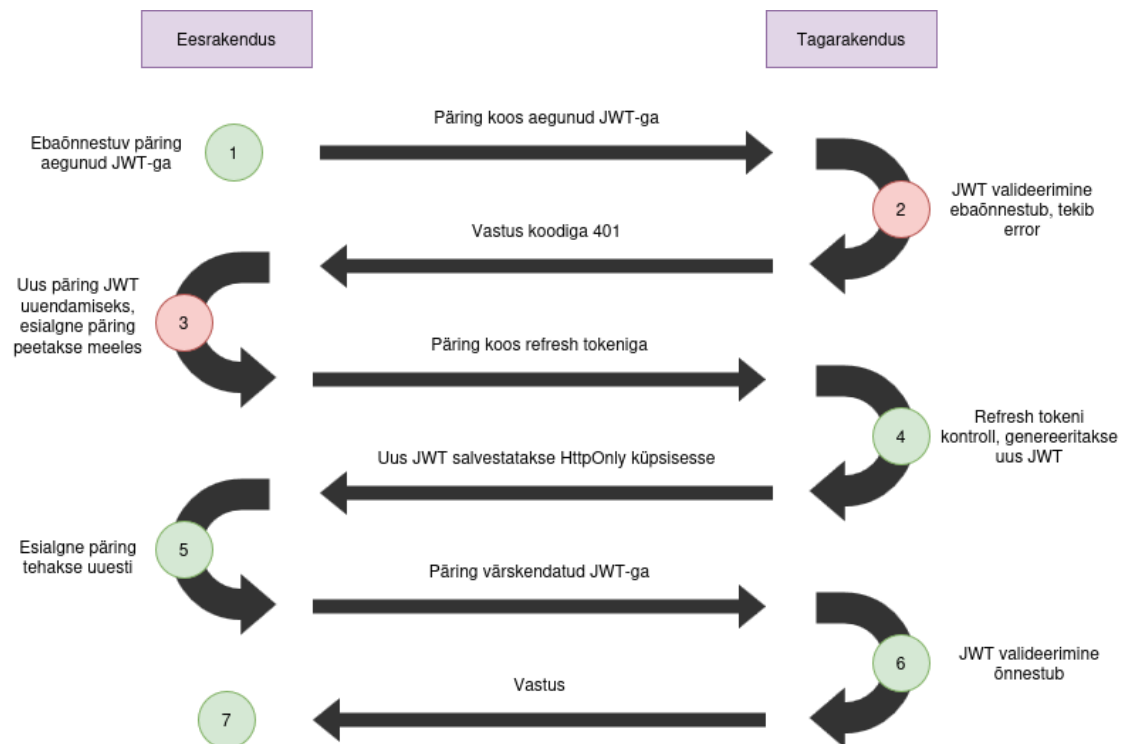


Figure 3. Aegunud JWT-ga päringu tegemine

Refresh token võimaldab seega kasutajakogemuse mugavamaks muutmist ning annab tagarakendusele võimaluse kasutaja ligipääsu teatud määral piirata - kuigi JWT kehtib ka (maksimaalselt) 15 minutit pärast *refresh token*-i eemaldamist andmebaasist, on uute JWT-de loomine *refresh token*-i kustutamisel võimatu ilma uuesti sisse logimata. Piiramata ajaga JWT-de puhul ei ole võimalik ei kasutaja ega kasutaja JWT-i enda valdusesse saanud pahatahtliku inimese juurdepääsu rakendusele võimalik piirata ilma tagarakenduses hoitava salajase võtme uuendamiseta. Võtme uuendamine tähendaks aga seda, et kõik kasutajad peavad uuesti sisse logima.

Kasutaja välja logimisel või JWT-i värskendamise ebaõnnestumisel kustutatakse JWT küpsistest ning *refresh token* nii küpsistest kui andmebaasist. JWT-i värskendamine ebaõnnestub sel juhul kui *refresh token*-i kehtivusaeg on läbi saanud. *Refresh token* kehtib antud rakenduses ühe nädala.

Tagarakenduses kasutatakse kasutaja autentimiseks Spring Security[32] raamistikku. Spring Security pakub paljusid erinevaid funktsionaalsusi, mis arendaja töö lihtsamaks teevad, nende hulgas näiteks paroolide kodeerimist, turvafiltreid ja kontekstide manageerimist. JWT-i ja *refresh-token*-iga autentimist Spring Security ei paku, seega JWT-de ja *refresh-token*-ite genereerimine ja valideerimine tuli autoril implementeerida. JWT-ide genereerimiseks ja valideerimiseks kasutati Jjwt teegi abi.

Eesrakenduses kasutatakse päringute tegemiseks Axiose teeki. Axios lihtsustab eesrakenduse päringute tegemist tagarakendusele ning võimaldab päringuid vahelt lõigata (inglise keeles *intercept*). Antud rakenduses kasutatakse vaheltlõikajat JWT-i aegumise tõttu ebaõnnestunud päringute vahelt lõikamiseks, JWT-i uuendamiseks ning siis esialgse päringu uuesti saatmiseks.

4.2 Projektid

Kui kasutaja on sisse loginud, suunatakse ta projektide vaatesse. Selles vaates kuvatakse iga projekti kohta peamine informatsioon nagu pealkiri ja kirjeldus. Projektid jagunevad kaheks: kasutaja projektid ning kasutajaga jagatud projektid. Projektide vaates saab valida, milliseid neist hetkel kuvatakse vajutades vastava nupu peale. Kasutajaga jagatud projektidest räägitakse hilisemas alapeatükis. Joonisel 4 on kujutatud sisselogimisel kuvatavat projektide vaadet.

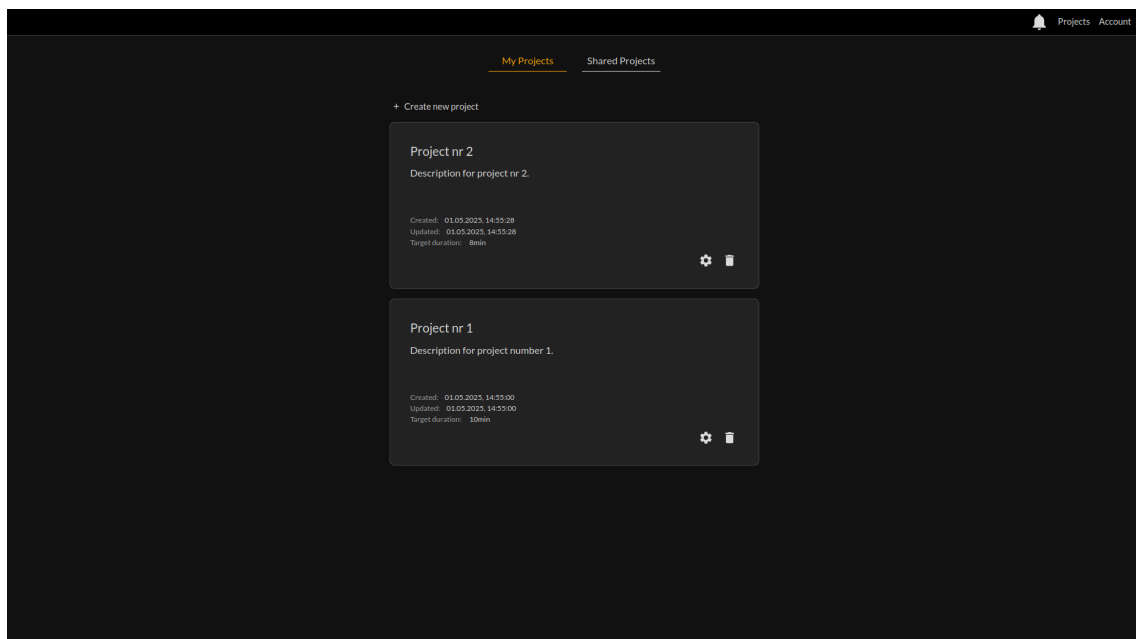


Figure 4. *Projektide vaade*

Iga projekt kuvatakse eraldi komponendina, millele vajutades projekt avaneb. Lisaks sellele on komponendil nupud säteade muutmiseks ja projekti kustutamiseks.

Uue projekti loomiseks tuleb vajutada vastaval nupul projektivaate ülaosas, mis avab komponendi uue projekti loomiseks. Projekti loomiseks peaks kasutaja määrama projektile pealkirja ning valikuliselt ka kirjelduse ja oodatava kestuse. Pärast seda võib projekti salvestada: sisestatud andmed saadetakse tagarakendusele ning projekt salvestatakse andmebaasi. Projekti salvestamisel avatakse automaatselt just loodud projekt. Joonisel 5

on kujutatud uue projekti loomisel avanavat komponenti andmete sisestamiseks.

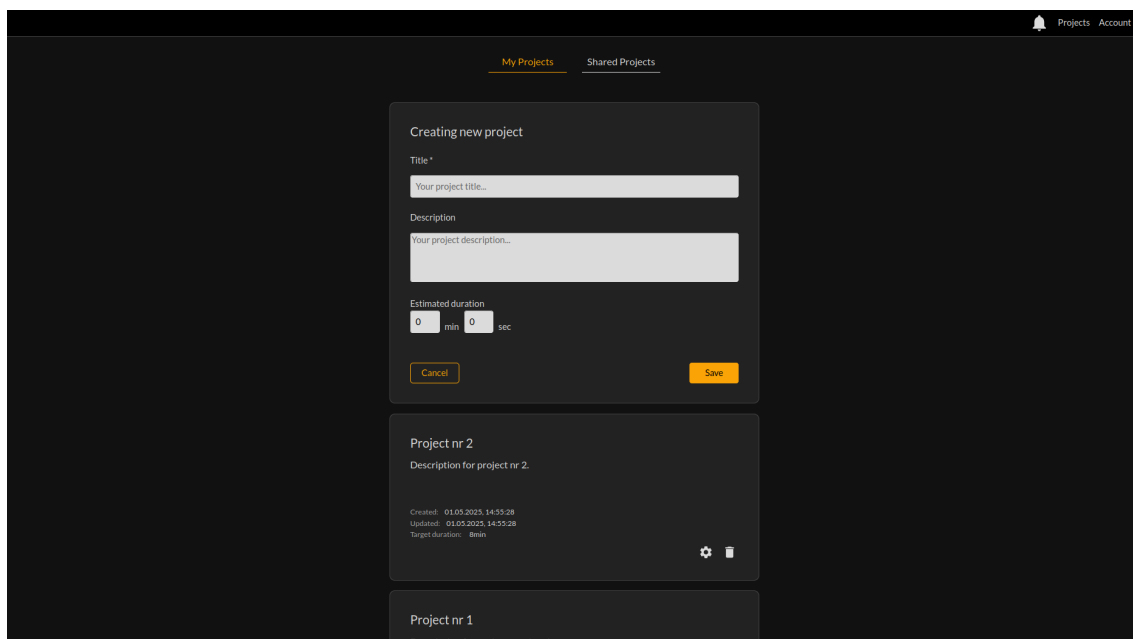
The screenshot shows a dark-themed web interface. At the top, there are tabs for 'My Projects' and 'Shared Projects'. A modal window titled 'Creating new project' is centered. It contains three input fields: 'Title*' with a placeholder 'Your project title...', 'Description' with a placeholder 'Your project description...', and 'Estimated duration' with two numeric input fields for 'min' and 'sec', both currently showing '0'. Below these fields are 'Cancel' and 'Save' buttons. Under the modal, there are two project cards. The top card is titled 'Project nr 2' and contains the text 'Description for project nr 2.' Below this, it lists 'Created: 01.05.2025, 14:55:28', 'Updated: 01.05.2025, 14:55:28', and 'Target duration: 8min'. To the right of this text are icons for settings and deletion. The bottom card is titled 'Project nr 1' and contains the text 'Description for project number 1.'

Figure 5. Uue projekti loomisel kuvatav vaade

4.3 Fragmendid

Kuigi filmi süžee on üks tervik, kaasnevad selle ühtse tervikuna hoiustamise ja kuvamisega teatud probleemid. Arendamise käigus süžee pidevalt muutub ja täieneb ning seda üldiselt ei kirjutata alustades esimesest stseenist ning lõpetades viimasega. Uusi ideid lisatakse tihti süžee keskele ning ideid järjestatakse ümber. Kui vaadelda ja hoiustada süžeed arendamise käigus ühtse tervikuna, on ideede ümberjärjestamine kasutaja jaoks ebamugav ning süžee uuendamine ebaefektiivne, sest iga muudatuse järel tuleb terve süžee asendada uue versiooniga. Samuti raskendab selline lähenemine ideede filtreerimist ning kestuse arvutamist. Mõlemad neist on funktsionaalsused, mida filmitegijad soovisid rakenduses näha.

Süžeed otsustati hoiustada väiksemate tükkidena, mis seotakse projektiga ning asetatakse kasutaja poolt soovitud järjekorda. Nii on ideede ümberjärjestamine, filtreerimine ja süžeele kestuse arvutamine mugavam. Selliseid tükke nimetatakse antud rakenduses fragmentideks ning need võivad kujutada endast ideed, stseeni, mitut stseeni korraga või mõnda muud väiksemat tükki kogusüžee.

Igal fragmendil on lühikirjeldus, detailne kirjeldus, kestus sekundites ning väli, mis näitab kas fragment on ajajoonel või mitte.

Lühikirjeldus võtab kokku fragmendi sisu ning aitab anda ülevaate, millise fragmendiga on tegu ilma, et peaks lugema läbi tervet fragmenti. See on abiks näiteks fragmentide ümberjärjestamisel. Samuti saab nii kiire ülevaate tervest süžeesst ning aukudest süžees. Lühikirjelduse lisamine ei ole kohustuslik. Näiteks võib ilma lühikirjelduseta fragment tähistada auku süžees, mis hiljem oleks vaja ära täita.

Detailne kirjeldus annab idee edasi nii põhjalikult kui võimalik. Detailsed kirjeldused moodustavadki kokku filmi süžee.

Fragmendi kestus on oluline süžee kogukestuse arvutamiseks. Ühe fragmendi kestust on filmitegijal lihtsam hinnata kui terve filmi kestust, eriti kui fragmente pidevalt juurde lisada, muuta või kustutada. Rakendus arvutab kogukestuse kokku ajajoonel olevate fragmentide kestuste kaudu.

4.3.1 Fragmentide loomine

Fragmentide loomiseks kaaluti kolme lähenemist. Esimene neist lisaks uue fragmendi fragmentide loendi lõppu. Sellist fragmentide lisamise viisi oleks kõige lihtsam implementeerida, kuid see oleks kasutaja jaoks ebamugav, sest tihti on vajalik fragmente lisada hoopis loendi keskele. Sel juhul toimuks kaks operatsiooni ning seega ka kaks päringut tagarakendusele: fragmendi salvestamine ning fragmendi liigutamine ühelt positsioonilt teisele.

Teise lähenemisena kaaluti igale fragmendile nupu lisamist, millele vajutades lisatakse uus fragment antud fragmendi järele. Implementeerimise mõttes oleks see enam-vähem sama keeruline kui esimene variant, kuid piisaks ainult ühest päringust tagarakendusele. Kasutajale oleks aga selline lähenemine pigem ebaintuiitivne. Pealegi peaks sel juhul looma kaks eri moodust fragmendi lisamiseks, sest täiesti esimest fragmenti poleks võimalik sel juhul lisada.

Otsustati kolmanda lähenemise kasuks, kus kasutajale luuakse eesrakenduses sisestatud info põhjal fragment, kuid see salvestatakse andmebaasi alles seejärel kui kasutaja on fragmendi tirinud positsioonile kuhu ta seda salvestada tahab. Nii toimub ainult üks päring tagarakendusele, kuid kasutajakogemus on palju intuitiivsem kui teise variandi puhul. Järgnevalt seletatakse täpsemalt, kuidas uue fragmendi lisamine rakenduses välja näeb.

Uue fragmendi loomiseks peab kasutaja vajutama vastaval nupul, misjärel avaneb külgpaneel, mis sisaldab nelja lehekülge andmete sisestamiseks: lühikirjeldus, detailne kirjeldus, kestus ning sildid. Siltidest räägitakse eraldi peatükis. Joonisel 6 on kujutatud fragmentide

loendit koos uue fragmendi loomisel avaneva külgpaneeliga paremas ekraani servas.

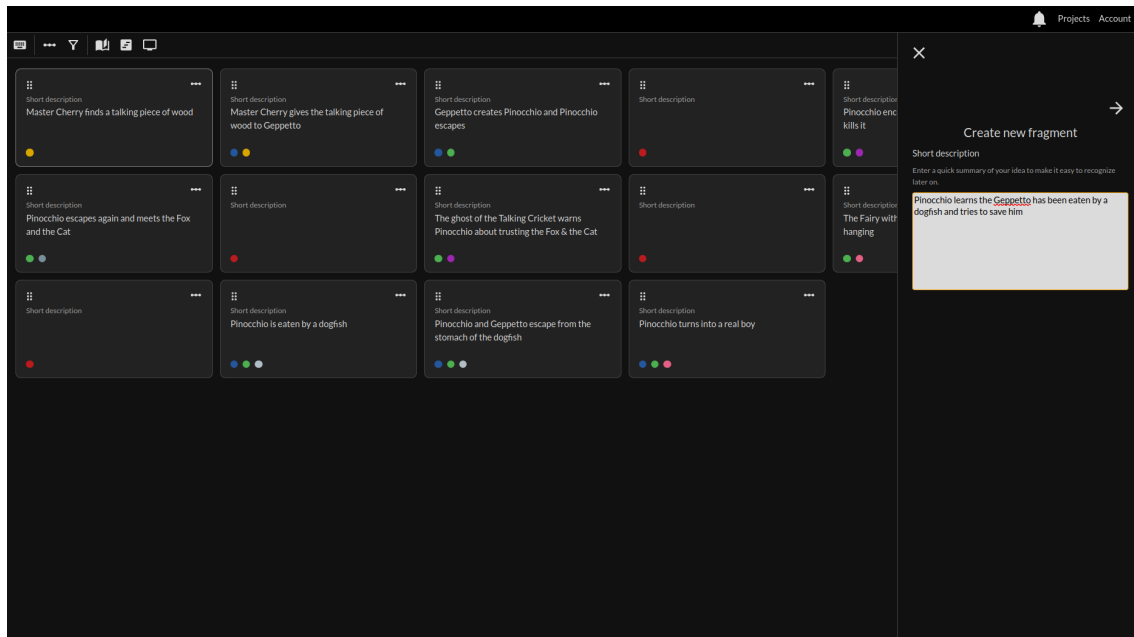


Figure 6. *Fragmentide loend koos uue fragmendi loomisel avaneva külgpaneeliga*

Fragmendi loomiseks peaks ära täitma vajalikud väljad ning sisestatud info põhjal luuakse fragmendi komponent, mille kasutaja saab hiirega fragmentide loendis soovitud kohale tirida. Sisestatud info saadetakse tagarakendusele pärast seda kui fragment on tiritud soovitud positsioonile loendis ning seal lahti lastud. Tagarakendus salvestab andmed andmebaasi. Joonisel 7 on näidatud kõiki nelja akent, kuhu kasutaja saab kirjutada fragmenti puudutava informatsiooni. Viimases aknas on näha informatsiooni põhjal loodud fragmendi komponenti.

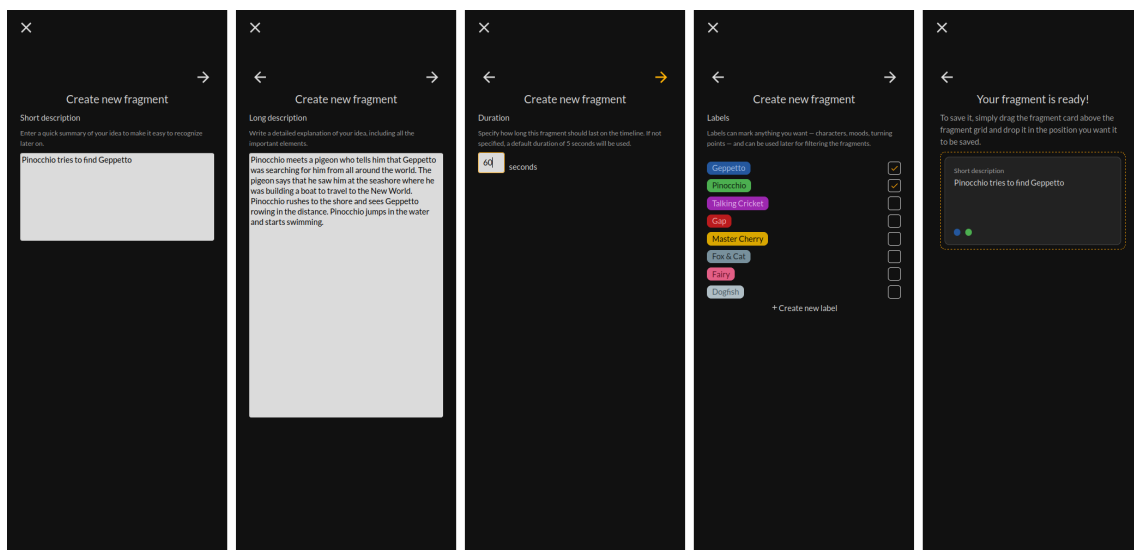


Figure 7. *Fragmendi väljad külgpaneelis ning andmete põhjal genereeritud fragmendi komponent*

Joonisel 8 on näidatud just loodud fragmendi tirimist fragmentide loendisse soovitud kohale. Tirimise ajal külgpaneel sulgub, kuid avaneb uuesti kui fragmendi loendisse asetamine peaks mingil põhjusel ebaõnnestuma.

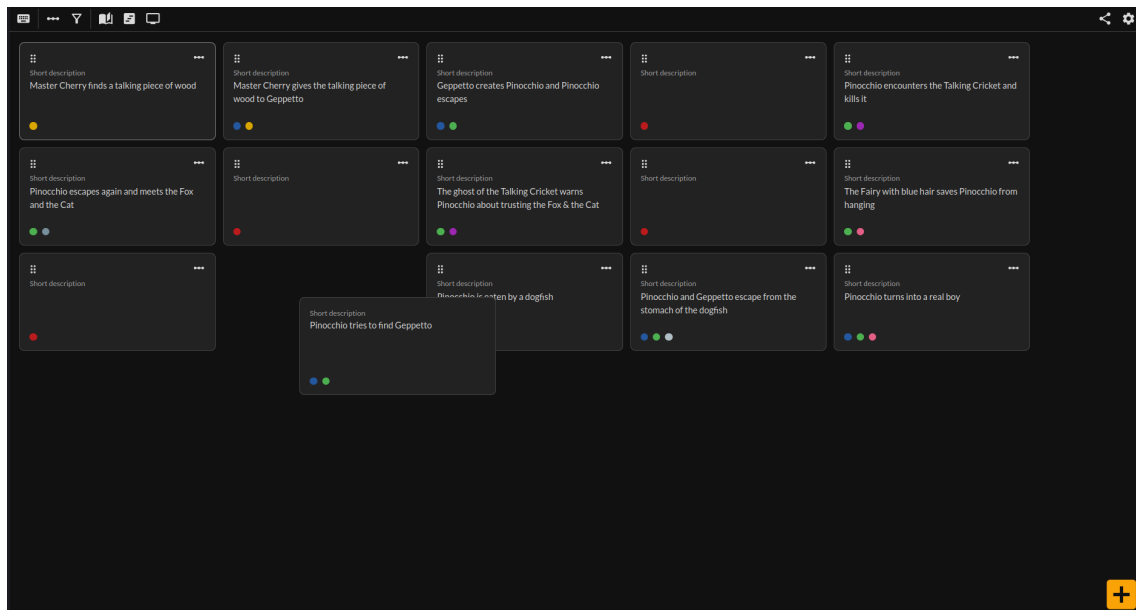


Figure 8. *Uue fragmendi tirimine fragmentide loendisse*

Fragmendi komponendil on kuvatud lühikirjeldus ja siltide värvid, nupp fragmendi aja-joonele lisamiseks või sealt eemaldamiseks ning nupp fragmendi positsiooni muutmiseks hiirega tirimisel. Vajutades fragmendil hiire parema nupuga avaneb menüü fragmendi muutmiseks, kustutamiseks ning piltide lisamiseks.

4.3.2 Fragmentide tirimine

Fragmentide ümberjärjestamiseks kaaluti kahte moodust. Üks neist oli hiirega tirimine, mis puhul saab kasutaja "haarata kinni" fragmendi nurgast ning liigutada see vajalikku kohta. Teise variandina kaaluti fragmendi liigutamist kas nupule vajutades või klahvikombinatsioonide kasutades. Fragmendi liigutamisel klahvikombinatsioonide abil on mitu probleemi. Esiteks on keeruline öelda, milline liigutus oli viimane. Seega tuleks iga liigutuse järel teha eraldi päring tagarakenduse pihta või luua funktsioon, mis loeb kokku liigutuste arvu ning jätab päringud seni ära, kuni tekib paus. Kui fragmendi vana ja uue positsiooni vahele jääb palju teisi fragmente, siis muutub selline samm sammu haaval fragmendi liigutamine kasutaja jaoks tülikaks. Samuti on hiirega tirimine enamike kasutajate jaoks intuitiivsem kui klahvikombinatsioonide kasutamine. Neil põhjustel otsustati esimese variandi kasuks.

Hiirega fragmendi tirimise implemanteerimiseks - seda nii fragmendi loomisel kui fragmen-

tide ümberjärjestamisel - kasutati teeki dnd-kit, mis pakub muu hulgas sensoreid tirimise tuvastamiseks, konteksti tiritavate objektide oleku manageerimiseks ning modifikaatoreid (inglise keeles *modifiers*) tirimispiirkondade piiramiseks.

Objektide tirimise loogika ehk mis juhtub tirimise alguses, ühe objekti tirimisel teise peale, tirimise katkestamisel ning objekti lahtilaskmisel, tuleb arendajal ise implementeerida. Antud rakenduses tuli implementeerida kaks erinevat tirimise viisi: fragmentide ümberjärjestamiseks ning uute fragmentide loomiseks.

Objektide tirimisel jaotatakse objektid kaheks: tiritav objekt (inglise keeles *draggable*) ning objekt mille kohal tiritav objekt lahti lastakse ehk langetatav objekt (inglise keeles *droppable*). Langetatavad objektid võivad olla nii konteinerid, kus objektid asuvad, näiteks fragmentide loend, aga ka objektid mille kohale tiritavat objekti tiritakse, näiteks fragmentide loendis asuvad fragmendid. Seega fragmendid on ühe korraga nii tiritavad kui langetatavad objektid, sest neid ise saab tirida, aga nende kohale saab tirida ka teisi fragmente. Kui fragment tirida teise fragmendi kohale, siis tekitatakse visuaalselt "auk", kuhu tiritav fragment pärast lahtilaskmist positsioneerub.

Fragmentide ümberjärjestamise implementeerimine on lihtsam, sest kõik fragmendid asuvad samas konteineris ning vaba "augu" tekitamine on juba dnd-kiti poolt implemeteeritud. Arendaja peab kirjutama ainult fragmendi lahtilaskmise loogika, mis hõlmab endas fragmentide loendi oleku muutmist.

Fragmendi loomise loogika on keerulisem, sest objekti liigutamine toimub ühest konteinerist teise ning dnd-kiti poolt on implementeeritud vaid objekti asukoha muutus tirimisel. Vaba "augu" tekitamiseks on sel juhul kaks moodust.

Üks neist on uue fragmendi lisamine fragmentide loendisse kohe tirimise alguses. Sel juhul kehtib ümberjärjestamisel kasutatav loogika. Samas tuleb sel juhul muuta tirimise katkestamisel konteiner tagasi esialgseks konteineriks. Sama tuleb teha siis, kui tiritud fragment peaks minema fragmentide loendi piiridest välja ning ka tirimise lõpus, sest fragmendi identifikaatori määrab tagarakendus ning enne identifikaatori määramist ei saa uut fragmenti lõplikult loendisse lisada, sest identifikaatorit on fragmendi muutmise jaoks hiljem vaja. Kuigi selline lähenemine võimaldab automaatset "augu" tekitamist dnd-kiti poolt, muutub programmi kood pideva konteineri muutmise tõttu raskesti loetavaks.

Autor otsustas teise lähenemise kasuks. Fragment lisatakse fragmentide loendisse alles pärast seda, kui kasutaja on fragmendi tirimise lõpetanud ning vaba "augu" tekitamine fragmentide loendis implementeeriti ise. Selleks loodi vastava langetatava objekti kõrvale

kasutajale nähtamatu ajutine objekt, mis tähistab vaba "auku" kuhu tiritav objekt mahuks. Selline lähenemine hoidis koodi puhtamana ja lihtsamini loetavana.

Lisaks objektide tirimise loogikale tuli implementeerida ka üks modifikaator, mis lubaks fragmentide loendis olevate fragmentide liigutamist ainult fragmentide loendi piires, kuid uue fragmendi liigutamist terve akna piires.

4.3.3 Fragmentide muutmine ja kustutamine

Fragmentide muutmiseks tuleb hiire vasaku nupuga vastava fragmendi komponendi peale vajutada. Seejärel avaneb külgpaneel fragmendi informatsiooniga ning pakutakse võimalust välja muuta. Paneelis saab muuta fragmendi lühikirjeldust, detailset kirjeldust, kestust ning lisada või eemaldada silte. Joonisel 9 on kujutatud fragmendi muutmisel avanevat külgpaneeli. Ülejäänud ekraan on jäetud jooniselt välja, sest seda on juba eelnevatel joonistel kujutatud.

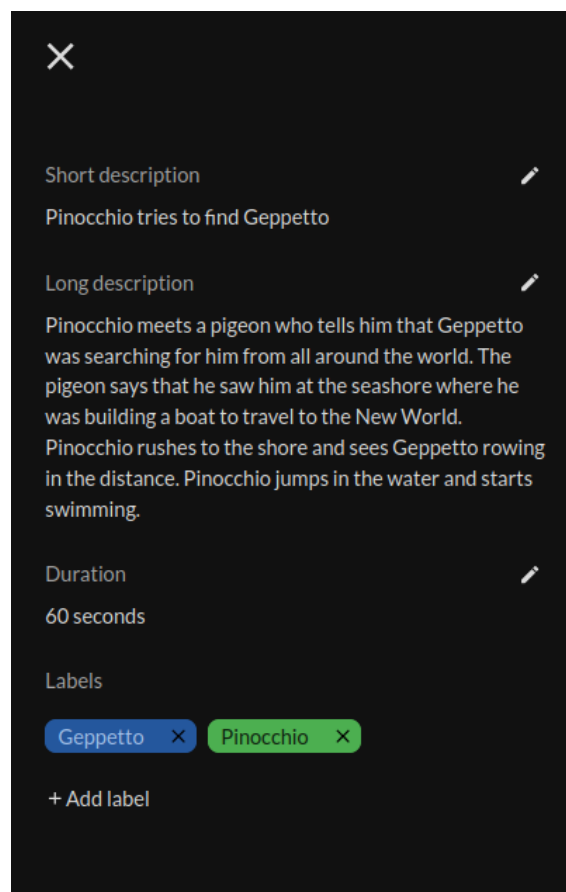


Figure 9. Külgpaneel fragmendi info muutmiseks

Samuti saab fragmenti ajajoonele lisada või sealt eemaldada fragmendi komponendil vastava nuppu peale vajutades. Fragmendi tirimiseks on komponendi vasakul üleval servas vastav ikoon. Fragmenti saab kustutada vajutades komponendil hiire paremat klahvi ning

valides "Delete fragment".

4.3.4 Fragmentide detailne vaade

Lisaks fragmentide loendile pakub rakendus ka fragmentide detailset vaadet. Selles vaates kuvatakse kogu fragmendiga seotud informatsioon, sealhulgas ka pildid ja kommentaarid. Piltidest ja kommentaaridest räägitakse hilisemas alapeatükis. Fragmentide vahel saab liikuda nooleklahvidega. Joonisel 10 on kujutatud fragmendi detailset vaadet.

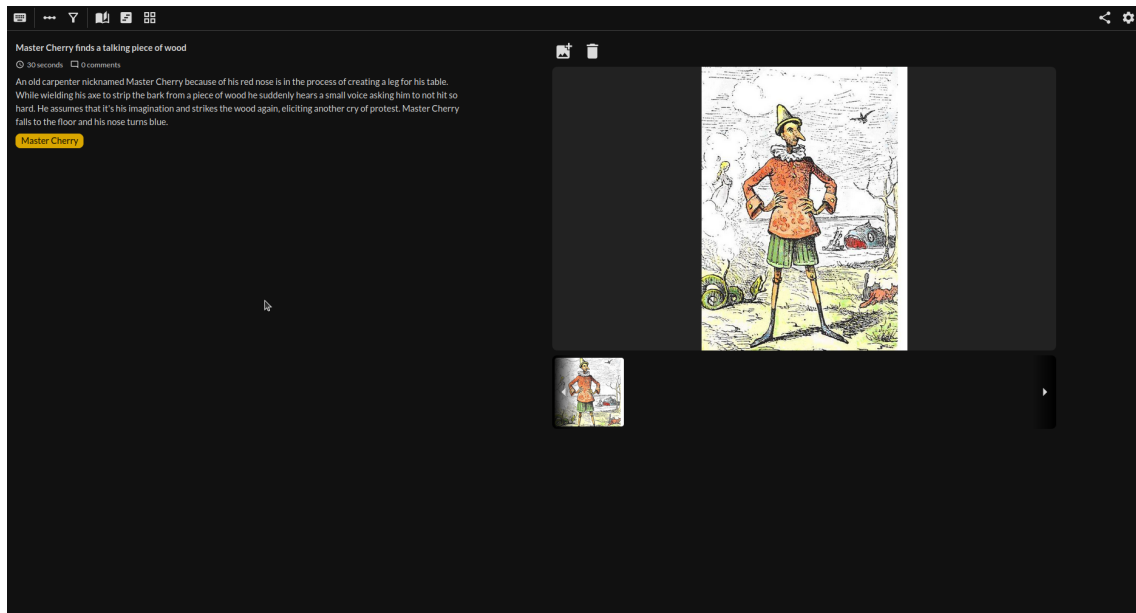


Figure 10. *Fragmentide detailne vaade*

4.3.5 Fragmentide agregatsioon

Kuigi süžee loomine ja muutmine toimub üldjuhul väiksemate tükkidena, on lõppkokkuvõttes tegemist siiski ühe tervikuga ning seda peab olema võimalik ka ühe tervikuna lugeda. Selle jaoks on rakenduses olemas paneel, mis agregeerib kokku kõik ajajoonel olevate fragmentide detailsed kirjeldused ning võimaldab süžeed lugeda ühtse tervikuna. Joonisel 11 on näidatud vasakul ekraani servas avanev terviksüžee paneel.

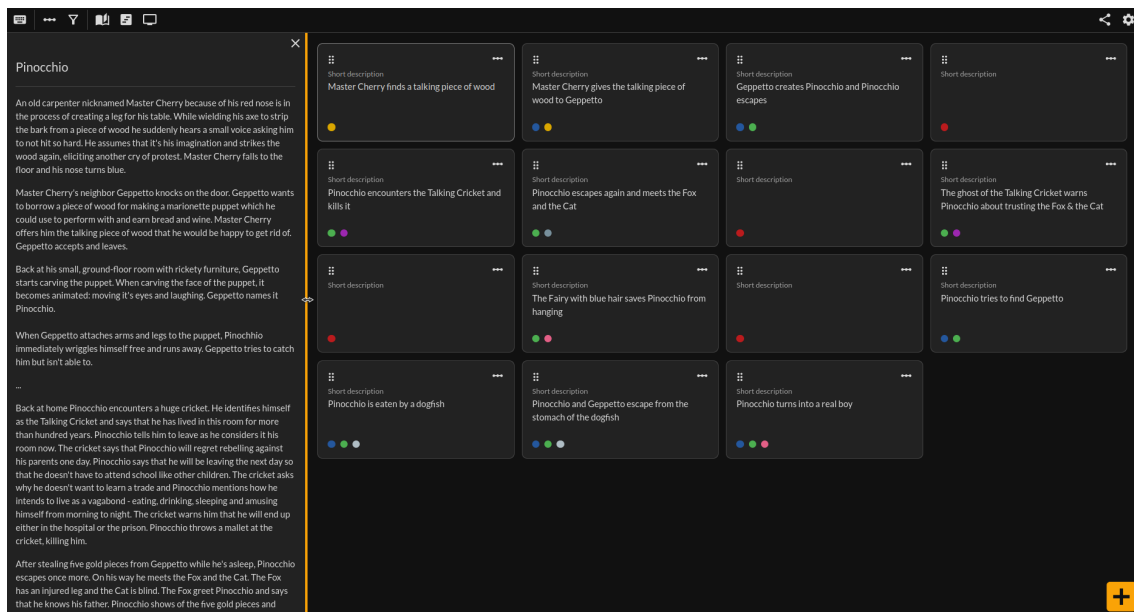


Figure 11. *Agregeeritud fragmendid (vasakul paneelis)*

Paneeli saab vastavalt vajadusele avada ja sulgeda ning selle laiust on võimalik muuta. Arvesse võetakse ka fragmentidele rakendatud filtreid ning kuvatakse vaid nende fragmentide kirjeldusi, mis filtrid läbivad. Filtritest räägitakse siltide alapealtükis.

4.4 Sildid ja fragmentide filtreerimine

Üsna tavapärane võte filmisüžee arendamisel on keskenduda ainult teatud osale süžest. Näiteks võib keskenduda ainult ühe tegelase loole ning muu hetkeks ära unustada, seejärel keskenduda järgmisele tegelasele jne. Igale tegelasele eraldi keskendumine aitab paremini paika seada tegelaste motivatsioonid ja eesmärgid. Samamoodi võib käituda ka näiteks paralleelsete süžee liinide puhul.

Et seda võimalust filmitegijatele pakkuda, peaks süžee fragmente millegi alusel grupeerima ning filtreerima. Ühe võimalusena kaaluti defineerida rakenduses teatud grupid nagu "tegelane", "asukoht" jms. Kasutaja saaks seejärel nendesse gruppidesse lisada silt - näiteks "Peeter" gruppi "tegelased" - ning seejärel siduda need sildid teatud fragmentidega. Nende siltide alusel saaks fragmente seejärel filtreerida. Selline lähenemine lubaks implementeerida lisafunktsionaalsusi nagu näiteks tegelaste või asukohtade soovitamise stsenaariumi kirjutamise faasis. Samas seab see lähenemine rakenduse poolt kasutajale piirangud gruppide osas. Loomulikult oleks võimalik lasta kasutajal ka gruppe luua ning eristada gruppe selle alusel kas rakendus oskab antud gruppi kuuluvate siltidega midagi peale hakata või mitte. Näiteks tegelaste gruppi kuuluvate siltide puhul on rakendatavad teatud lisafunktsionaalsused, kuid kasutaja defineeritud gruppidesse kuuluvate siltide puhul

mitte. See aga muudaks rakenduse kasutamise segasemaks ning prototüübi arendamisel otsustati lihtsama lähenemise kasuks: kasutaja saab silte vabalt defineerida ning neid fragmentidele lisada ning neid ei grupeerita. Siltide alusel toimub fragmentide filtreerimine.

4.4.1 Siltide lisamine

Sildid koosnevad kolmest osast: kirjeldus, värv ja identifikaator. Silte saab luua fragmentide loomise ajal või fragmenti muutmise paneelis. Kuigi neid luuakse fragmentide vaates, on sildid tegelikult projektipõhised ning ühte silti on võimalik lisada mitmele fragmentile. Samuti on võimalik ühele fragmentile lisada mitu silti.

Sildi loomisel tuleb kirjutada sildile kirjeldus ning valida värv. Värvide valimiseks on kaks võimalust: valida värv pakutud värvide seast või lasta rakendusel genereerida suvaline värv. Pärast kirjelduse ja värvi valimist saab sildi projekti salvestada ning fragmentile lisada. Fragmentile saab silte lisada ka juba olemasolevate siltide seast. Joonisel 12 on kujutatud vaadet, mis avaneb kasutajale uue sildi loomisel.

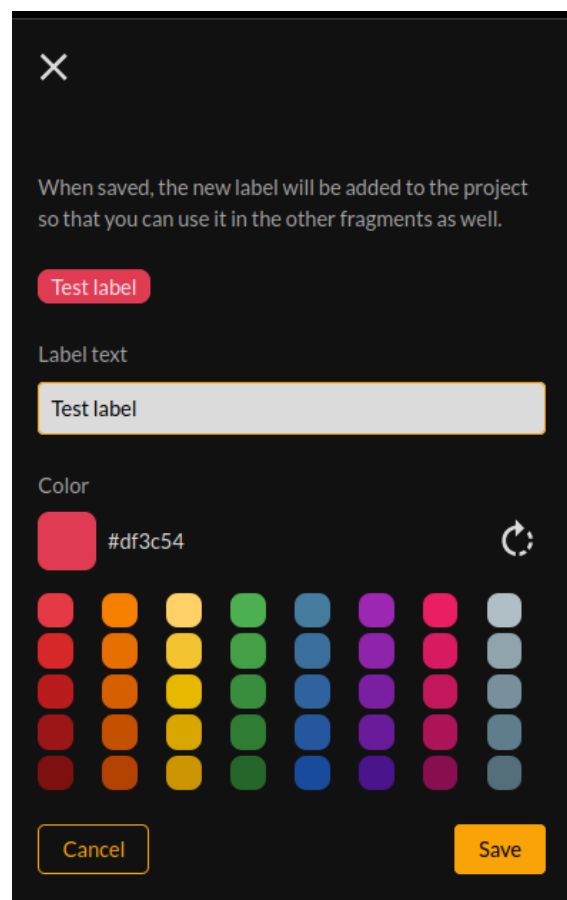


Figure 12. Sildi loomise vaade

Siltide alusel on võimalik fragmente filtreerida. Filtreerimisel kuvatakse kasutajale ainult need fragmendid, millel on olemas vähemalt üks valitud siltidest. Joonisel 13 on näidatud menüüd filtrite lisamiseks ja eemaldamiseks.

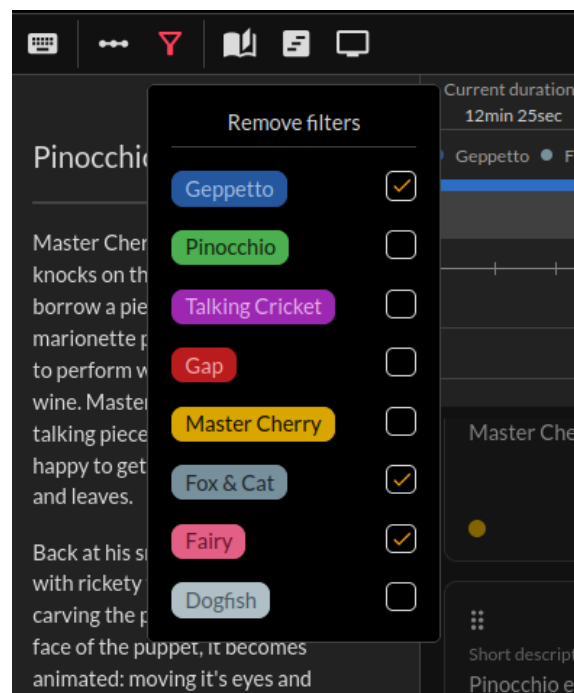


Figure 13. *Fragmentide filtreerimine*

4.4.2 Ajajoon

Siltide alusel luuakse kasutajale ka ajajoon. Ajajoon on mõeldud süžee visualiseerimiseks ning kujutab millised sildid milliste fragmentidega seotud on. Olenevalt sellest, kuidas filmitegija otsustab silte kasutada, võiks ajajoon anda ülevaate loo rütmist, tegelaste esinemissagedusest, lünkadest süžees, loo haripunktidest vms. Joonisel 14 on kujutatud projekti siltide põhjal loodud ajajoon.

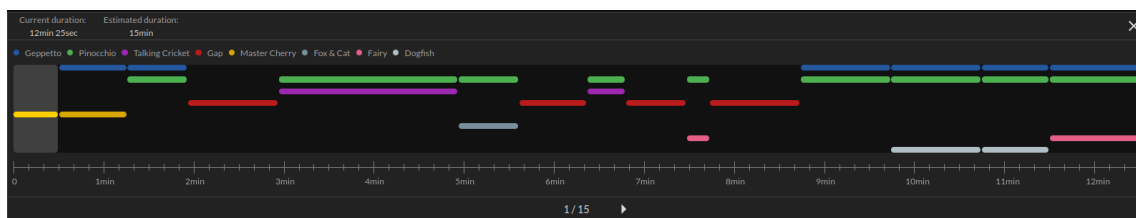


Figure 14. *Siltide põhjal loodud ajajoon*

Ajajoonel on ära märgitud ka filmi oodatav pikkus ning hetkepikkus. Hetkepikkus arvutatakse kokku ajajoonel olevate fragmentide pikkuste põhjal.

Fragmente on võimalik ajajoonelt eemaldada. See lubab fragmentide alleshoidmist ilma, et need süžees kajastuks ning on kasulik fragmentide puhul, mis hetkel süžeesse ei sobi või milles filmitegija pole täiesti kindel, kuid mida ta ei taha ka ära kustutada.

4.5 Pildid

Mõnda ideed või idee osa on vahepeal lihtsam kirjeldada läbi visuaalse materjali. Näiteks stseenis kasutatava tausta detailse kirjeldamise asemel oleks lihtsam lisada foto või joonistada pilt.

Piltide lisamiseks tuleks vajutada fragmentide detailses vaates vastavale nupule või fragmentide loendis parema hiireklahviga vastaval fragmendil ning valida avanenud menüüst piltide lisamise võimalus. Rakendus pakub fragmendile pildi lisamiseks kahte moodust: seadmest üles laadides või ise joonistades. Mõlemal puhul saab soovi korral lisada pildile ka kommentaari. Joonisel 15 on näide sketši joonistamisest fragmendile. Sketš on joonistatud rakendust testinud filmitegija poolt.

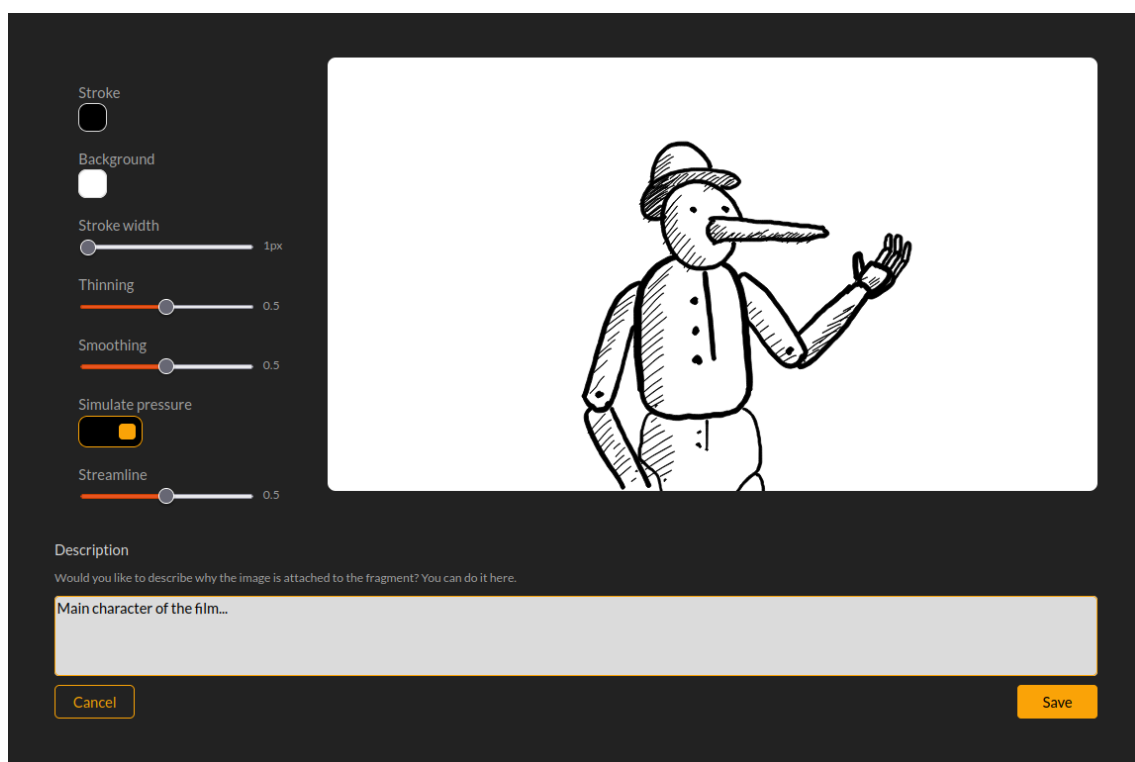


Figure 15. *Fragmendile pildi joonistamine*

Piltide joonistamisel kasutatakse teegi Perfect Freehand abi. Teek pakub ühtainsat meetodit, mis saab sisendiks järjendi joont kujutavatest koordinaatidest ning väljastab polügoni. Rakendus kogub kokku lõuendile joonistamisel tekkivad koordinaadid ning saadab need

Perfect Freehand-i pakutavale meetodile. Sealt tagastatud polügon teisendatakse ümber SVG jooneks (inglise keeles *path*), mis seejärel lõuendile kantakse. Rakendus hoiab infot joonistatud joonte kohta pinus (inglise keeles *stack*) ning võimaldab eelnevaid tegevusi tagasi pöörata (inglise keeles *undo*) või tagasi pööratud tegevusi uuesti teha (inglise keeles *redo*).

Salvestamisel saadetakse pilt tagarakendusele. Seadmest üles laetud pildi puhul vähendatakse vajadusel juba eesrakenduses pildi mõõtmeid, kuid mõõtmete kontroll ja vajadusel nende vähendamine toimub ka tagarakenduses. Mõõtmete muutmiseks kasutatakse tagarakenduses teeki *ImgScalr*. Tagarakendus kontrollib ka saadetud faili MIME tüüpi kasutades selleks *Apache Tika* teeki. Pildile antakse unikaalne identifikaator ning pilt salvestatakse tagarakendust jooksvatava masina failisüsteemi. Viide pildile salvestatakse andmebaasi. Viite kustutamisel andmebaasist kustutatakse ka pilt automaatselt failisüsteemist.

4.6 Projektide jagamine teiste kasutajatega

Teistelt kasutajatelt tagasiside saamiseks võimaldab rakendus projekti jagamist teiste kasutajatega. Jagatud projektis muudatusi teha ei saa, kuid on võimalik fragmente kommenteerida. Kui keegi kasutajaga projekti jagab või keegi kasutaja jagatud projektis kommnetare lisab, saab kasutaja selle kohta rakendusesisese teavituse reaalajas. Teavitused, mis saadeti ajal, mil kasutaja rakendust ei kasutanud, kuvatakse järgmisel rakenduse kasutamisel.

Kasutaja näeb jagatud projektis täispikka süžeed, mille moodustavad fragmentide detailsed kirjeldused. Süžee kõrval kuvatakse fragmentide pealkirjad, millele vajutades keritakse süžee autmaatselt vastava fragmendi juurde. Samuti on näha hetkel aktiivse fragmendiga seotud pilte ja kommentaare ning kommentaarilahtrit, kuhu kasutaja võib kirjutada oma tagasiside. Joonisel 16 on kujutatud jagatud projekti vaadet pealkirjade loendiga vasakul servas, süžeeaga vasakul keskel, piltidega paremal keskel ning kommentaarilahtriga paremal servas.

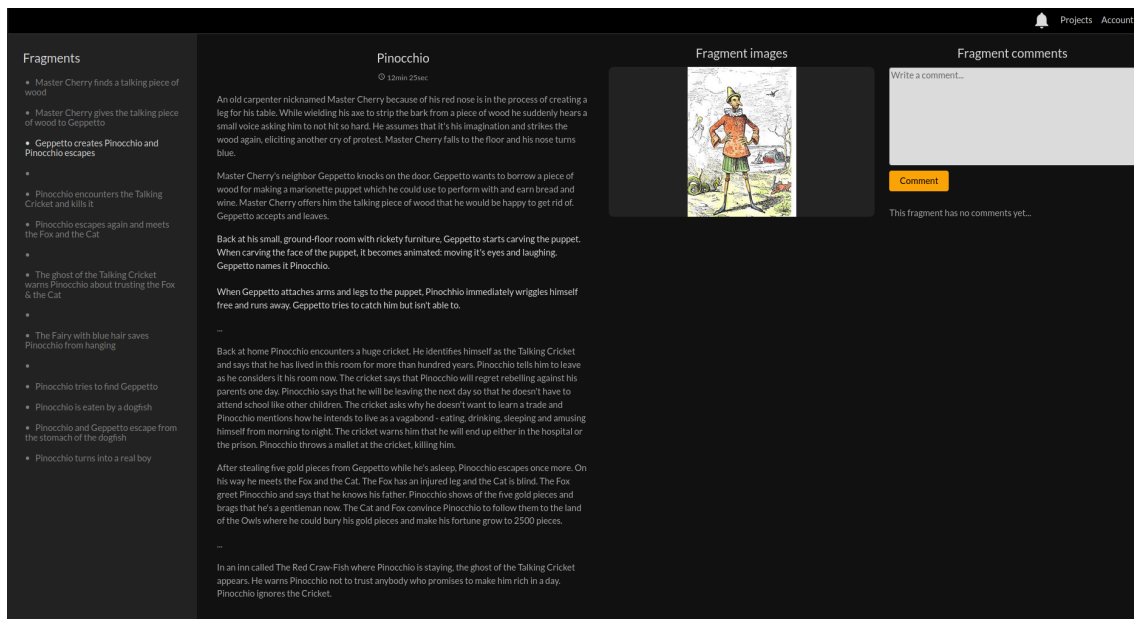


Figure 16. *Kasutajaga jagatud projekti vaade*

4.6.1 Kasutajate otsing

Kasutajate otsimiseks kaaluti kahte erinevat lähenemist: inkrementaalne otsing ja mitte-inkrementaalne otsing. Inkrementaalne otsing tähendab seda, et otsing sooritatakse uuesti iga otsitud fraasi muudatuse korral ning kasutajale kuvatakse hetke tulemused. Lihtsam neist lähenemistest on mitte-inkrementaalne otsing, sest sel juhul sisestab kasutaja otsitava fraasi - antud rakenduse puhul otsitud kasutaja ees- ja perekonnanime - ning seejärel tehakse päring tagarakendusele, mis tagastab tulemused. Selle lähenemise eeliseks on see, et tagarakendusele tehakse ainult üks päring. Inkrementaalse otsingu korral tehakse päring iga lisatud või eemaldatud tähe korral. Et vältida mittevajalike päringute tegemist kasutaja kiire trükkimise korral, võib päringut seni edasi lükata, kuni trükkimisse tekib paus ning alles seejärel teha tagarakendusele päring. Kuid isegi sellisel korral tehakse üldiselt tagarakendusele päringuid rohkem kui mitte-inkrementaalse otsingu korral. Inkrementaalse otsingu eeliseks on meeldivam kasutajakogemus. Selle tõttu otsustatigi antud rakenduses kasutada inkrementaalset otsingut kasutades teegi Lodash pakutavat *debouncer*-it otsingu edasilükkamiseks.

Et otsing oleks tõhusam, kasutatakse andmebaasis GIN indekseid (*Generalized Inverted Index*) ja trigramme läbi pg_trgm laienduse. Nii ees- kui perenimest tekitatakse kolmetähelised tükid (trigrammid), mis vastavad antud nimele. Nende tükkide kaudu saab kasutajat otsida ning kõige täpsemad vastused saadetakse päringule vastuseks. Näiteks nimele Peeter loodavad trigrammid on "Pee", "eet", "ete" ning "ter". Kui otsing sisaldab mõnda neist trigrammidest, siis on antud rida tabelis päringule vasteks. Vastete hulgast valitakse välja

kõige suurema trigrammide kattuvusega vasted ning saadetakse need päringu vastusena tagasi eesrakendusele.

Otsingu tulemuste seast on kasutajal võimalik välja valida otsitud kasutaja ning temaga projekti jagada.

4.6.2 Teavitused

Kui keegi kasutajaga projekti jagab või kasutaja projekti kommenteerib, siis saadetakse selle kohta kasutajale teavitus. Kui kasutaja parasjagu rakendust kasutab, saadetakse teavitus reaalajas, vastasel korral kuvatakse kõik lugemata teavitused järgmisel rakenduse kasutamisel vastavas menüüs. Kasutajal on võimalik teavitusi nii ühe kaupa kui korraga loetuks märkida.

Teavituste reaalajas saatmiseks kaaluti kolme võimalust: *Long polling*[33], *WebSockets*[34] ning *Server-Sent Events*[35] (SSE). *Long Polling*-i korral teeb eesrakendus tagarakendusele HTTP päringu ning olenevalt sellest, kas tagarakendusel on sellele vastus või mitte, vastab või hoiab päringut lahti seni kuni see aegub. Nii vastuse kui päringu aegumise korral teeb eesrakendus uue päringu ning seda protsessi korratakse rakenduse kasutamise lõpuni. *Long Polling*-i eeliseks on see, et tegu on HTTP päringutega ja on seega toetatud kõigi brauserite poolt. Antud rakenduses otsustati aga seda meetodit mitte kasutada, sest pidevate uute ühenduste loomise tõttu ei ole *Long Polling* väga ressursisõbralik ning kuna teavituste saatmine ei ole väga tihe, siis ei ole see lähenemine õigustatud.

WebSocket API lubab kahepoolset reaalajas suhtlust kliendi ja serveri vahel kasutades spetsiaalset protokollit. HTTP-d kasutatakse *WebSocket*-i protokollile üleminekuks. *WebSocket*-id nõuavad vähem ressursi kui *Long Polling*[36], kuid neid on keerulisem implementeerida. Kuna antud rakenduses pole vaja kahepoolset suhtlust, vaid ainult teavituste saatmist serverilt kliendile, siis parimaks lahenduseks osutus hoopis kolmas variant.

SSE võimaldab ühepoolset suhtlust serverilt kliendile. Suhtlemiseks luuakse HTTP ühendus, mida kasutatakse sündmuste (inglise keeles *event*) edastamiseks. SSE on nõuab vähe ressursi, sest kasutatakse ainult ühte ühendust. Samuti on brauserites implementeeritud automaatne ühenduse taasloomine ühenduse katkemise korral. SSE nõrkuseks on see, et suhtlus on ühepoolne ning brauserid toetavad ainult teatud arvu korraga lahtiolevaid aknaid SSE ühenduse loomiseks, kuid antud rakenduses pole kumbki neist probleemiks.

Tagarakenduselt teavituste saamiseks luuakse eesrakenduses *EventSource* objekt, mis saadab päringu vastavale API lõpp-punktile (inglise keeles *endpoint*). Õnnestunud päringu

järel on tagarakendus võimeline läbi loodud objekti eesrakendusega suhtlema. Eesrakenduses on defineeritud, kuidas tagarakendusest tulnud sõnumitega ümber käia.

Tagarakenduses luuakse eesrakenduselt saadud päringu järel *emitter* objekt, mis hakkab edaspidi eesrakendusega suhtlema. Kui kasutaja saab teavituse, salvestatakse see andmebaasi ning kui tagarakendusel on antud kasutajale vastav *emitter*, siis saadetakse teavitus lisaks andmebaasi salvestamisele ka eesrakendusele SSE-na. Kui kasutajaga seotud *emitter*-it ei eksisteeri tähendab see seda, et kasutaja pole hetkel rakendust kasutamas ning reaajas teavituse saatmist pole vaja (ega võimalik) teha.

4.7 Stsenaariumi kirjutamine

Loo arendamise lõpus on see võimalik kirjutada ümber standarditele vastavaks stsenaariumiks[37]. Stsenaarium kirjeldab olevikuvormis, mida filmi vaataja ekraanil nägema hakkab, kui film valmis saab. Stsenaariumis välditakse tegelaste mõtete ja tunnete kirjeldamist, selgitusi, miks miski juhtub ja kõike muud, mida filmi vaataja tegelikult ekraanil ei näe.

Korralik stsenaarium on filmile rahalise toetuse taotlemise aluseks. Kui täpset stsenaariumi formaadi jälgimist stsenaaristilt nõutakse sõltub rahastusega tegelevast organisatsioonist, kuid üldiselt püüavad stsenaaristid järgida kokkulepitud standardeid ning stsenaariumi kirjutamiseks kasutatakse rakendusi, mis neist kinni peavad.

Stsenaariumi kirjutamiseks kaaluti kahte varianti. Esimese puhul näidatakse kasutajale kahte eri vaadet, millest ühes kirjutaks ta Fountain[38] formaadis stsenaariumi ning teises kuvatakse see vormistatuna õigesse formaati. Sellise lähenemise tugevuseks oleks see, et Fountain formaat on kompaktne ning aktsepteeritud paljude eri stsenaariumikirjutamise rakenduste poolt. Samuti poleks sel juhul vaja rakendusse lisada rikasteksti (inglise keeles *rich text*) redaktorit. Puuduseks on kehv kasutajakogemus.

Teise variandi puhul implementeeritaks rikasteksti redaktor, mis kuvaks kirjutatud stsenaariumit peaaegu WYSIWYG (*What You See Is What You Get*) kujul, ainult ilma lehekülgedele jagamata. Sellise lähenemise positiivne külg on parem kasutajakogemus, sest kasutaja töötab samas redaktoris, kus talle teksti kuvatakse. Samas nõuab see rikasteksti redaktori kasutuselevõtmist. See aga tähendab, et info tuleb struktureerida sarnaselt sellele, kuidas redaktor infot kuvab või implementeerida funktsioon, mis infot pidevalt ühest struktuurist teise konverteeriks. Kuna filmitegijad on harjunud stsenaariumit kirjutama redaktoris, mis kuvab stsenaariumit õiges formaadis, siis otsustati viimase lähenemise kasuks.

4.7.1 Stsenaariumi formaat

Stsenaarium algab tiitellehega, kus on kirjas filmi pealkiri ning autor. Samuti võib sinna lisada kuupäeva ja autori kontaktandmed.

Tavaliselt defineeritakse lehekülgedele jäetavad veerised tollides. Üks toll on võrdne 25.4 millimeetriga. Stsenaariumi igale leheküljele jäetakse vasakuks veeriseks 1.5 tolli ning paremale, üles ja alla 1 toll.

Igale leheküljele peale tiitellehe ja esimese lehekülje lisatakse leheküljenumber. Tiitellehte nummerdamisel ei arvestata. Esimest lehekülge arvestatakse, kuid sellele numbrit ei lisata. Lehekülje number lisatakse lehekülje paremale üles, pidades silmas paremat veerist. Ülevalt jäetakse veeriseks pool tolli. Leheküljenumbri lõpus on punkt.

Stsenaariumis kasutatakse šriftiks on Courier suurusega 12pt.

Stsenaarium koosneb erinevatest plokkidest, millest igaüks kirjeldab teatud osa stsenaariumist ning millel on oma kindlad vormistuse nõuded. Stsenaarium sisaldab järgnevaid plokke:

- Pealdis - kirjeldab aega ja kohta, kus stseen aset leiab (inglise keeles *heading*)
- Tegevus - kirjeldab, mis stseenis juhtub (inglise keeles *action*)
- Karakter - märgistab, milline tegelane räägib (inglise keeles *character*)
- Sulgplokk - kirjeldab, kuidas tegelane räägib (inglise keeles *parenthetical*)
- Dialoog - kirjeldab, mida tegelane räägib (inglise keeles *dialogue*)
- Üleminek - märgistab, kuidas toimub üleminek ühest stseenist teise (inglise keeles *transition*)

Pealdis on alati rasvases kirjas ning kirjutatud suurtähtedega. Pealdis algab sõnaga "INT." või "EXT.", mis kirjeldab, kas stseen toimub siseruumides või väljas. Pärast seda defineeritakse koht ning aeg, näiteks "ÜLIKOOL - PÄEV".

Tegevuse kirjeldamiseks formaadile erinõudeid ei ole, kuid tihti kirjutatakse tegelase nimi esmakordsel mainimisel suurtähtedega.

Karakteri defineerimiseks kasutatakse suurtähti ning lehekülje vasakule veerisele lisatakse omakorda 2.2 tolline veeris, kokku 3.7 tolli.

Sulgplokk on alati sulgudes ning vasakule veerisele lisatakse 1.6 tolline veeris, kokku 3.1

tolli.

Dialoogi puhul lisatakse vasakule veerisele veel 1 toll, kokku 2.5 tolli.

Üleminek märgitakse lehekülje paremasse serva, see kirjutatakse suurte tähtedega ning lõppu lisatakse koolon.

4.7.2 Stsenaariumi kirjutamise vaade

Eesrakenduses on stsenaariumi kirjutamiseks loodud spetsiaalne vaade, mis sisaldab fragmentide pealkirju vaate vasakul ääres, fragmentide detailseid kirjeldusi keskel ning stsenaariumi kirjutamiseks vajalikku redaktorit paremal.

Vajutades hiirega vasakul asuvate pealkirjade peale, keritakse vaate keskmine osa vastava fragmendi detailse kirjelduseni. Nii on võimalik navigeerida eri fragmentide vahel. Stsenaariumit saab kirjutada toetudes varemalt lisatud fragmentidele ning selle plokid kuvatakse kasutajale õigesti vormistatuna.

Stsenaarium ise on fragmentidest sõltumatu. Otsustati sellise lähenemise kasuks, et võimaldada kirjutada stsenaariumeid ka juhul kui kasutaja ei soovi esmalt fragmentide kaudu süžeed kirja panna. Joonisel 17 on kujutatud stsenaariumi kirjutamise vaadet fragmentide loeteluga vasakul servas, süžeeaga vasakul keskel ning stsenaariumi redaktoriga paremal.

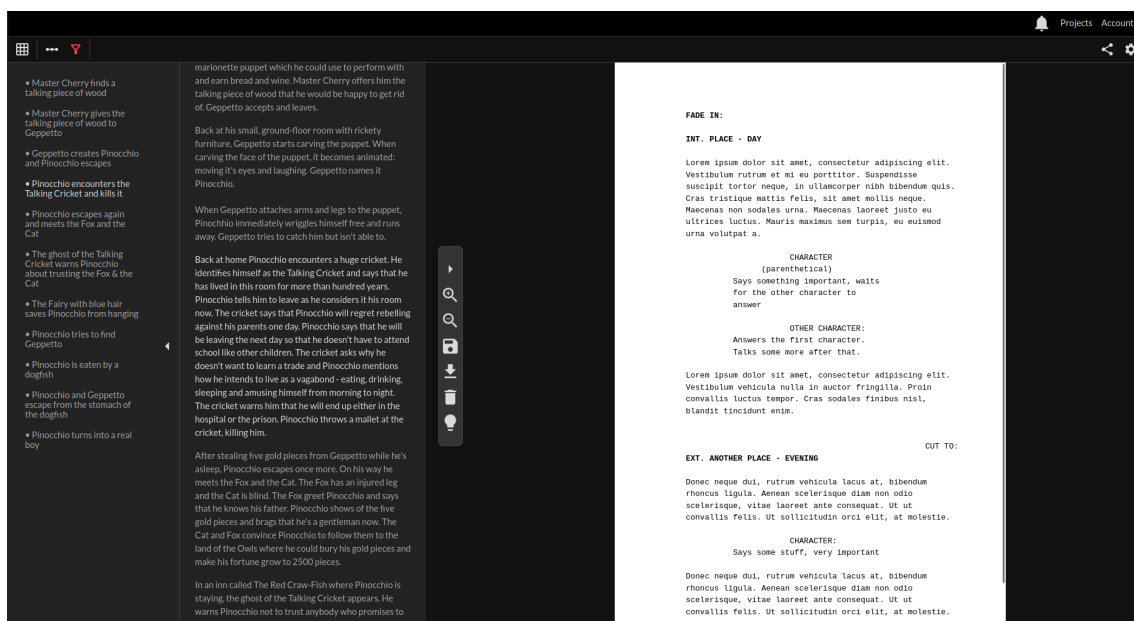


Figure 17. Stsenaariumi kirjutamise vaade

Stsenaariumi kirjutamiseks kasutatakse teeki Slate-React (edaspidi Slate). Slate võimaldab arendajal luua spetsiaalsete funktsionaalsuste ja andmestruktuuridega töötavaid tekstiredaktoreid. Dokumenti käsitletakse kui puud ehk tsükliteta graafi, mille sõlmi on võimalik vastavalt vajadusele modifitseerida. Slate pakub API-t, mille kaudu on võimalik sõlmi lisada, eemaldada, ümber järjestada, muuta jne.

Rakenduses on defineeritud igale eelnevalt kirjeldatud stsenaariumi plokile vastavad komponendid, millele on CSS-i kaudu sätestatud vajalik kujundus. Igal sõlmel dokumendis on oma tüüp ning tüübi põhjal otsustatakse, milline komponent parasjagu kasutajale kuvada. Teksti lisamise, kustutamise ja muutmise operatsioonid on implementeeritud Slate'is. Samuti tegeleb Slate automaatselt tekstioperatsioonide ajaloo meespidamisega. Ülejäänud operatsioonid, näiteks sõlme tüüpi muutmine või klahvikombinatsioonide kasutamine, tuleb arendajal implementeerida ise.

4.7.3 Klahvikombinatsioonid

Stsenaariumi kirjutamise mugavamaks muutmiseks on implementeeritud klahvikombinatsioonid (inglise keeles *shortcuts*), mida kasutatakse ka teistes stsenaariumi kirjutamise programmides:

- Plokk muudetakse automaatselt pealdiseks kui plokki alustatakse fraasiga "int." või "ext." (tõstutundetu)
- Plokk muudetakse automaatselt karakterplokiks kui plokki alustatakse tabeldusklahvi vajutamisega (inglise keeles *tab*)
- Plokk muudetakse automaatselt sulgplokiks, kui plokki alustatakse suluga
- Plokk muudetakse automaatselt üleminekuplokiks, kui plokki alustatakse suurtest tähtedest koosneva fraasiga, mille lõppu lisatakse koolon

Lisaks sellele on implementeeritud automaatne loogiline plokide üleminek ühest plokist teise:

- Pealdisploki järel tekitatav plokk muudetakse automaatselt tegevusplokiks
- Karakterploki järel tekitatav plokk muudetakse automaatselt dialoogiplokiks
- Üleminekuploki järel tekitatav plokk muudetakse automaatselt pealdisplokiks

Uus plokk luuakse *Enter* klahvi vajutades.

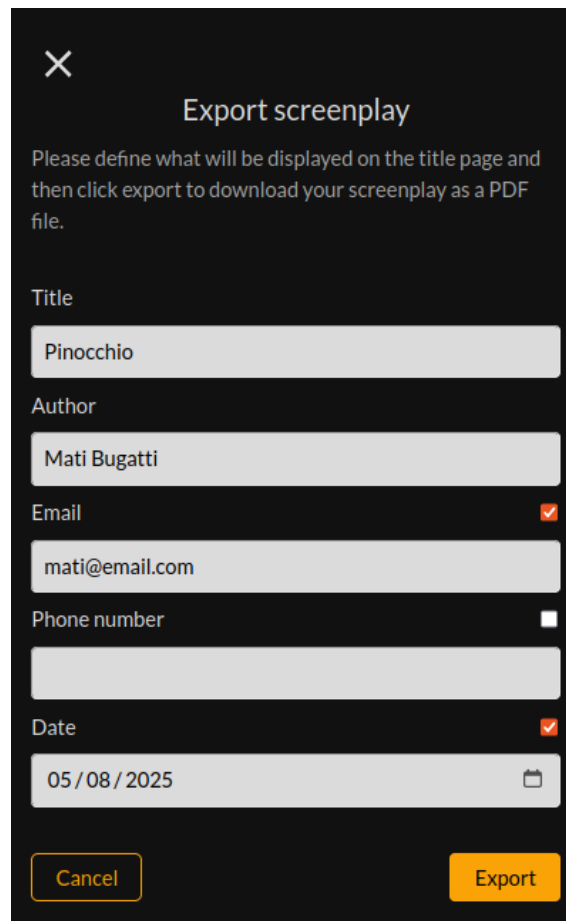
Kasutajal on võimalik juba eksisteeriva plokki tüüpi käsitsi muuta valides stsenaariumivaate menüüst vastava plokkitüübi või kasutades klahvikombinatsioone:

- Alt + H - pealdisplokk
- Alt + A - tegevusplokk
- Alt + C - karakterplokk
- Alt + P - sulgplokk
- Alt + D - dialoogplokk
- Alt + T - üleminekuplokk

Stsenaariumi salvestamiseks võib vajutada vastaval nupul stsenaariumivaate menüüs või kasutada klahvikombinatsiooni Ctrl + S. Stsenaariumi salvestamisel saadetakse stsenaariumi sisu tagarakendusele, kus see JSON formaadis andmebaasi salvestatakse.

4.7.4 Stsenaariumi eksportimine

Kirjutatud stsenaariumit on võimalik PDF failina alla laadida. Stsenaariumi eksportimiseks tuleb kasutajal vajutada vastavat nuppu stsenaariumivaate menüüs. Seejärel avaneb kasutajale külgpaneel, kus ta võib tahtmise korral muuta filmi pealkirja ja autori nime ning lisada kontaktandmed ja kuupäeva. Vaikesätetena pakutakse filmi pealkirjaks projekti pealkirja, autoriks kasutaja ees- ja perekonnanime, e-mailiks kontoga seotud e-maili ning kuupäevaks hetke kuupäeva. Joonisel 18 on kujutatud tiitllehe loomiseks avanevat paneeli, mis kuvatakse kasutajale stsenaariumi eksportimisel.



✕

Export screenplay

Please define what will be displayed on the title page and then click export to download your screenplay as a PDF file.

Title

Pinocchio

Author

Mati Bugatti

Email ☒

mati@email.com

Phone number ☐

Date ☒

05/08/2025

Cancel Export

Figure 18. *Stsenaariumi eksportimisel avanev paneel*

PDF faili loomiseks kasutatakse tagarakenduses teeki Apache PDFBox. PDFBox võimaldab PDF failide lugemist, muutmist ja loomist, millest antud rakenduses kasutatakse viimast. PDFBox võimaldab teksti lisamist ning lihtsamat kujundamist, näiteks peamisi šrifte ja rasvast kirja. Muu kujundamine tuleb arendajal implementeerida.

Stsenaariumi eksportimisel luuakse alguses objektid dokumendile ja esimesele leheküljele. Lehekülje objekte luuakse hiljem vastavalt vajadusele jooksvalt juurde ning lisatakse need dokumenti. Esiteks luuakse ja lisatakse dokumenti tiitelleht.

PDFBox ei jälgi leheküljele lisatava teksti pikkust ning ei tegele automaatsete reavahetustega. Seega stsenaariumi plokkide teksti tervikuna leheküljele lisamisel jookseks tekst leheküljelt välja. Selle vältimiseks jagab tagarakendus kõikide stsenaariumi plokkide tekstid ridadeks võttes arvesse plokkidele seatud veeriseid ning arvutades kui palju teksti ühele reale antud plokis võib lisada. Read lisatakse ridade järjendisse. PDFBox-is toimuvad arvutused *point* ühikutes ning tollid tuleb nendesse ümber arvutada.

Ridade järjendis olevad read lisatakse ühekaupa leheküljele võttes jällegi arvesse veeriseid ning nõudeid kujundusele olenevalt sellest, millisesse plokki antud rida kuulus. Kui rida leheküljele ei mahu, luuakse uus lehekülg ning lisatakse rida hoopis sinna. Igale leheküljele peale esimese lisatakse leheküljenumber.

Kui kõik read on dokumenti lisatud, saadetakse dokument tagasi eesrakendusele, kust see kasutaja masinasse salvestatakse.

5. Tulemused

Antud peatükis räägitakse bakalaureusetöö tulemustest. Esmalt kirjeldatakse lõpptulemust ning kasutajate tagasisidet sellele ning seejärel rakenduse edasiarenduse võimalusi.

5.1 Loodud rakendus

Käesoleva lõputöö tulemusena arendati välja lühianimafilmidele süžee ja stsenaariumi kirjutamiseks mõeldud veebirakenduse prototüüp. Rakendus võimaldab registreeritud kasutajatel kirja panna mõtteid või stseene, mida hoitakse rakenduses järjestatuna ning mis moodustavad kokku filmi süžee. Rakenduses nimetatakse neid süžee tükke fragmentideks.

Kasutajad saavad mugavalt fragmente luua, muuta, ümber järjestada ning kustutada. Sisestatud info põhjal kuvatakse kasutajale lühikestest kirjeldustest koosnev ülevaade, detailsetest kirjeldustest agregeeritud terviklik süžee ning ajajoon, mis annab ülevaate loo rütmist. Samuti arvutab rakendus kokku filmi kogukestuse vastavalt fragmentidele määratud pikkustele.

Fragmentidele on võimalik lisada silte, mille alusel neid saab hiljem filtreerida. Filtreerimine rakendub nii lühiülevaatele, terviklikule süžeele kui ka ajajoonele. Samuti saab fragmentidele lisada pilte, mis on eriti kasulik just animafilme luues. Kasutaja saab pilte üles laadida oma arvutist või joonistada ise lihtsamaid sketše.

Enne süžee stsenaariumiks vormistamist võib kasutaja oma projekte teiste kasutajatega jagada ning saada fragmentidele lisatud kommentaaride näol neilt tagasisidet. Nii projekti jagamisel kui kommentaaride lisamisel saabub kasutajale rakendusesisene teavitus.

Lõpuks pakub rakendus ka süžee stsenaariumiks kirjutamise võimalust. Stsenaariumit vormindatakse vastavalt kokkulepitud standarditele juba kirjutamise käigus ning kirjutamise protsess tehakse võimalikult mugavaks kasutades üldtuntud klahvikombinatsioone erinevate stsenaariumi plokkide valimiseks. Valmis stsenaariumi saab PDF kujul eksportida ja alla laadida.

5.2 Kasutajate tagasiside

Rakenduse algidee tuli Horvaatia animafilmistuudio Adriatic Animation produtsendi poolt. Koostöös temaga pandi kirja algsed nõuded rakenduse prototüübile. Kuna rakendus on mõeldud filmitegijatele, siis täiendati neid nõudmisi nii Adriatic Animationis kui teistes animafilmistuudiotest töötavate filmitegijate abiga. Filmitegijad olid huvitatud rakendusest, mis sisaldaks nii süžee arendamist mugavdavaid funktsionaalsusi kui stsenaariumi kirjutamiseks vajalikke võimalusi.

Arendamise käigus paluti rakendust filmitegijatel katsetada ning nende tagasiside ja soovide põhjal viidi sisse parandusi ja täiendusi. Samuti avastati katsetamise käigus mõningaid vigasid, mis autoril programmi arendamisel märkamata jäid, kuid pärast avastamist lahendatud said.

Tagasisidet andnud filmitegijad olid väga rahul kasutajaliidesega ning leidsid, et rakendust on mugav kasutada ning see lihtsustab filmi arendusprotsessi. Leiti, et menüüd ja vaated on loogilised ja intuitiivsed. Kiideti ka võimalust fragmentide sisu ühtseks tervikuks agregeerida ning siltide põhjal loodavat ajajoont ja märgiti, et kuigi need on filmitegijatele väga kasulikud funktsionaalsused, on need sarnastest rakendustest tavaliselt puudu. Samuti olid tagasiside andjad rahul stsenaariumi kirjutamise vaatega, mida peeti intuitiivseks andes võimaluse kasutada stsenaariumikirjutamise tarkvaras üldkehtivaid klahvikombinatsioone.

Kasutajad kiitsid ka filtreerimise võimalust ning mainisid, et see aitab süžee arendamisele tugevalt kaasa, kuid ühe puudujäägina toodi välja piiratud filtrite valik. Hetkel filtreeritakse välja need fragmendid, millel pole ühtki filtris defineeritud silti. Kasutajad leidsid, et fragmente võiks saada filtreerida ka järgnevalt:

- Kuva fragmendid, millel on olemas kõik valitud sildid
- Kuva fragmendid, millel on olemas täpselt valitud sildid
- Kuva fragmendid, millel puuduvad valitud sildid

Lisaks toodi välja, et stsenaariumi kirjutamise vaates võiks näha ka fragmentidele lisatud pilte ja stsenaariumi osad võiksid olla seotud fragmentidega, et filtrid rakenduks ka stsenaariumile.

Kokkuvõttes oli tagasiside positiivne. Kasutajad mainisid, et nad oleksid tulevikus väga huvitatud taolise rakenduse kasutamisest ning et see muudaks süžee arendamise lihtsamaks.

5.3 Edasiarenduse võimalused

Lisaks eelmises alapealtükis mainitud puudujääkide kõrvaldamisele on mitmeid võimalusi rakenduse edasiarenduseks ja täiustamiseks.

Esiteks, kuna tegu on prototüübiga, ei implementeerinud autor kasutajakonto valideerimist e-maili kaudu. Enne rakenduse päriselt kasutuselevõtmist tuleks see kindlasti implementeerida. Samuti tuleks HTTP asemel kasutusele võtta HTTPS.

Arendusprotsessi alguses lepiti kokku, et rakenduse prototüüp võiks esmajoonel olla suunatud laua- ja sülearvuti kasutajatele, sest filmitegijad ei kasuta tavaliselt mobiilseadmeid loo arendamiseks või stsenaariumi kirjutamiseks kuna väikse ekraaniga seadmetes on see ebamugav. Sellegipoolest võiks tulevikus olla rakendus kasutatav ka mobiilseadmetes. Väiksema ekraaniga seadmetele tuleks luua spetsiaalsed vaated, mille jaoks ei jäänud autoril lõputöö raames aega.

Vaadetes, kus kuvatakse kõik fragmendid ning stsenaariumi vaates, mis koosneb stsenaariumi plokkidest võiks kasutusele võtta virtualiseerimistehnikad vältimaks, et rakendus muutub aeglaseks. Lühifilmide puhul seda ei juhtu, sest fragmente ja stsenaariumi plokkide pole väga palju, kuid fragmentide või stsenaariumi plokkide arvu kasvades võib nii juhtuda. Virtualiseerimine tähendab ainult nende elementide lisamist DOM-i (*Document Object Model*), mis kasutajale parasjagu näha on. Kuna stsenaariumi plokkide kõrgus ei ole kindlaksmääratud, siis on virtualiseerimine keeruline. Prototüübi loomise raames oli prioriteediks nõutud funktsionaalsuste loomine ning virtualiseerimine ei mahtunud lõputöö skoopi.

Hetkel saadetakse stsenaariumi salvestamisel tagarakendusele kogu stsenaariumi dokument. Selle asemel võiks implemteerida ainult muudetud osade salvestamise. Selline lähenemine võimaldaks ka automaatse salvestamise implementeerimise, mis terve dokumendi saatmise korral ei ole mõistlik.

Stsenaariumi kirjutamise vaates võiks implementeerida soovitud karakteri või pealdise kirjutamisel. Kasutajale võiks pakkuda loetelu varem mainitud tegelastest kui parasjagu kirjutatud tekst vastab osaliselt varem mainitud tegelase nimele. Selle asemel, et tegelase nime täielikult välja kirjutada, saaks kasutaja valida nimekirjast vajaliku tegelase. Tellijaga lõputöö skoopi arutades lepiti kokku, et lõputöö raames implementeeritakse vaid baasfunktsionaalsused stsenaariumi kirjutamiseks, seega taolised võimalused jäeti lõputöö raames realiseerimata.

Mugavamaks fragmentide vahel navigeerimiseks võiks saada fragmente grupeerida. Kasutaja saaks luua vabalt valitud nimega gruppe, kuhu ta saab fragmente lisada. Grupis olevate fragmentide nimekirja saaks avada ja sulgeda, muutes vajalike fragmentide leidmise lihtsamaks ning vähendades seeläbi müra.

6. Kokkuvõte

Lõputöö eesmärk oli ehitada filmitegijatele suunatud veebirakenduse prototüüp, mis lihtsustaks filmisüžeede loomist ning ühendaks endas rakenduse ideede ülesmärkimiseks, organiseerimiseks ja agregeerimiseks ning stsenaariumi kirjutamise tarkvara. Kasutusel olevad stsenaariumi kirjutamise rakendused ei anna kasutajatele mugavat süžee arendamise võimalust ning märkmete tegemise rakendustel, mida muidu saaks süžee loomisel kasutada, puuduvad muu hulgas funktsionaalsused märkmete agregeerimiseks ning stsenaariumi kirjutamiseks. Samuti on enamik neist vähemalt osaliselt tasulised.

Eesmärgi saavutamiseks pandi koostöös animaafilmivaldkonnas tegutsevate inimestega paika neile kasulikud funktsionaalsused, mis rakendusel peaks olema, analüüsiti kasutuselolevaid lahendusi, valiti välja sobivad tehnoloogiad ning arendati välja rakenduse prototüüp. Arendamise käigus paluti rakendust filmitegijatel testida ning parandati ja täiendati seda vastavalt tagasisidele.

Lõputöö raames arendati välja Reactis kirjutatud eesrakendusest, Spring Bootis kirjutatud tagarakendusest ja PostgreSQL andmebaasist koosnev prototüüp pidades silmas kokkulepitud nõudeid. Tagasiside prototüübile oli positiivne ning mitmed filmitegijad avaldasid soovi taolist rakendust tulevikus kasutada.

Rakendusel on ka mitmeid edasiarendusvõimalusi, mis sisaldavad nii optimeerimisvõimalusi kui uusi funktsionaalsusi kasutajakogemuse veel mugavamaks muutmiseks.

Kasutatud kirjandus

- [1] *Animafilmi arendus ja tootmine*. [Kasutatud: 22-04-2025]. URL: <https://filmi.ee/tegijale/taotlus/animafilmi-arendus-ja-tootmine/>.
- [2] SC Lannom. *How to Write a Movie Script: Screenplay Format and Examples*. [Kasutatud: 22-04-2025]. URL: <https://www.studiobinder.com/blog/how-to-write-a-screenplay/>.
- [3] *Adriatic Animation*. URL: <https://adriaticanimation.hr/>.
- [4] *Final Draft*. URL: <https://www.finaldraft.com/>.
- [5] *Fade In*. URL: <https://www.fadeinpro.com/>.
- [6] *Celtx*. URL: <https://www.celtx.com/>.
- [7] *Arc Studio*. URL: <https://www.arcstudiopro.com/>.
- [8] *React documentation*. [Kasutatud: 27-04-2025]. URL: <https://react.dev/learn>.
- [9] *Render and Commit*. [Kasutatud: 27-04-2025]. URL: <https://react.dev/learn/render-and-commit>.
- [10] *Axios*. URL: <https://github.com/axios/axios>.
- [11] *dnd kit*. URL: <https://github.com/claudeeric/dnd-kit>.
- [12] *Lodash*. URL: <https://github.com/lodash/lodash>.
- [13] *Perfect-freehand*. URL: <https://github.com/steveruizok/perfect-freehand>.
- [14] *React-error-boundary*. URL: <https://github.com/bvaughn/react-error-boundary>.
- [15] *React icons*. URL: <https://github.com/react-icons/react-icons>.
- [16] *Slate react*. URL: <https://github.com/ianstormtaylor/slate>.
- [17] *Zustand*. URL: <https://github.com/pmndrs/zustand>.
- [18] *Spring Booti dokumentatsioon*. URL: <https://docs.spring.io/spring-boot/index.html>.
- [19] *Apache PDFBox*. URL: <https://pdfbox.apache.org/>.
- [20] *Apache Tika*. URL: <https://tika.apache.org/>.
- [21] *Flyway*. URL: <https://github.com/flyway/flyway>.

- [22] *Imgscalr*. URL: <https://github.com/rkalla/imgscalr>.
- [23] *JJWT*. URL: <https://github.com/jwtkt/jjwt>.
- [24] *JUnit*. URL: <https://junit.org/junit5/>.
- [25] *Lombok*. URL: <https://projectlombok.org/>.
- [26] *MapStruct*. URL: <https://mapstruct.org/>.
- [27] *Mockito*. URL: <https://github.com/mockito/mockito>.
- [28] *OpenAPI*. URL: <https://github.com/springdoc/springdoc-openapi>.
- [29] *PostgreSQL indexes*. URL: <https://www.postgresql.org/docs/current/indexes-types.html>.
- [30] *Introduction to JSON Web Tokens*. URL: <https://jwt.io/introduction>.
- [31] *HttpOnly*. URL: <https://owasp.org/www-community/HttpOnly>.
- [32] *Spring Security*. URL: <https://spring.io/projects/spring-security>.
- [33] *What is Long Polling?* URL: <https://www.pubnub.com/guides/long-polling/>.
- [34] *The WebSocket Protocol*. URL: <https://datatracker.ietf.org/doc/html/rfc6455>.
- [35] *Server-sent events*. URL: <https://html.spec.whatwg.org/multipage/server-sent-events.html#server-sent-events>.
- [36] *Long Polling vs WebSockets: What's best for realtime at scale?* URL: <https://ably.com/blog/websockets-vs-long-polling>.
- [37] *Screenplay Format: Everything You Need to Know*. URL: <https://www.nfi.edu/screenplay-format/>.
- [38] *Fountain*. URL: <https://fountain.io/>.

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks¹

Mina, Morten Tšinakov

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose “Veebi-rakendus lühianimafilmide süžee arendamiseks”, mille juhendaja on Siim Rebane
 - 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.

04.06.2025

¹Lihtlitsents 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 loomingulise tegevusega ning lõputöö kaas- või ühisautor(id) ei ole andnud lõputööd kaitsvale üliõpilasele kindlaksmääratud tähtjaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktidele 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.

Lisa 2 - Andmebaasiskeem

Järgnevalt on välja toodud andmebaasi tabelite struktuur. Joonisel 19 on välja toodud kogu tabelite struktuur ning joonistel 20, 21 ja 22 on kujutatud vastavalt kasutajate, projektide ning fragmentidega seonduvate tabelite struktuurid.

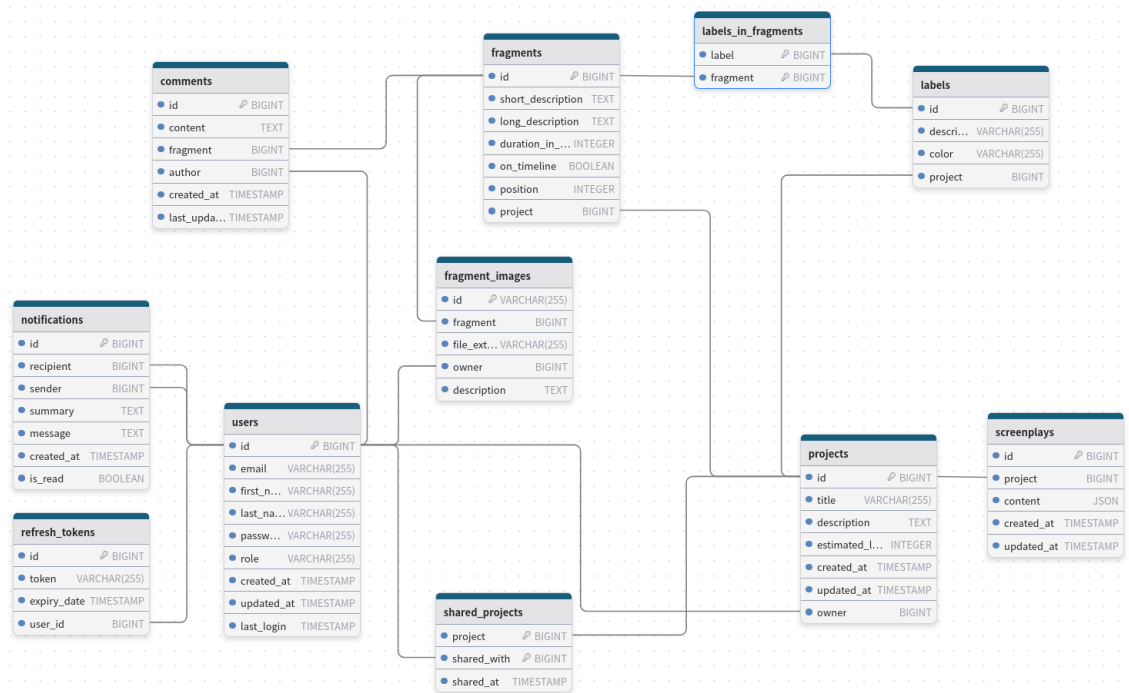


Figure 19. Andmebaasidiagramm



Figure 20. Kasutajaga seostuvad tabelid

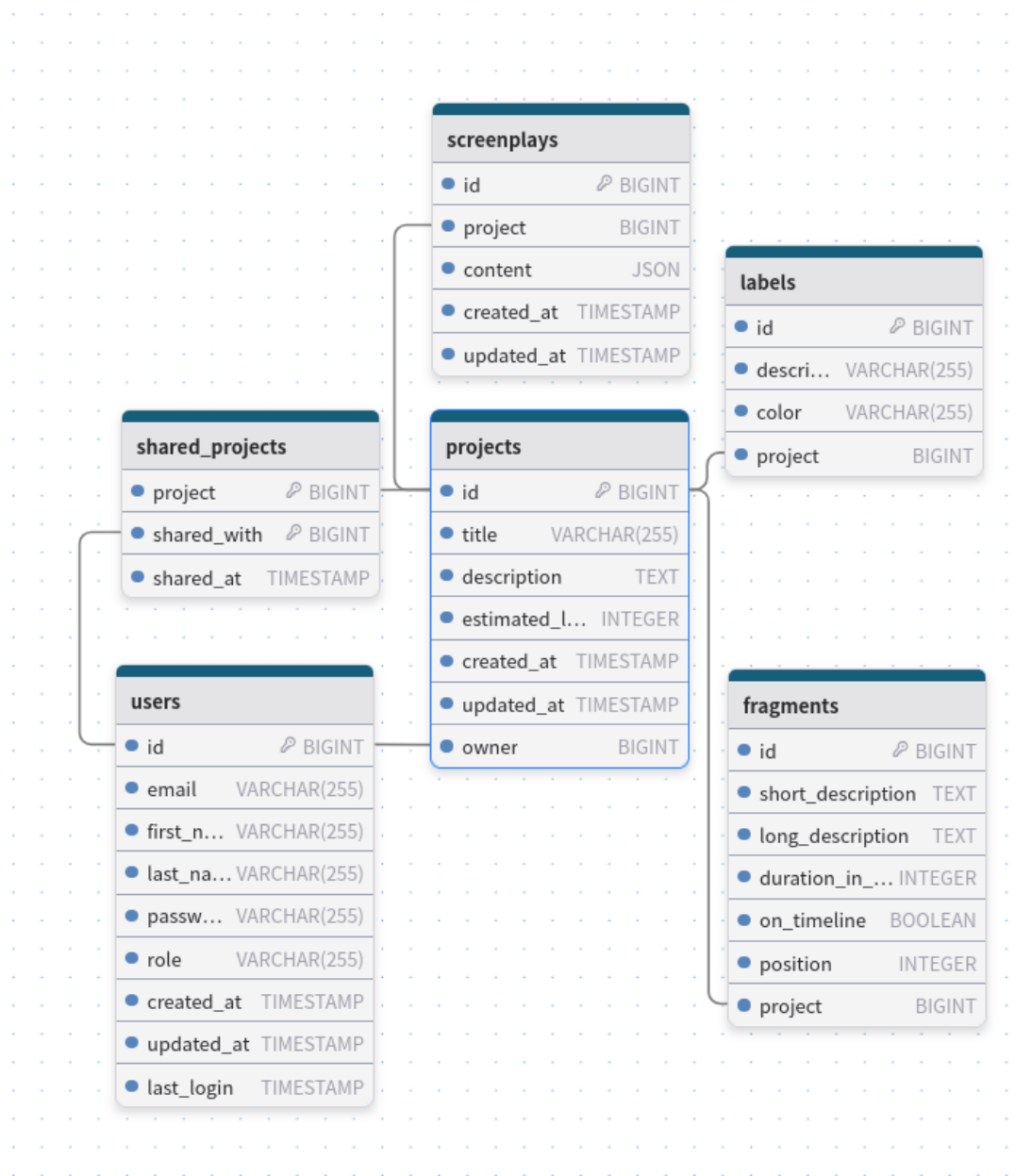


Figure 21. *Projektidega seostuvad tabelid*

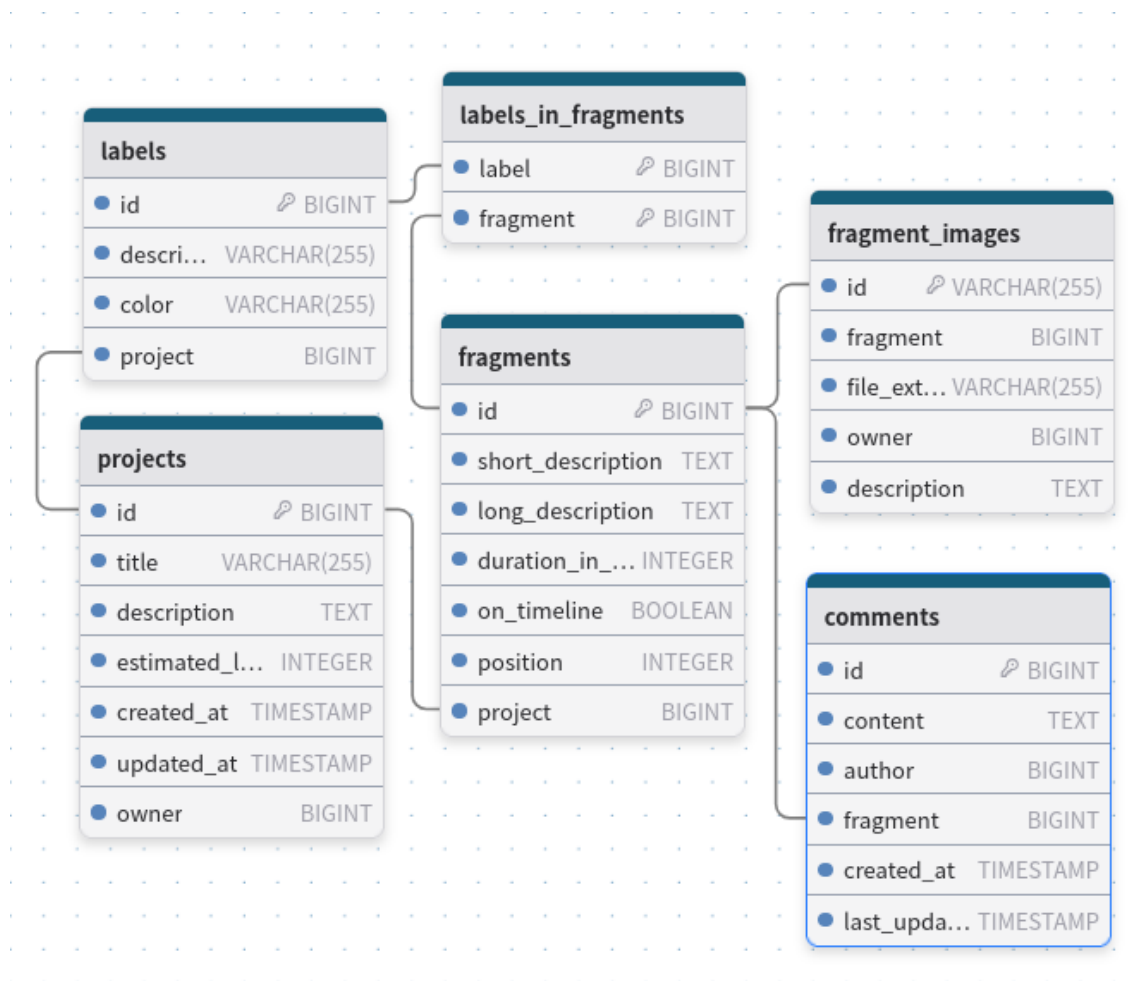


Figure 22. *Fragmentidega seostuvad tabelid*