

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Märten Vuks 179617

ETTEVÕTTE- JA ERARAAMATUKOGUSID HALDAV VEEBIRAKENDUS

Bakalaureusetöö

Juhendaja: Meelis Antoi
Magistrikraad,
Mait Sarv
Bakalaureus

Tallinn 2021

Autorideklaratsioon

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

Autor: Märten Vuks

29.04.2021

Annotatsioon

Käesoleva bakalaureusetöö eesmärgiks on luua väike ja hõlpsasti kasutatav raamatukogu tarkvara, mida on võimalik kasutada ettevõtetes ning isiklikuks tarbeks.

Arenduskäigus luuakse veebirakenduse raamatu lisamise ja inventuuri vaate prototüüp. Arenduskäiku alustati disainist ning prototüüp valmistatakse vastavalt disainile. Prototüübis on võimalik otsida raamatuid käsitsi või skaneerides raamatu triipkoodi. Raamatut on võimalik lisada isiklikku inventuuri, mis kasutab andmebaasi, mis salvestab raamatud veebibrauseri. Veebirakendus on peamiselt mõeldud ettevõtetele kes ostavad raamatuid, et töötajatel oleks võimalus end harida. Rakendus aitab inventuuri hallata ja hoida laenu-
tustel silma peal.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 24 leheküljel, 5 peatükki, 20 joonist.

Abstract

Web Application for Managing Corporate and Private Libraries

The goals of this bachelor's thesis is to develop a web application for small and easy-to-use library software that can be used in corporate offices and for personal use.

During the development process, a prototype of the web application book addition and inventory is created. The development process started with the design and the prototype is made according to the design. In the prototype, you can search for books manually or by scanning the barcode of the book. Books can be added to a personal inventory that uses a database that stores books in a web browser. The web application is designed for companies that buy books so that employees have the opportunity to educate themselves. The application helps to manage inventory and keep an eye on rentals.

The thesis is in Estonian and contains 24 pages of text, 5 chapters, 20 figures.

Lühendite ja mõistete sõnastik

.NET	Microsofti tarkvaraplatvorm
API	<i>Application Programming Interface</i> – rakendustarkvara liides
BEM	Block Element Modifier
CLI	<i>Command-line interface</i> – käsurea kasutajaliides
CSS	<i>Cascading Style Sheet</i> – märgistuskeel, mida kasutatakse veebilehe kujundamiseks
ean13	<i>International Article Number</i> – rahvusvaheline artikli number
<i>Frontend</i>	Kliendirakendus
HTML	<i>HyperText Markup Language</i> – märgistuskeel dokumentide kuvamiseks veebibrowseris
ISBN	<i>International Standard Book Number</i> – raamatute identifitseerimiseks kasutatav numbrikombinatsioon
LESS	<i>Leaner Style Sheets</i> – dünaamiline märgistuskeel
Npm	Node Package Manager – JavaScripti teekide register
PHP	<i>Hypertext Preprocessor</i> - skriptimiskeel
REST	<i>Representational state transfer</i> – tarkvaraarhitektuuri laad
SASS	<i>Syntactically awesome style sheets</i> – märgistus keel mis laiendab CSS-i võimekust
SCSS	<i>Sassy Cascading Style Sheet</i> – märgistuskeel uuema süntaksiga
SPA	<i>Single-page applicatison</i> - üheleherakendus
SQL	<i>Structured Query Language</i> – andmebaasi suhtlus keel
UDK	<i>Universal Decimal Classification</i> – kümnendjaotusel põhinev rahvusvaheline liigitussüsteem
Wasm	WebAssembly

Sisukord

1	Sissejuhatus	9
2	Analüüs.....	10
2.1	Visioon.....	10
2.2	Loodava lahenduse skoop.....	10
2.3	Nõuded.....	11
2.4	Olemasolevad lahendused.....	12
2.4.1	Ester.....	12
2.4.2	Campus Book Rentals	13
2.4.3	Libib	13
2.4.4	Olemasolevate lahenduste kokkuvõte	14
2.5	Tehnoloogia analüüs	15
2.5.1	Eesrakenduse valik	15
2.5.2	Tagarakenduse valik	16
2.5.3	Andmebaasi valik	17
2.5.4	Disain tarkvara valik.....	17
2.6	Analüüsi kokkuvõte	18
3	Rakenduse prototüübi realiseerimine	19
3.1	Prototüübi disain Figma abil.....	19
3.2	Klientrakenduse lahendus	22
3.2.1	Anglar keskkonna ülespanek.....	22
3.2.2	Komponendid ja vaated	23
3.2.3	Bootstrap ja märgistuskeel.....	24
3.2.4	Raamatu otsimine käsitsi	25
3.2.5	IndexedDB liidestamine	26
3.2.6	Triipkoodi lugeja	27
4	Tulemused	29
4.1	Edasiarendus	29
5	Kokkuvõte	30
	Kasutatud kirjandus	31
	Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks	33

Lisa 2 – IndexedDB raamatu tabeli kirjeldus	34
Lisa 3 – JavaScripti ja WebAssembly suhtlus.....	35

Jooniste loetelu

Joonis 1. Skoopi skeem	11
Joonis 2. Ester otsingu avaleht [1].....	12
Joonis 3. Campus Book Rentals raamatute inventuuri vaade [2].....	13
Joonis 4. Libib raamatu skaneerimise kuvatõmmis.....	14
Joonis 5. Raamatu otsimise valiku vaade	20
Joonis 6. Raamatu skaneerimine	20
Joonis 7. Leitud raamatu tulemuse vaade.....	21
Joonis 8. Raamatute inventuuri disaini prototüüp	21
Joonis 9. Angular CLI installimine.....	22
Joonis 10. Projekti genereerimise käsk.....	22
Joonis 11. Lisa seadistused projektile	23
Joonis 12. Angulari rankenduse käivitamise käsk.....	23
Joonis 13. Algrakenduse avaleht	23
Joonis 14. Raamatute inventuuri komponendi jaotus.....	24
Joonis 15. SCSS muutujad ja BEM struktuur	25
Joonis 16. Raamatu nimekirja GET päring	26
Joonis 17. IndexedDB ühenduse loomine	27
Joonis 18. Meetod raamatute saamiseks andmebaasist	27
Joonis 19. Kasutajalt video pildi luba küsimine.....	28
Joonis 20. Triipkoodi lugeja välja kutsumine raamatu otsimise vaates	28

1 Sissejuhatus

Erialakirjandust kasutavad ja ostavad ettevõtted, et töötajad oleks võimalik end harida, kuid miks piirduda ainult oma töölistega. Ettevõtted kes oleksid nõus oma raamatukogust erialakirjandust välja laenutama inimestele kes ei tööta antud ettevõttes. Probleem tuleneb sellest, et inimesed kes tahavad endale raamatut laenutada, kuid ei ole kursis sellega, kas keegi on sama raamatu endale laenutanud.

Eesmärk on luua väike ja hõlpsasti kasutatav raamatukogu tarkvara, mida on võimalik kasutada ettevõtetes ning isiklikuks tarbeks. Põhirõhk on erialakirjandusel (peamiselt tehnikakirjandus), mida ostetakse ettevõtete poolt ning mis on mõeldud töötajatele eneseharimiseks. Enamasti arendus- ja / või tootmisettevõtted, kus on vaja palju tehnilisi teadmisi. Rakendusega saab paremat ülevaadet firmale olemasolevatest erialakirjandustest mida saavad firma töötajad laenutada.

2 Analüüs

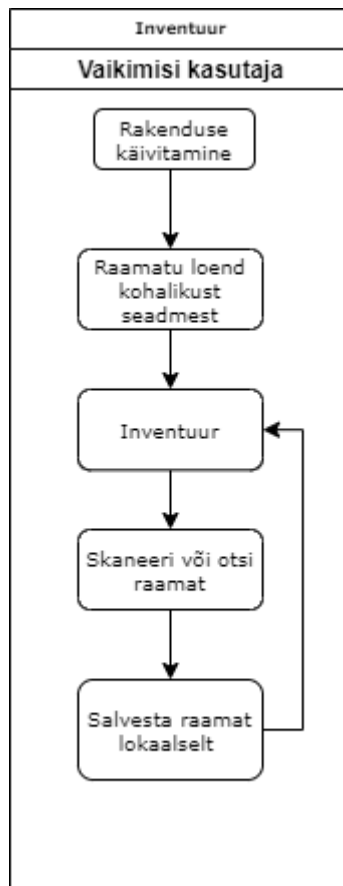
Analüüsi esimeses pooles kirjeldatakse lõputöö visiooni, eesmärki ja skoopist. Seejärel on välja toodud nõuded loodavale lahendusele ja tulevikus loodavad funktsionaalsused. Lõputöö lahenduses võetakse aluseks välja selgitatud nõuded, võrreldakse juba olemasolevaid lahendusi ja tuuakse välja puudujäägid. Samuti valitakse välja tehnoloogiad mis on vajalikud lahenduse ehitamiseks.

2.1 Visioon

Töö visioon on algust teha veebirakenduse, kliendipoolse koodi ehitamisega, mis aitab ettevõtte- ja eraraamatukogusid hallata. Veebirakenduses saab lisada nimekirja raamatuid mis on ettevõttel või eraisikul, et saada parem ülevaade olemasolevatest raamatutest. Samuti võimaldab rakendus ilma interneti ühenduseta nimekirja kuvada ja seda hallata ning organiseerida ettevõtte siseselt raamatute laenutust. Rakenduse ehitamist alustatakse prototüübi disainimisest, raamatu lisamise funktsionaalsusest läbi triipkoodi skanneri, võimalusel käsitsi, seejärel lisatakse raamat inventuuri. Hiljem on võimalik näha inventuuris salvestatud raamatuid ka ilma interneti ühenduseta.

2.2 Loodava lahenduse skoop

Lõputöö käigus valmiv skoop on kliendi poolel ehitada prototüüp raamatu lisamine inventuuri kasutades triipkoodi skaneerimise funktsionaalsust ning võimalust otsida raamat käsitsi (Joonis 1). Funktsionaalsuse juurde jääb IndexedDB ja triipkoodi lugeja liidestamine eesrakendusega. Skoopi jääb lahenduse esialgse kasutajaliidese disainimine.



Joonis 1. Skoopi skeem

2.3 Nõuded

Nõuete määramisel räägiti kaasjuhendajaga läbi milliseid nõudeid on loodava rakendusele vajalikud lähtudes skoobist.

Funktsionaalsed nõuded:

1. Käsitsi otsimine andmebaasist
2. Triipkoodi kaudu raamatu leidmine andmebaasist
3. Otsitud raamatu lisamine inventuuri
4. Salvestatakse lokaalses seadmes
5. Vale raamatu leidmisel tühistamise võimalus

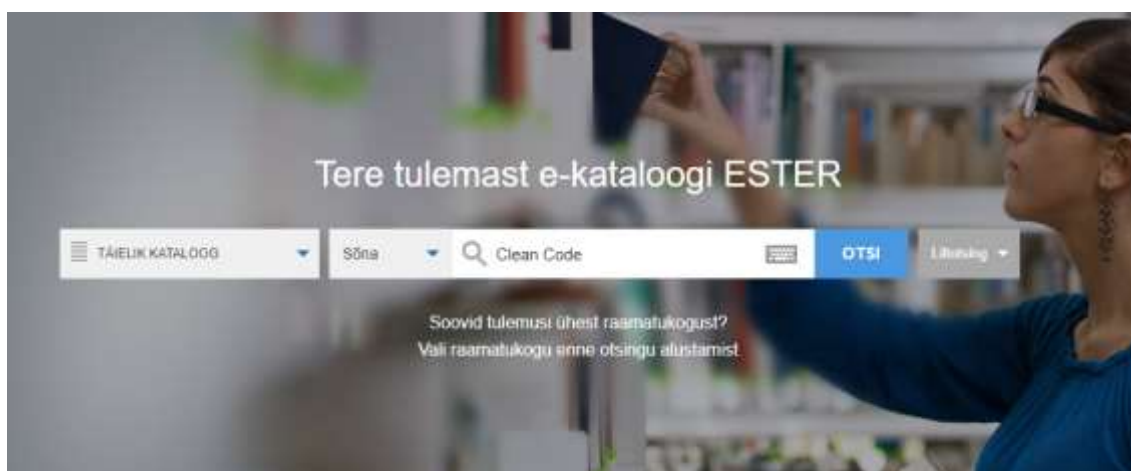
Mittefunktsionaalsed nõuded:

1. Leht peab olema inglise keeles
2. Nõusolek kaamera kasutamiseks
3. Raamatul triipkoodi olemas olek
4. Nimekirja kättesaadavus ka ilma internetiühendusega
5. Võimalikkus veebirakendust kasutada nii arvutis kui ka mobiilis
6. Triipkoodi loetakse mobiili kaamera abil

2.4 Olemasolevad lahendused

2.4.1 Ester

Ester on veebikeskkond, mis on e-kataloog, kus on Eesti suurimate raamatukogude ühis-kataloog. Kataloogis leidub andmeid ülikoolide kolledžite ja erialaraamatukogudes leiduvate raamatute kohta. Veebileht sisaldab kirjeid raamatute, perioodikaväljaannete, nootide, helisalvestite, e-ressursside jms kohta. Kataloogist on võimalik otsida erinevate tunnuste abil nt autor, pealkiri, märksõna, UDK liik, ISBN. Veel on võimalik valida kas otsida täielikust kataloogist või endale sobivast raamatukogust (Joonis 2) [1]. Antud veebirakenduses ei ole võimalik raamatuid hallata ja inventuuris hoida.



Joonis 2. Ester otsingu avaleht [1]

2.4.2 Campus Book Rentals

Campus Book Rentals on veebirakendus mis asub Ameerikas, kus registreerujatel on võimalik otsitud raamatuid laenutada. Selle veebirakenduse peakontor asub Nebraska osariigis Lincolni linnas ja on tegutsenud alates 2007. aastast. Lehel on võimalik raamatuid otsida samuti ISBN, autori, pealkirja või märksõna abil. Sobiva raamatut on võimalik nii laenutada kui ka endale osta. Laenutamise aega on võimalik ise valida, sellest sõltub ka laenutusehind (Joonis 3). Peale laenutamist on võimalik seesama raamat endale päriseks osta [2]. Antud veebirakenduses ei ole võimalik raamatuid hallata ja inventuuris hoida.



Joonis 3. Campus Book Rentals raamatute inventuuri vaade [2]

2.4.3 Libib

Libib on raamatukogu halduse teenus mis on suunatud rohkem eraisikutele ja väikestel organisatsioonidele. Süsteem lubab teha erinevaid kogusi, kuhu on võimalik lisada kuni 5000 eset. Raamatuid saab otsida ISBN, autori, pealkirja või märksõna abil ning lisada, on olemas CSV tabelite import. Lisaks saab raamatuid otsida läbi mobiilirakenduses oleva triipkoodi skaneerija (joonis 4). On olemas tasuline versioon, mis maksab alates 9€ kuus, mis võimaldab lisada rohkem esemeid nimekirja ja aitab laenutatud raamatutel silma peal hoida [3].



Joonis 4. Libib raamatu skaneerimise kuvatõmmis

2.4.4 Olemasolevate lahenduste kokkuvõte

Sarnaseid lahendusi on mitmeid, rakendused, mille eesmärgiks on ainult raamatute laenutamine ja haldamine. Kõige sarnasem olemasolev lahendus on Libib, kus on võimalik hallata raamatuid ning tasulises versioonis neid laenutada ja laenutamistel silm peal hoida.

Pärast mitmete raamatute lisamist ja otsimist haldavates süsteemides, oli puuduseks eesti keelsete raamatute leidmine andmebaasidest. Raamatuid on võimalik lisada küll käsitsi täites vormi, kuid see täitmine on ajakulukas. Prooviti Libib raamatu triipkoodi skaneerimist, see on kiire, aga juhul kui raamat ei ole andmebaasis, on triipkoodi skaneerimine ajakulukas protsess. Autor proovis triipkoodi skaneerimist andmebaasis mitte oleva raamatuga, raamatut otsiti ligikaudu 30 sekundit ning otsimise käigus ei olnud võimalik otsimise akent kinni panna.

Suurimaks erinevuseks loodava rakenduse ja olemasolevate rakenduste vahel oleks see, et loodavat rakendust on võimalik kasutada ka ilma interneti ühenduseta ning andmebaas leiduks rohkem eesti keelseid teoseid. Kasutades IndexedDB andmebaasi mis võimaldab salvestada asju veebibrauseris. Andmebaasi valikut kirjeldatakse hiljem andmebaasi valiku peatükis.

2.5 Tehnoloogia analüüs

Veebirakenduse arendamisel on võimalik valida paljude programmeerimiskeelte ja tehnoloogiate vahel [4]. Autor võtab vaatluse alla levinumad programmeerimiskeeled ja tehnoloogiad mis on hästi dokumenteeritud ja millel on suur kasutajabaas.

Andmete vahendamiseks kliendi ja serveri vahel võetakse kasutusele REST (*Representational State Transfer*). 2000. aastal tutvustas Roy Fielding uut lähenemist kuidas veebi-teenusega suhelda [5].

2.5.1 Eesrakenduse valik

Kliendirakenduse lahenduse puhul tuleb kaaluda erinevaid Javascripti raamistike ja abistavaid pakette mis aitavad rakendust kiiremini ja lihtsamini ehitada.

Populaarsemad raamistikud ja abistavad paketid:

- Angular – platvorm ja raamistik mis kasutab HTML ja Typescripti SPA kliendirakenduste loomiseks [6].
- Vue.js – progressiivne raamistik kasutajaliideste ehitamiseks. Lihtne kasutada ja kergesti integreerida teiste olemas olevate JS pakettidega ja projektidega [7].
- React – Javascripti pakett kasutajaliidese ehitamiseks [8].
- Bootstrap – CSS raamistik mis aitab kiiremini ja lihtsamini ehitada kasutajaliidest kergema vaevaga ja saada rakendus, mis on skaleeruv nii veebibrauseris kui ka mobiilis [9].
- Webassembly – uut tüüpi kood, mida saab kasutada kaasaegsetes veebibrauserites JavaScripti kõrval ja mis pakub uusi funktsioone ning suuremat jõudlust. Pole mõeldud käsitsi kirjutamiseks tõhusaks kompileerimise eesmärgiks lähtekeeltele nagu C, C ++, Rust [10].

Lõputöö kirjutaja omab oskust Vue.js ja Angular osas, kuid need on ka minimaalsed. Mõlema raamistiku baasil on loodud palju õppematerjali ja hästi dokumenteeritud dokumentatsioon, mille abil on kerge veebirakendust püsti seada, kuid hetkel on Angular varem ja stabiilsem, seega on eelistatud valik Angular. Lisaks võtab kasutusele Bootstrapi

mis aitab lihtsamalt ehitada skaleeruvaid vaateid ja Webassembly, mis aitab triipkoodi lugejat ehitada.

2.5.2 Tagarakenduse valik

Serveripoolseks sobivaid programmeerimiskeeli on palju, kuid välja tuuakse levinumad. Veel tuuakse välja programmeerimiskeeled, mis sobivad kokku Webassembly'ga.

- PHP – vabavaraline dünaamiline - ja skriptimiskeel, mis on loodud veebilehetede arenduseks, mida on võimalik HTML'i siduda ning on lihtne kasutada [11].
- Java – objekt-orienteeritud keel, mis jookseb mitmetel platvormidel [12]. Üks levinumaid programmeerimiskeeli maailmas [13], mille tõttu on kergesti ehitatav kuna on hästi dokumenteeritud.
- C# – moderne, objekt-orienteeritud keel. C# võimaldab ehitada mitut tüüpi turvalisi rakendusi, mis töötavad .NET ökosüsteemis [14].
- C – on üldotstarbeline programmeerimiskeel, mis on väga populaarne. Struktureeritud keel, mis kasutatakse operatsioonisüsteemide (näiteks Windowsi) ja erinevate keerulisemate programmide kirjutamiseks [15].
- C++ – on riistvaralähedane, mis võimaldab protsessoriga manipuleerida madalamal tasemel, mis suurendab programmi käivitamise kiirust, kuid hiljem on vaja reserveeritud ruum vabastada [16].
- Rust – Tänapäevane süsteemide programmeerimiskeel, mis keskendub ohutusele, kiirusele ja samaaegsusele [17].

Tagarakenduse poolel tuleb kasutusele PHP, kuna see on skriptimiskeel, mis võimaldab ülesandeid kergesti lahendada ning on piisava pika ajalooaga, et on palju stabiilseid pluginaid. PHP-s tuleb kasutusele Yii 2 raamistik.

Hetkel on peamiselt 3 programmeerimiskeelt mis kompileerub hästi Webassembly'ga. Lõputöös võetakse kasutusele Rust. Rust on uuem ning on juba hinnatum ja armastatum kui C/C++ [18].

2.5.3 Andmebaasi valik

Andmete hoiustamiseks tuleb valida erinevate andmebaaside ja tüüpide vahel [19]: objekt-orienteeritud, relatsioonilised, võti-väärtus ning dokument-orienteeritud andmebaasid. Vaatuse alla tulevad tasuta ehk avatud lähtekoodiga andmebaasid.

- SQLite – relatsiooniline serverita andmebaas, millel on limiteeritud andmebaasi suurus, kuid on kergesti õpitav ja hallatav [20].
- MySQL – kõige populaarsem relatsiooniline andmebaas [21].
- MariaDB – üks populaarsem relatsiooniline andmebaas mis on tehtud MySQL tegijate poolt, mis on kiirem kui MySQL. MariaDB on loodud kiiruse, töökindluse ja kasutajamugavuse jaoks [22].
- PostgreSQL – objekt relatsiooniline andmebaas millel on suur kasutajasbaas. Paljud avatud lähtekoodiga rakendused ei toeta PostgreSQL ning on aeglasem kui Mysql [23].
- MongoDB – väga skaleeruv ja paindlik, dokument-orienteeritud NoSQL andmebaas, mis koosneb võti-väärtus paaridena, mida kasutatakse mahukate andmete salvestamiseks [24].
- IndexedDB – salvestab võti-väärtus paaridena kasutaja veebibrauserisse. Kasulik rakenduste jaoks mis salvestavad suurt hulka andmeid ja ei vaja töötamiseks püsivat interneti-ühendust [25].

Avatud lähtekoodiga andmebaase on palju mille seast valida endale parim. Serveri poolel tuleb kasutusele MariaDB, kuna see on piisavalt stabiilne, relatsiooniline, toetab SQL tabeleid kui ka NoSQL tüüpi tabeleid. Võimaldab hierarhilist struktuuri kergesti ehitada ning plaanis on kasutada OQGRAPH pluginat.

Frontend poolel tuleb kasutusele IndexedDB, mis aitab salvestada inventuuris olevad raamatud veebibrauseri, et hiljem oleks võimalik ilma interneti ühenduseta inventuurile ligi pääseda.

2.5.4 Disain tarkvara valik

Valikuid kasutajaliidese disainimiseks on palju [26].

- Figma – on ühe tööriista lahendus, mis koosneb võimsatest kaasaegsetest funktsioonidest, kus meeskonnal on võimalik koostööna viia algusest lõpuni paindlik ja ligipääsetav kujundus [26]. Tasuta versioonis saab korruga tegutseda 2 disainerit ja on võimalik teha 3 projekti piiramatul pilvemäluga ning 30 päeva dokumendi ajaloo [27].
- Adobe XD – vektoripõhiline veebilehtede ja rakenduste kujundamise tööriist. Tasuta versioonis saab tegutseda 2 disainerit ja teha 1 projekti 2 GB pilvemäluga ja 10 päeva dokumendi ajaloo [28].
- Webflow – Võimaldab ehitada reageerivaid brauseripõhiseid veebisaite, disainides genereerib Webflow automaatselt HTML-i, CSS-i ja JavaScripti. Tasuta versioonis võimalik teha 2 projekti, kuid tasuta versioonis pole võimalik genereeritud koodi eksportida [29].
- Axure RP – professionaalne kasutajakogemuse disaini tööriist, mis võimaldab luua paindlikke prototüüpe nii veebisaitidele kui ka rakendustele. Tasuline versioon lubab piiramatuid prototüüpe ja 1000 projekti ning üliõpilastel on võimalik seda kasutada tasuta [30].

Lõputöö autoril on kogemusi Figma disainimis tööriistaga. Võrreldes teiste kasutajate liidestega tasuta versioonidega, võimaldab antud rakenduse tasuta versioon enim, seetõttu osutus ka valituks.

2.6 Analüüsi kokkuvõte

Analüüsi käigus vaadeldi erinevaid tehnoloogiaid ja käsitleti funktsionaalseid ja mittefunktsionaalseid nõudeid.

Serveripoolseks keelteks osutus valituks PHP, kuna see on skriptimiskeel, mis võimaldab ülesandeid kergesti lahendada ning on piisava pika ajaloo ning Rust. Serveripoolseks raamistikuks on valitud PHP põhine Yii 2 ja mikroteenustes tuleb kasutusele Rust Actix Web raamistik. Kliendipoolse rakenduse ehitamiseks valiti Angular, kuna on piisavalt stabiilne ja on olemas piisavalt õppematerjale. Andmebaasideks võetakse kasutusele avatud lähtekoodiga MariaDB ja IndexedDB. Andmete vahetamiseks tuleb kasutusele REST. Prototüübi disainimisel võetakse kasutusele Figma.

3 Rakenduse prototüübi realiseerimine

Rakenduse ülesehitus on jagatud kaheks: raamatu skaneerimise prototüübi disainimine ja selle realiseerimine.

3.1 Prototüübi disain Figma abil

Prototüüp aitab mõista ja uurida võimalusi ning piiranguid, mis ilmnevad siis, kui need on visuaalse ja interaktiivse lahendusena üles ehitatud. Prototüüpimine võimaldab veel kiirelt katsetada mitmeid erinevaid ideid ja stsenaariume. Prototüüpimisel saadud tulemus ei tähenda lõpliku disaini, vaid annab paremat ülevaadet, kuidas elemendid tuleb paigutada [31].

Prototüübi disainimist alustati uuest projektist, mille nimeks pandi „Enlivoy“, mis tuleb veebirakenduse nimeks. Alustuseks võeti uus mobiili raam ettenähtud nimekirjast, raam tuleb 414 pikslit lai ja 812 pikslit kõrge. Järgmiseks tuleb paika panna päis, ning disainitud päist saab igas järgmises vaates taaskasutada. Veel tuleb lisada nupp, mis avab raamatu skaneerimise funktsionaalsuse ja sisend, millega otsida raamat andmebaasist, sisestades raamatu pealkirja või ISBN numbrit (joonis 5).



Joonis 5. Raamatu otsimise valiku vaade



Joonis 6. Raamatu skaneerimine

Joonisel 6 on näha raamatu skaneerimist. Enne skaneerimist küsitakse kasutajalt luba rakendusel kasutada kaamerat. Skaneerimist on võimalik lõpetada üleval paremast olevast ristist.



Joonis 7. Leitud raamatu tulemuse vaade

Kui triipkoodilt saadud ISBN annab tulemuse või käsitsi on leitud raamat siis tuleb ette raamatu detailsem vaade, kus on kuvatud pealkiri, autor ja väljalaskeaasta ning on võimalik valida, kas raamat lisada inventuuri või tühistada ning uuesti skaneerida (Joonis 7).



Joonis 8. Raamatute inventuuri disaini prototüüp

Peale raamatu salvestamist liigub raamat inventuuri, kus on näha teisi salvestatud raamatuid (Joonis 8). Vajutades luubi peale tuleb otsinguriba, kus on võimalik inventuurist otsida raamatuid. Paremalt üleval paremal nurgas väike ikoon, mis viib raamatu otsimise vaatele.

3.2 Klientrakenduse lahendus

Klientrakenduse arenduse tulemusena loodakse prototüüp raamatu lisamisest inventuuri kasutades triipkoodi skaneerimist või otsides käsitsi. Veebirakendus luuakse kasutades Angular raamistikku.

Lõputöö baasrakendus on loodud kasutades Angular CLI liidest, mis on Angulari enim kasutatud baasrakenduse genereerija, täpsemalt kirjeldatakse järgmises peatükis.

3.2.1 Anglar keskkonna ülespanek

Angulari keskkonna aitab paigaldada Angular CLI [32]. Selleks tuleb arvutisse tõmmata Node.js ja npm (*Node package manager*). Node.js on platvorm kiirete ja skaleeritavate võrgurakenduste hõlpsaks ehitamiseks [33], ning npm on maailma suurim tarkvararegister kust saab jagada ja kasutada JavaScripti pakette [34]. Läbi npm ja käsurea saad endale tõmmata Angulari CLI (joonis 9) ning omakorda läbi CLI saada endale Angulari keskkonna.

```
C:\Users\vuksm\OneDrive\Töölaud>npm install -g @angular/cli
```

Joonis 9. Angular CLI installimine

Läbi käsurea kasutades käsku „ng“ on võimalik kasutada Angulari CLI-d ning lihtsa käsuga „ng new projekti nimi“ saab genereerida baas rakenduse (joonis 10).

```
C:\Users\vuksm\OneDrive\Töölaud>ng new <projekti nimi>_
```

Joonis 10. Projekti genereerimise käsk

Käsu käivitamisel saab veel lisada 2 seadistust ja valida millist märgistuskeelt tahad projektis kasutada, valikus on CSS, SCSS, SASS, LESS ja Stylus. Valituks osutus SCSS (joonis 11).

```
C:\Users\vuksm\OneDrive\Töölaud>ng new projekti-nimi
Do you want to enforce stricter type checking and stricter bundle budgets in the workspace?
This setting helps improve maintainability and catch bugs ahead of time.
For more information, see https://angular.io/strict Yes
Would you like to add Angular routing? Yes
Which stylesheet format would you like to use? SCSS [ https://sass-lang.com/documentation/syntax#scss
```

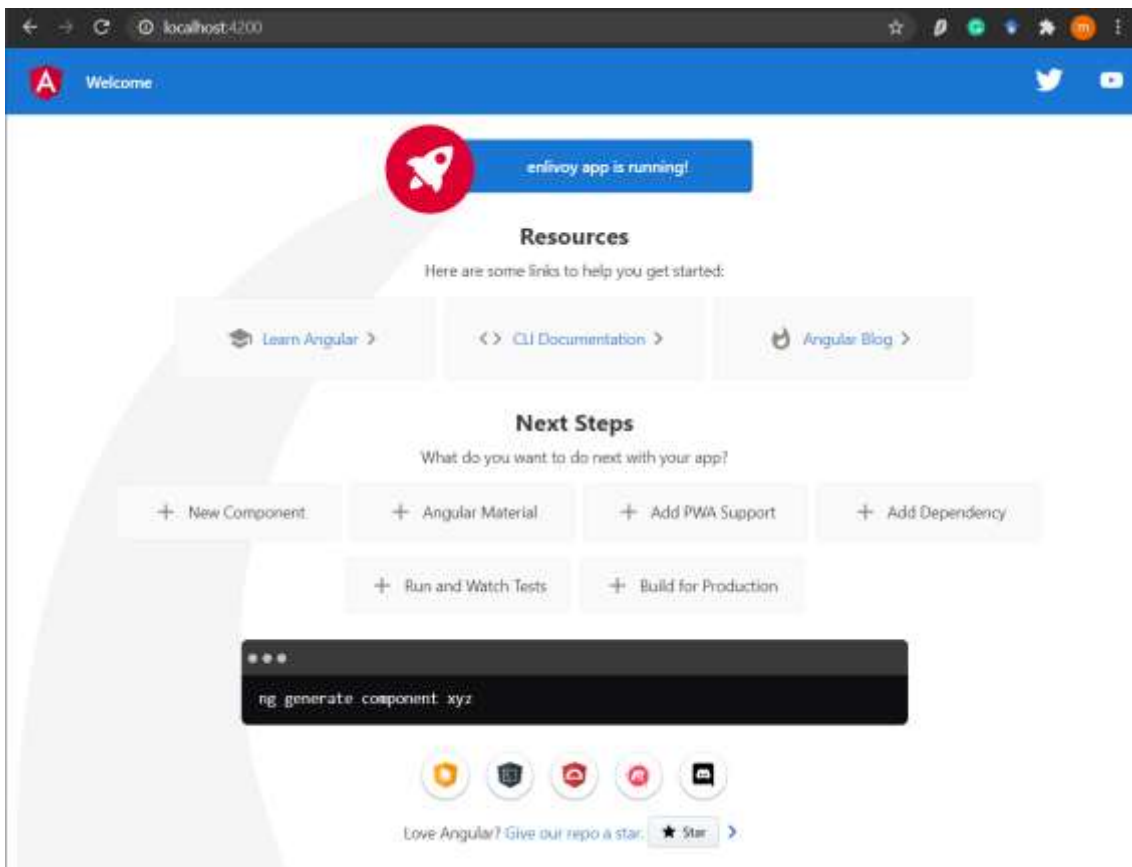
Joonis 11. Lisa seadistused projektile

Rakenduse käivitamiseks tuleb kasutada käsku „ng serve“ olles käsuraal rakenduse juur kataloogis (joonis 12).

```
C:\Users\vuksm\OneDrive\Töölaud\Enlivoy>ng serve
```

Joonis 12. Angulari rakenduse käivitamise käsk

Rakendus töötab port 4200 peal ja võimalik kätte saada aadressilt „localhost:4200“. Avalehel on viiteid Angulari dokumentatsioonile ja palju teistele kasulike materjalidele (joonis 13).



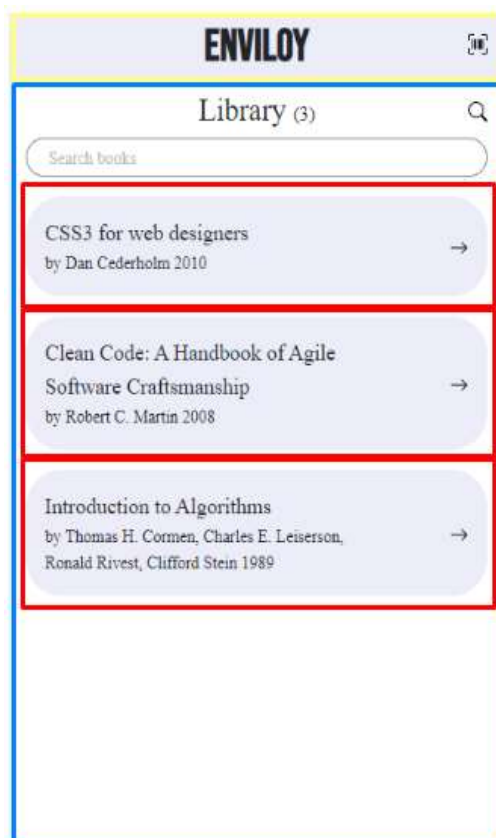
Joonis 13. Algrakenduse avaleht

3.2.2 Komponentid ja vaated

Komponendid on Angulari peamised ehitusblokid. Iga komponent koosneb HTML-mallist, Typescriptist, ja CSS failist, mis rakendatakse valikuliselt mallile. Komponenti saab luua läbi Angulari CLI kasutades käsku „ng generate component <komponendi nimi>“.

Komponente on kahte tüüpi olekuta (*stateless*) ja olekuga (*stateful*). Olekuta komponendid on need, mis edastavad edasi lihtsalt teavet, sellel komponendil pole teadmisi mineviku, praeguse ega tuleviku olekumuutuste kohta. Ning olekuga komponendid on need, millel on teadmisi mineviku, praeguse ja tuleviku olekumuutuste kohta [34].

Lõputöö skoobi lahendamisel on kasutusel enamasti olekuga komponendid. Joonisel 14 on näha „raamatute inventuuri“ vaadet ning on ära märgitud erinevad komponendid. Kollane kast viitab navigatsiooni komponendile, kus erinevates vaadetes näitab erinevaid võimalusi. Inventuuri vaates tuleb raamatu skaneerimise ikoon, mis viib raamatu skaneerimise vaatele. Sinine raam viitab raamatute inventuuri komponendile, kus on raamatu otsimiseks väli ning selle sees on punane komponent, mille ülesandeks on raamatute info teha visuaalselt ilusaks ja loetavaks.



Joonis 14. Raamatute inventuuri komponendi jaotus

3.2.3 Bootstrap ja märgistuskeel

Bootstrap on tasuta populaarseim HTML, CSS ja JavaScripti raamistik reageerivate veebirakenduste ehitamiseks, mis on loodud endiste Twitteri töötajate poolt [37]. Lõputöös

tuleb kasutusele Bootstrapi 5, selle saamiseks tuleb lisada koodi 2 rida koodi, mida saab Bootstrapi kodulehelt [35].

Kasutusele tuleb CSS asemel SCSS mis on uusim ja rohkemate võimalustega märgistuskeel. SCSS lubab oma uute võimalustega kasutada muutujaid, mida saab hiljem taaskasutada läbi mitmete SCSS failide. Kasutusele tuleb veel BEM (Block, Element, Modifier), mis on populaarne klasside nimetamise tava (Joonis 15) [36].

```
$list-bg: #ecedf9;
$sm-font-size: 14px;

.listitem {
  background-color: $list-bg;
  padding: 1rem;
  border-radius: 30px;
  margin-top: 1rem;
}

.listitem__section {
  font-size: $sm-font-size;
}
```

Joonis 15. SCSS muutujad ja BEM struktuur

3.2.4 Raamatu otsimine käsitsi

Käsitsi otsimiseks ehitati valmis uus vaade, kus on sisendikast, kust on võimalik otsida raamatuid kas ISBN numbri või raamatu pealkirja kaudu. Raamatu leidmiseks saadetakse tagarakendusele API GET päring koos otsingu sõnega ning tagarakendus tagastab raamatute loendi, kus on kajastatud otsingu sõne (joonis 16).

```
export class BooksearchService {
  private serverApiUrl: string;

  constructor(private http: HttpClient, appConfigService: AppConfigService) {
    this.serverApiUrl = appConfigService.serverApiUrl;
  }

  searchBook(str): Observable<Book[]> {
    var searchFormat = str.match(/^(\\d{13})\\s*?$/)? "ean13" : "str";
    return this.http.get<Book[]>(`${this.serverApiUrl}/book/find=${searchFormat}=${str}`);
  }
}
```

Joonis 16. Raamatu nimekirja GET päring

Eesrakenduse kasutaja saab valida endale õige raamatu pealkirja järgi tagastatud loendist. Peale raamatu valiku tegemist näidetakse detailsemat vaadet kus on raamatu pealkiri, autor(id) ja väljalaskeaasta. Soovitud raamatu puhul on võimalik lisada see inventuuri või vale raamatu puhul minna tagasi otsingu vaatesse.

3.2.5 IndexedDB liidestamine

IndexedDB võimaldab salvestada suuri koguseid andmeid kasutaja *browserisse*. Andmebaas on objektorienteeritud, kus saab andmeid hoiustada võti-väärtus paaridena, väärtused võivad olla keerukad struktureeritud objektid. IndexedDB API on täiesti asünkroonne [25]. Lõputöö raames tuleb kasutusele väline teek mis teeb IndexedDB API suhtluse kergemaks.

IndexedDB API on üles ehitatud põhimõttel, et andmebaasi avades tuleb ette anda versiooni number ja kui see muutub võrreldes hetke versiooniga, siis kutsutakse välja *onupgradeneeded* meetod, mille sees tuleb baas viia soovitud struktuurile. Klassikaliselt lähendatakse seda tingimusreeglitega, kus vastavalt igale versioonile tehakse andmebaasi struktuuri muudatus. Antud teek läheneb struktuuri uuendustele teistmoodi. Kõigepealt tuleb ette anda kogu andmebaasi struktuur, koos tabeli versioonidega ja siis teegi sees tehakse andmebaasi uuendused vajadusel automaatselt ära. See võimaldab alati koodis näha hetke baasi struktuuri ja hoiab ära pika tingimusreeglite jada, mis võib andmebaasi arenedes tekkida. Teeki on lisatud veel võimalus otsida mitut kirjet korraga indeksi kaudu.

Andmebaasiga suhtlemiseks tuleb luua andmebaasiga ühendus, andes kaasa andmebaasi nimi ja kogu andmebaasi struktuur (joonis 17).

```
export class IndexeddbService {  
  myIdb: IndexedDBModelCache;  
  constructor() {  
    this.myIdb = new IndexedDBModelCache('my-idb', IndexedDbStores.storeList);  
  }  
}
```

Joonis 17. IndexedDB ühenduse loomine

Loomisel antakse kaasa raamatute struktuur, struktuuris on kirjeldatud tabeli nimi, primaarne võti, indeksid, versioon ja tüüp mis on raamatu objekt. Raamatu objektis on raamatu pealkiri, avaldamise kuupäev, väljaanne, kirjastaja, EAN-13 ehk triipkood, lehtede arv ning autor(id) (Lisa 2).

Inventuuri vaates kutsutakse välja meetod *getAll* kus antakse kaasa raamatu tabeli nimi (joonis 18).

```
export class BookService {  
  
  constructor(private service: IndexeddbService) {  
  }  
  getAllBooks() {  
    return this.service.myIdb.getAll("book");  
  };  
}
```

Joonis 18. Meetod raamatute saamiseks andmebaasist

IndexedDB liidestamisel tekkisid probleeme kasutusele võetud teegiga, kuna see oli spetsiaalselt kirjutatud selle projekti jaoks ning probleemid tekkisid selle vähesese testimise tõttu. Lõputöö autoril tuli probleeme andmebaasi loomisel kui tuli välja, et ilma primaarse võtmega ei saa andmebaasi koostada mis polnud testimise juures antud ajahetkel oluline. Testimisega tuli veel välja, et indexedDB ei toeta bigint tüüpi ning see tuli asendada number'i tüübiga. Teegi probleemide lahendamiseks võeti ühendust teegi autoriga ning ühekoos said probleemid lahendatud.

3.2.6 Triipkoodi lugeja

Paljudel raamatutel mis väljastatakse tuleb tagaküljel kaasa ean13 standardile vastav triipkood. Triipkoodilugeja teeb raamatu otsimise ja lisamise inventuuri palju kergemaks.

Pildilt triipkoodi tuvastamine on implementeeritud Rustiga, mis on kompileerub WebAssemblisse. WebAssemblisse kompileerimine on tehtud wasm-pack tööriista abil. Wasm-pack aitab muutujaid kopeerida Webassembli ja JavaScripti vahel, tehes valmis vajaliku koodi JavaScripti ja WebAssembli poolel.

Triipkoodi lugeja liidestamist alustati uue komponendi loomisega kus on paika pandud HTML elemendid video ja *canvas*. Video elemendile on lisatud *autoplay* atribuut, mis paneb kohe video tööle kui ta on selleks loa saanud. Joonisel 19 on näha *askStream* funktsioon mis küsib kasutajalt luba kaamera kasutamiseks, kaasa andes piirangud. Piirangu- teks on audio ja video objektid, antud töös antakse video objektile kaasa kaamera resolu- tsioon ja millist seadme kaamerat eelistatult käivitada.

```

async askStream(): Promise<any> {
  if (this.stream && this.stream.active) {
    return;
  }
  this.stream = await navigator.mediaDevices.getUserMedia(this.constraints);
  this.video.nativeElement.srcObject = this.stream;
}

```

Joonis 19. Kasutajalt video pildi luba küsimine

Enne kasutajalt loa saamist on JavaScript teinud valmis video ja *canvas* elemendid ja *canvas* elemendi ID saadetakse Rustile. Peale loa saamist hakkab JavaScript videoelement pilti näitama ja kaadrilt pilte kopeerima *canvas* 'ile. Wasm-pack küsib *canvas*'elt pildi ID kaudu ja saadab edasi wasm'ile tuvastamiseks. Wasm hakkab pildilt tuvastama triipkoodi, kui kood tuvastatud siis arvutatakse kontrollnumber, ning kui EAN13 kood on kehtiv, tagastakse number JavaScriptile (Lisa 3).

Joonisel 20 on näha triipkoodi lugeja välja kutsumist, mis toimub nupule vajutusega. Toimub kõik mis eelnevalt kirjeldatud sai, ning tuvastamisel korral tehakse API GET päring, mis tagastab raamatu andmed, ning raamatu andmed kuvatakse ekraanil ja lõpetatakse video.

```

onScanStart() {
  this.videoCapture.getBarcode().then((ean13) => {
    this.bookSearchService.searchBook(ean13).pipe()
      .subscribe(
        res => {
          this.book = res[0];
        });
    this.videoCapture.stopCamera();
  });
}

```

Joonis 20. Triipkoodi lugeja välja kutsumine raamatu otsimise vaates

4 Tulemused

Antud töö käigus loodi tulevasele rakendusele disain ja prototüüp. Arvestades ajalist piirangut võeti projekti skoopiks ehitada raamatu otsimise, salvestamise ja kuvamise prototüüp. Lõputöö käigus saadi uusi teadmisi ja kogemusi uute tehnoloogiate näol. Eesrakenduse ehitamisel esines probleeme, aga sai edukalt probleemidele lahendus leitud. Töö käigus sai analüüsitud erinevaid tehnoloogiaid ja valitud kõige sobilikumad. Valmis sai inventuuri mobiili vaade mis on liidestatud IndexedDB-ga ja triipkoodi lugeja ning käsitsi raamatu otsimise võimalus.

4.1 Edasiarendus

Antud tööd on võimalik edasi arendada paljude võimalustega. Alustades eesrakenduse sisselogimise ja registreerimis vaadete loomisega ja sidumine tagarakendusega. Suurimaks edasi arendamiseks oleks laenutamise mooduli ehitamine. Arvestades, et rakendus on mõeldud erialakirjanduse jaoks siis võib kohata raamatuid millel pole triipkoodi, et hiljem oleks kergem skaneerida siis pakkuda võimalus genereerida triipkood ja see raamatu peale printituna teipida. Triipkoodi lugemise prototüübi juurde võiks lisada funktsionaalsuse, mis on võimeline tuvastada teisi triipkoodi vorme, mis ei ole EAN13 standard. Eesti keelsete teoste suurendamiseks serveris on võimalik kirjutada automaatne kood mis võtab raamatupoodidest raamatute info ja lisab selle serveri.

5 Kokkuvõte

Antud diplomitöö eesmärk oli luua väike ja hõlpsasti kasutatav raamatukogu tarkvara firmadele ja erakasutamiseks, mis aitab luua ülevaade olemasolevatest erialakirjandustest mille kaudu oleks võimalik neid laenutada. Lahenduse esialgne skoop on raamatute lisamine süsteemi läbi triipkoodi lisaks käsitsi otsimine ja õige raamatu lisamine nimekirja.

Lõputöö käigus tehti analüüs funktsionaalsetele ja mittefunktsionaalsetele nõuetele, olemasolevate lahendustele ning analüüsiti erinevaid tehnoloogiaid. Töökäigus alustati uue Angulari rakendusega, kus sai valmis raamatu lisamine läbi triipkoodi ja läbi käsitsi otsingu. Soovitud raamatu leidmisel on võimalik see lisada inventuuri.

Tulevikuplaanid veebirakendusele oleks teiste vaadete ehitamine ja tagarakendusega sidumine. Sinna juurde kuulub triipkoodi lugejale teiste triipkoodi vormide ära tundmine ja raamatu laenutamise moodul.

Lõputööd võib lugeda õnnestunuks, kuna rakendust alustati disainimisest, mis tegi selgemaks kui oluline disain on projekti juures ning saadi uusi teadmisi uute tehnoloogiate kasutamisest.

Kasutatud kirjandus

- [1] „Ester,“ [Võrgumaterjal]. Available: www.ester.ee. [Kasutatud 10 2 2021].
- [2] „Campus book rentals,“ [Võrgumaterjal]. Available: <https://www.campusbookrentals.com/>. [Kasutatud 10 2 2021].
- [3] „Libib,“ [Võrgumaterjal]. Available: www.libib.com. [Kasutatud 10 2 2021].
- [4] „What Is a Technology Stack? Choosing the Right Tech Stack For Your Web Project,“ [Võrgumaterjal]. Available: <https://medium.com/swlh/what-is-a-technology-stack-choosing-the-right-tech-stack-for-your-web-project-3f295cf60f10>. [Kasutatud 15 2 2021].
- [5] „What is REST,“ [Võrgumaterjal]. Available: <https://restfulapi.net/>. [Kasutatud 15 2 2021].
- [6] „Angular Docs,“ [Võrgumaterjal]. Available: <https://angular.io/docs>. [Kasutatud 20 2 2021].
- [7] „Introduction What is Vue.js?,“ [Võrgumaterjal]. Available: <https://vuejs.org/v2/guide/index.html>. [Kasutatud 21 2 2021].
- [8] „Getting Started,“ [Võrgumaterjal]. Available: <https://reactjs.org/docs/getting-started.html>. [Kasutatud 22 2 2021].
- [9] „Bootstrap Introduction,“ [Võrgumaterjal]. Available: <https://getbootstrap.com/docs/4.2/getting-started/introduction/>. [Kasutatud 22 2 2021].
- [10] „WebAssembly Concepts,“ [Võrgumaterjal]. Available: <https://developer.mozilla.org/en-US/docs/WebAssembly/Concepts>. [Kasutatud 22 2 2021].
- [11] „What is PHP?,“ [Võrgumaterjal]. Available: <https://www.php.net/manual/en/intro-whatis.php>. [Kasutatud 28 2 2021].
- [12] „What is Java technology and why do I need it?,“ [Võrgumaterjal]. Available: https://www.java.com/en/download/help/whatis_java.html. [Kasutatud 28 2 2021].
- [13] „PYPL PopularitY of Programming Language,“ [Võrgumaterjal]. Available: <https://pypl.github.io/PYPL.html>. [Kasutatud 28 2 2021].
- [14] „A tour of the C# language,“ [Võrgumaterjal]. Available: <https://docs.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/> . [Kasutatud 28 2 2021].
- [15] „What is C Programming Language? Basics, Introduction, History,“ [Võrgumaterjal]. Available: <https://www.guru99.com/c-programming-language.html> . [Kasutatud 28 2 2021].
- [16] „What is C++?,“ [Võrgumaterjal]. Available: <https://www.educative.io/edpresso/what-is-cpp>. [Kasutatud 28 2 2021].
- [17] „Rust by Example,“ [Võrgumaterjal]. Available: <https://doc.rust-lang.org/stable/rust-by-example/>. [Kasutatud 28 2 2021].
- [18] „Most Loved, Dreaded, and Wanted Languages,“ [Võrgumaterjal]. Available: <https://insights.stackoverflow.com/survey/2020#technology-most-loved-dreaded-and-wanted-languages-loved>. [Kasutatud 28 2 2021].
- [19] „The Types of Modern Databases,“ [Võrgumaterjal]. Available: <https://www.alooma.com/blog/types-of-modern-databases>. [Kasutatud 3 3 2021].
- [20] „Sqlite advantages and disadvantages,“ [Võrgumaterjal]. Available: <https://www.javatpoint.com/sqlite-advantages-and-disadvantages>. [Kasutatud 3 3 2021].

- [21] „What is MySQL?“, [Võrgumaterjal]. Available: <https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html>. [Kasutatud 3 3 2021].
- [22] „What is MariaDB? How Does MariaDB Work?“, [Võrgumaterjal]. Available: <https://www.tecmint.com/what-is-mariadb-how-does-mariadb-work/>. [Kasutatud 5 3 2021].
- [23] „What is PostgreSQL? Introduction, Advantages & Disadvantages“, [Võrgumaterjal]. Available: <https://www.guru99.com/introduction-postgresql.html>. [Kasutatud 6 3 2021].
- [24] „What is MongoDB? Introduction, Architecture, Features & Example“, [Võrgumaterjal]. Available: <https://www.guru99.com/what-is-mongodb.html>. [Kasutatud 6 3 2021].
- [25] „IndexedDB API“, [Võrgumaterjal]. Available: https://developer.mozilla.org/en-US/docs/Web/API/IndexedDB_API. [Kasutatud 6 3 2021].
- [26] „14 best prototyping tools for UI/UX designers“, [Võrgumaterjal]. Available: <https://webflow.com/blog/prototyping-tools>. [Kasutatud 10 3 2021].
- [27] „Figma Pricing“, [Võrgumaterjal]. Available: <https://www.figma.com/pricing/>. [Kasutatud 10 3 2021].
- [28] „What Is Adobe XD? a 101 Intro“, [Võrgumaterjal]. Available: <https://designshack.net/articles/software/what-is-adobe-xd/>. [Kasutatud 12 3 2021].
- [29] „Webflow pricing“, [Võrgumaterjal]. Available: <https://webflow.com/pricing>. [Kasutatud 13 3 2021].
- [30] „What is Axure RP? Is it right for you and worth buying?“, [Võrgumaterjal]. Available: <https://uxplanet.org/what-is-axure-rp-is-it-right-for-you-and-worth-buying-e4acb77a7992>. [Kasutatud 13 3 2021].
- [31] „Kellele on vaja prototüüpi?“, [Võrgumaterjal]. Available: https://blog.twn.ee/et/kellele_on_vaja_prototuuupi. [Kasutatud 20 3 2021].
- [32] „CLI Overview and Command Reference“, [Võrgumaterjal]. Available: <https://angular.io/cli>. [Kasutatud 20 3 2021].
- [33] „About Node.js“, [Võrgumaterjal]. Available: <https://nodejs.org/en/about/>. [Kasutatud 21 3 2021].
- [34] „Stateful and stateless components, the missing manual“, [Võrgumaterjal]. Available: <https://ultimatecourses.com/blog/stateful-stateless-components>. [Kasutatud 3 4 2021].
- [35] „Introduction“, [Võrgumaterjal]. Available: <https://getbootstrap.com/docs/5.0/getting-started/introduction/>. [Kasutatud 4 4 2021].
- [36] „BEM 101“, [Võrgumaterjal]. Available: <https://css-tricks.com/bem-101/>. [Kasutatud 9 4 2021].
- [37] „What is Bootstrap: A Beginner's Guide“, [Võrgumaterjal]. Available: <https://careerfoundry.com/en/blog/web-development/what-is-bootstrap-a-beginners-guide/>. [Kasutatud 15 4 2021].

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks¹

Mina, Märten Vuks

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose " Ettevõtte- ja eraraamatukogusid haldav veebirakendus" , mille juhendaja on Meelis Antio, Mait Sarv.
 - 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.

29.04.2021

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

Lisa 2 – IndexedDB raamatu tabeli kirjeldus

```
import { IndexedDbStores } from '../indexedDB/indexed-db-stores';
import { Author } from './author'

export class Book {
  book_id: number;
  title: string;
  pubDate: Date;
  edition: string;
  publisher: string;
  ean13: number;
  pages: number;
  authors: Author[];
}

/* See on ühe tabeli definitsioon,
mis lisatakse andmebaasi defineerivate tabelite jadasse */

IndexedDbStores.storeList.push({
  name: 'book',
  primaryKey: 'book_id',
  indexes: [
    { name: 'ean13', keypath: 'ean13', options: { unique: true } }
  ],
  type: Book,
  version: 1,
  conversions: null
});
```

Lisa 3 – JavaScripti ja WebAssembly suhtlus

