

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Kent-Marcus Pluutus 185042IADB

Kasutajapõhise arveldusrakenduse prototüübi loomine väikeettevõtjale

Bakalaureusetöö

Juhendaja: Kristiina Hakk
PhD

Tallinn 2023

Autorideklaratsioon

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

Autor: Kent-Marcus Pluutus

15.05.2023

Annotatsioon

Arveldamine ning maksetehingute haldamine on igale ettevõttele oluline äriiline osa. Väikeettevõtjana on tihtipeale keeruline leida platvormi, mis ei nõuaks maksekohustusi ning samaaegselt võimaldaks mitmekülgseid funktsionaalsuseid ja korrektset arvete koostamist.

Antud töö eesmärk on luua kasutajapõhise arveldusrakenduse prototüüp arvete, kontaktide ning majandustehingute haldamiseks. Rakendus on peamiselt suunatud väikeettevõtjatele. Arendusprotsessi tulemuseks on luua ühtse kasutajaliidese stiiliga veebirakendus, mis koondab suure osa arveldusega seonduvast informatsioonist kokku ühele platvormile ilma, et kasutaja peaks tasuma igakuiseid makseid. Võimaldatakse PDF-vormingus arvete koostamist ja allalaadimist, kontaktiraamatu haldamist, maksutehingute sisestamist ning nende ülevaate kuvamist.

Töös tutvustatakse lahendatavat probleemi ning püstitatud eesmärki, teostatakse turuanalüüs ja seatakse paika rakenduse nõuded. Samuti analüüsitakse erinevaid arendustehnoloogiaid, et sobiv valik töö arendamiseks langetada. Järgnevalt kirjeldatakse nii esi-, kui ka tagarakenduse arhitektuurilist struktuuri ja üldist arendamise käiku. Viimane peatükk sisaldab töö järeldusi ning autori poolt väljatoodud edasiarenduste võimalusi.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 34 leheküljel, kuut peatükki, 19 joonist, 4 tabelit.

Abstract

Creating a User-Based Invoicing Application Prototype for Small Businesses

Invoicing and managing payment transactions is an important business part of every company. As a small business, it could often be difficult to find a platform that does not require payment obligations, and at the same time allows versatile functionalities and valid invoice creation.

The aim of this thesis is to develop a prototype of a user-based invoicing application for managing invoices, contacts, and business transactions in forms of sales- and purchase invoices. The application is mainly aimed toward small businesses. The result of the development process is to create a web application with a similarly styled user interface appearance throughout the webpages, which brings together a large part of the billing-related information, without any monthly payment obligations. It is possible to create and download invoices in PDF-format, manage the contact book view, enter business transactions, and manage them.

The first content chapter of the work introduces the problem and the aim. Following this, a market analysis is performed, and the application requirements are set. As well as multiple different development technologies are analyzed, to make a suitable choice for the process.

The development chapter describes the architectural structure of both the front-end and back-end applications and the overall development process. The last chapter contains the conclusions of the completed application development, and the author brings out further development possibilities.

The thesis is in Estonian and contains 34 pages of text, six chapters, 19 figures, 4 tables.

Lühendite ja mõistete sõnastik

API	<i>Application Program Interface</i> - programmiliides.
CSS	<i>Cascading Style Sheets</i> - veebirakenduste küljendamise märgistuskeel.
CRUD	<i>Create-Read-Update-Delete</i> - põhioperatsioonid veebisuhtluses.
CSRF	<i>Cross-Site request forgery</i> – veebikeskkonna autentimist pettev rünnakutüüp.
DTO	<i>Data Transfer Object</i> - objekt andmete edastamiseks.
HTML	<i>HyperText Markup Language</i> - veebilehtede märgendamiskeel.
JWT	<i>Json WebToken</i> – laialdselt kasutatav autentimistunnus.
MVC	<i>Model-View-Controller</i> – arhitektuuriline rakenduse ülesehitus.
PDF	<i>Portable Document Format</i> – elektrooniliste dokumentide vorming.
REST	<i>Representational State Transfer</i> – arhitektuuriline stiil veebirakendustevahelisel suhtlusel.
SOAP	<i>Simple Object Access Protocol</i> – arvutivõrkude protokoll, et veebiteenuste andmeid vahetada.
SQL	<i>Structured Query Language</i> – struktuuripäringukeel.
URL	<i>Uniform Resource Locator</i> – internetiaadress.
XSS	<i>Cross-Site scripting</i> – arvutiturbe raames kasutatav rünnakutüüp.

Sisukord

1 Sissejuhatus	10
2 Probleemi taust ja rakenduse eesmärk.....	11
2.1 Probleem.....	11
2.2 Eesmärk	12
3 Analüüs.....	13
3.1 Alternatiivsed platvormid.....	13
3.2 Turuanalüüsi järeldused.....	16
3.3 Lähtetingimused ja nõuded.....	17
3.3.1 Eeltingimused	18
3.3.2 Arveldusrakenduse prototüübi funktsionaalsed nõuded.....	18
3.3.3 Arveldusrakenduse prototüübi mittefunktsionaalsed nõuded	19
3.4 Projekti ettevalmistused.....	20
3.5 Tagarakendus.....	20
3.5.1 Tagarakenduse programmeerimiskeele valik	21
3.6 Eesrakendus	24
3.6.1 Eesrakenduse programmeerimiskeele valik	24
3.6.2 Esirakenduse raamistiku valik.....	25
3.7 Rakenduse andmebaas	25
3.7.1 Andmebaasitehnoloogia	25
4 Lahenduse loomine.....	26
4.1 Eeltöö planeering	26
4.2 Andmebaas	26
4.3 Tagarakenduse seadistamine	27
4.3.1 Tagarakenduse ja andmebaasi ühildamine	27
4.4 Tagarakenduse ülesehitus	28
4.4.1 Rakendusliidese lõpp-punktid	30
4.4.2 Turvameetmed	31
4.4.3 Arvelehtede loomine	31
4.5 Eesrakendus	34
4.5.1 Eesrakenduse ülesehitus	34
4.5.2 Kasutajaliides	35

5 Töö tulemused ja edasiarendused	40
5.1 Töö tulemused	40
5.2 Edasiarendusvõimalused	41
6 Kokkuvõte	43
Kasutatud kirjandus	44
Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks	48
Lisa 2 – Isolta arveldusrakenduse arvete koostamise eelvaade	49
Lisa 3 – OnArvel veebikeskkonna poolt loodud arve näidis	50
Lisa 4 – Merit Aktiva arvete koostamise vaade	51
Lisa 5 - H2 andmebaasimootori veebikonsooli vaated	52
Lisa 6 – prototüübi andmebaasimudeli ERD skeem	53
Lisa 7 – Kasutajaloo näide.....	54
Lisa 8 – Prototüübi arvete loomise eesrakenduse vaade	55
Lisa 9 - Tagarakenduses loodud arve näidis.....	56
Lisa 10. Prototüübi koodi repositooriumi lingid	57

Jooniste loetelu

Joonis 1. Näide kasutajalugude kaardistamisest Asana keskkonnas.	26
Joonis 2. Spring Initializr seadistamine.	27
Joonis 3. H2 serveri konfiguratsioon Spring Boot projektis.	28
Joonis 4. DAL ja ORM kihtide vahelise suhtluse näide customerId põhiste andmete pärimiseks.	29
Joonis 5. Projekti ülesehitus failide tasandil.	30
Joonis 6. @GetMapping()-uga tähistatud päring, CompanyContactInfo tabelist andmete kuvamiseks.	31
Joonis 7. iTextPDF abil PDF dokumendi loomine.	32
Joonis 8. getInvoiceBodyFields meetodi lühendatud koodinäidis.	33
Joonis 9. getInvoiceBodyFields meetodi poolt loodud arveridade tabeli päise näide.	33
Joonis 10. Eesrakenduse ülesehitus failide tasandil.	34
Joonis 11. Platvormi registreerimisvaade.	35
Joonis 12. Registreerimisvaatele järgnev kontaktandmete vormi vaade.	36
Joonis 13. Rakenduse kontaktiraamatu vaade.	36
Joonis 14. Modaalvaade kontaktide lisamiseks.	37
Joonis 15. Prototüübi arvete arhiivivaade.	37
Joonis 16. Prototüüprakenduse arve saaja rippmenüü ning valikujärgne andmeväljade täitumise vaade.	38
Joonis 17. Prototüübi sissekannete vaade.	38
Joonis 18. Ostuarve lisamise modaalvaade.	39
Joonis 19. Rakenduse andmesuhtluse skeem.	41

Tabelite loetelu

Tabel 1. Isolta keskkonna kuutasude maksumus.....	14
Tabel 2. Merit Aktiva teenuste versioonianalüüs.....	15
Tabel 3. Prototüübi SWOT-analüüs.....	17
Tabel 4. Programmeerimiskeelte võrdlus.....	23

1 Sissejuhatus

Arveldamine ning maksetehingute jälgimine on kaupsid või teenuseid pakkuvale ettevõttele oluline äriiline osa. See hõlmab arvete koostamist ja edastamist klientidele, et märku anda äritehingust ning seda kaardistada. Kuigi arveldussüsteeme on erinevate funktsionaalsustega - alates manuaalsetest platvormidest kuni keerukate veebirakendusteni, võib Eestis paikneva väikeettevõtjana taoliste keskkondade kasutamine raskendatud olla. Sageli kaasnevad selliste süsteemidega maksekohustused või puudujäägid, nagu näiteks vale arveformaad, ebaselge kasutajaliides, või vähesed funktsionaalsused. Seetõttu ei pruugi taoliste keskkondade kasutamine end õigustada ja ettevõtjal tuleb arveid või maksetehinguid käsitsi koostada ning hallata.

Käesoleva lõputöö eesmärk on luua väikeettevõtjatele suunatud lihtsasti kasutatav arveldusplatvormi kasutajaliidese prototüüp, mille põhifunktsionaalsusteks on *PDF*-vormingus arvete koostamine, teiste ettevõtete kontaktide lisamine ning võimalus maksetehinguid hallata. Tulemusena on soov pakkuda mugavat maksevaba platvormi.

Lõputöö on jaotatud kuueks peatükiks. Töö teises peatükis annab autor ülevaate probleemi taustast ning sellega kaasnevast eesmärgist, tuginedes ühtlasi ka enda eelnevale väikeettevõtte haldamise kogemusele. Kolmandas peatükis antakse ülevaade turul eksisteerivatest peamistest alternatiivrakendustest, tuues välja nende eelised ja puudused. Koostatakse kasutajalood ja nõuded, millele prototüübi loomise käigus tuginetakse. Samuti põhjendatakse programmeerimiskeelte ning raamistike valimisel langetatud otsused. Neljandas peatükis kirjeldatakse arveldusrakenduse prototüübi arendusfaase ning näidatakse selle käigus loodud kasutajaliidest ja arendatud funktsionaalsuste tulemit. Selgitatakse täpsemalt lahti ees- ja tagarakenduse käik ning andmebaasi ülesehitus. Viiendas peatükis tutvustatakse töö tulemusi ning edasiarenduse võimalusi.

2 Probleemi taust ja rakenduse eesmärk

Ettevõtte jaoks on arvete esitamine klientidele osutatud toodete või teenuste eest äärmiselt tähtis protsess. Just seeläbi tagatakse kaupade ja tehtud vaeva eest tasu ning kindlustatakse endale suurem rahaline jätkusuutlikkus. Täpne ja korrektselt vormistatud õigeaegne arveldamine loob klientides kindlustunde ning tagatakse edasiseks tegutsemiseks ning kasvamiseks vajalike ressursside olemasolu. Selge kirjapildi ning hästi struktureeritud arve abil saab ettevõtja väljendada, et ta väärtustab oma kliente ja võtab arveldusprotsesse tõsiselt. See omakorda aitab luua positiivset mainet ning võib kaasata endaga korduvat äritegevust. Ühtlasi kehtib arve ka juriidilise dokumendina, mis sümboliseerib ettevõtte ja tema klientide vahelist äritehingut ning sellega kaasnevaid tingimusi. [1] Välise arveldussüsteemi kasutamine aitab ettevõttel hallata finantsdokumente ja raamtupidamistoiminguid täpsemalt ja mugavamalt, koondades suure hulga andmetest ühte keskkonda ning võimaldades ettevõtjal maksekuulekust käepärasemalt tagada.

Lõputöö teema tuleneb autori enda kogemustest, kes on väikeettevõtjana kasutanud mitmeid turul pakutavaid arvelduskeskkondasid või -platvorme, kuid piirangute või puudujääkide tõttu ei ole äratasuvat ning sobivat keskkonda leidnud.

2.1 Probleem

Autor leiab, et Eestis tegutseva väikeettevõtjana, kes väljastab mõnikümne arvet majandusaastas, on keeruline leida töökindlat, targalt ülesehitatud ning äratasuvat arveldusrakendust, mis omaks head arvete koostamis- ja arhiveerimissüsteemi ning võimaldaks ühtlasi mugava kasutajaliidese abil majandustehinguid hallata.

Kuigi turul on sarnaseid teenused pakkuvaid arveldusrakendusi olemas, omavad mitmed nendest väga kitsast skoopi, keskendutakse peamiselt fikseeritud kujul arvete loomisele ja puudub tehingute haldamise võimalus. Mõned platvormid pakuvad algelisel kujul erinevaid majandustehingute märkimisvõimalusi, kuid kasutavad automatiseeritud või vigaseid tõlkesüsteeme, kuna tegemist on välismaa tootjate poolt loodud keskkondadega. Seetõttu võivad ka arveformaadid eksida Eestis seatud arvestandardite vastu. Sageli on enimlevinud eestikeelsed platvormid suunatud suurematele ettevõtetele ning nõuavad peale 14-päevast prooviperioodi mitme-kuulist, kuni üheaastast liitumislepingu

sõlmimist. Erinevad kuutasud võivad ulatuda kuni 49 või enama euroni kuus. Autor leiab, et sellisel kujul igakuise liitumistasu põhine arveldamine ei pruugi igal ettevõtjal ära tasuda, sest võib puududa kindel sissetulekuallikas, mis tähendab, et igakuine tasu muutub ebavajalikuks kohustuseks.

2.2 Eesmärk

Bakalaureusetöö eesmärk on luua kergesti kasutatav *REST-API* liidestuse põhine rakendus, mille sihtgrupiks on väikeettevõtjad. Prototüübi abil peab olema võimalik koostada arveid, anda nendest kronoloogiline ülevaade ning filtreerida majandustehinguid, ilma, et peaks muretsema prooviperioodide või igakuiste maksukohustuste pärast. Lisaks on eesmärk pakkuda lahendust, mis võimaldab ettevõtjale ajalist kokkuhoidu ja ühtset, lihtsasti kasutatavat kasutajaliidese mustrit, kuna suur osa arveldusega seonduvast informatsioonist on kokku koondatud ühele platvormile. Rakenduse kasutajaliidese koostamisel analüüsitakse erinevate turul olemasolevate platvormide disaini ning kasutajasõbralikkust ning pööratakse tähelepanu lihtsale välimusele.

Eduka prototüübi loomine võimaldab tulevikus edasiarenduste kujul platvormile lisada rohkem näiteks arvete koostamise ja raamatupidamisega seonduvaid funktsionaalsusi, et oleks võimalik majandusaastaaruandeid ja ettevõttega seonduvat informatsiooni ühe keskkonna raames veelgi laiemalt hallata.

3 Analüüs

Projekti loomine saab alguse turuanalüüsist, millele järgnevalt kaardistatakse projekti nõuded, arvestades nii äriliste, kui ka tehniliste aspektidega. Sammude ette planeerimine, tuginedes kogutud informatsioonile, on hea viis teostada edukas tootearendus ning maandada ebaõnnestumise riski. [2]

3.1 Alternatiivsed platvormid

Internetipõhistest arveldusrakendustest on saanud igas suuruses ettevõtete jaoks asendamatu tööriist, mis aitab arveldusprotsesse hallata ja majandustehingutega kursis olla [3]. Tänapäeval on veebis saadaval üha rohkem lõputöö käigus valmivale rakendusele analoogseid platvorme, mis omavad kõik erinevaid ainulaadseid eeliseid ja funktsionaalsusi. Paraku lisaks positiivsetele külgedele, leidub ka mitmeid puudujääke ja kitsaskohtasid. Käesoleva lõputöö lahendus proovib olemasolevate platvormide positiivsete külgede pealt inspiratsiooni koguda ja negatiivsetele külgedele lahendusi leida.

Eeltöö käigus analüüsitakse plaanitavale prototüübile sarnanevate funktsionaalsustega platvorme ning keskendutakse kindlatele kriteeriumitele, et ühtsemat mustrit hoida. Kirjeldatakse teenuse taust ning ka rakenduste kasutamistasud, kuid peamiselt jälgib autor rakenduste kasutajaliidest ja funktsionaalsusi – sealhulgas mängib olulist rolli arvete vormistus.

Isolta [4] – tõenäoliselt üks suurimaid konkurente Eestis eksisteerivate arveldustarkvarade seas, omades üle 30 000 kliendi. Tegemist on Soome päritolu rakendusega, mis võimaldab ettevõtjatel tekitada uusi arveid, luua raamatupidamisaruandeid, koostada tooteid ning lisada kontakte. Kasutajaliides on väga intuitiivne – põhifunktsionaalsused on reastatud vasakpoolsesse külgribasse kategooriate põhiselt ja alamvaated ei sisalda üleliigset informatsiooni. Arvete koostamise vaade on hästi teostatud, võimaldades andmete sisestamisel luua kliendile esialgne eelaimdus, milline lõpptulemuse paigutus välja võib näha (vt lisa 2).

Üheks Isolta eeliseks mõningate konkurentide ees on ka võimalus luua e-arveid ning müügiraportite aruandeid.

Miinuseks on see, et kuigi veebikeskkond reklaamib kuni kümnet tasuta arvet aastas, ei võimaldata ettevõtjal aasta möödudes siiski uusi arveid enam koostada. Olukorda, kus kasutaja soovib arve koostamisel sisestada tekstiväljale vabas vormis saaja andmeid, ilma, et neid kontaktiraamatus eksisteeriks, ei võimaldata – tuleb menüü kaudu uus kontakt käsitsi lisada. Ka kasutajaandmete sätestamine ei ole kõige mugavamalt lahendatud, kuna andmete muutmiseks peab pöörduma veebikeskkonna kasutajatoe poole. Lisaks, võimaldab tasuta versioon arveid ainult koostada, kuid edasine täiendamine on lukustatud tasulise versiooni taha.

Isolta toetub kuumaksepõhisele ärimudelile ning hinnad jaotuvad erinevate maksekohustuste põhisel. Kuutasude maksumused on esitatud tabelis 1.

Tabel 1. Isolta keskkonna kuutasude maksumus [4].

Lepingu miinimum kestvus	Igakuine tasu
1 aasta	7,90€ + käibemaks
3 kuud	8,90€ + käibemaks
1 kuu	10,90€ + käibemaks

OnArvel [5] - võimekas Belgia päritolu platvorm, mis reklaamib ennast kui: „Ilma kohustusteta tasuta keskkond algavatele ettevõtetele.“. OnArvel võimaldab uuel ettevõtjal liituda 14-päevase prooviperioodiga, pakkudes kuni kolme arve loomist ja haldamist. Perioodi lõppedes, et platvormi edasi kasutada, tuleb ettevõtjal siduda end põhiprogrammi 15-eurose kuumakse põhise arveldusmeetodiga.

Positiivse küljena on veebikeskkonna kasutajaliides lihtsasti kasutatav ja arusaadav ning lisaks arvete koostamisele, saab rakenduse kaudu majandustehinguid hallata. Sarnaselt Isoltale, on põhifunktsionaalsused reastatud külgribasse ning vaadete avamisel kuvatakse selged ülevaated juba olemasolevatest andmetest. Suurimateks negatiivseteks külgedeks on autori meelest platvormi üldine käitumine ning lehekülgede vaheline suhtlus ja suunamine. Kui autor endale esmakordselt kasutaja löi, et platvormiga lähemalt tutvuda, suunati ta kontaktandmete sisestamiseks edasi välisesse hollandikeelsesesse keskkonda, mis tekitas suurt segadust, sest puudus võimalus keelt muuta ning kasutajaliidese abil tagasi eelnevasse kasutajavoo sammu liikuda. Taoline olukord tekib ka igakordselt

veebikeskkonda sisse logides. Eelnevatest keeleprobleemidest tingituna, olid tekitatud arved arusaamatul põhjusel hollandikeelsed (vt Lisa 3). Samuti koostab süsteem arveridade individuaalsel täitmisel automaatselt uued arveread, isegi kui selleks ettenähtud nupule kasutaja ei vajuta.

Kuigi tegemist on eesti turul kasutatava rakendusega, ei leidnud autor arvete koostamisel võimalust valida või sisestada sobivat käibemaksuprotsendi väärtust, vaid pidi selle ettemääratud väärtuste seast valima.

Merit Aktiva [6] – tegemist on eelkõige suurettevõtete ning raamatupidajate jaoks loodud programmiga, mis pakub kasutajatele kuni 180 päevast tasuta prooviperioodi, millele järgnevalt lisanduvad maksekohustused (vt. Tabel 2).

Tabel 2. Merit Aktiva teenuste versioonianalüüs [6].

Paketid	Kuumakse	Kasutajate limiit
Standard	19€ + käibemaks	1
Pro	39€ + käibemaks	10+
Premium	49€ + käibemaks	10+

Aktiva on äärmiselt võimekas raamatupidamisprogramm, mis võimaldab koostada ostu-, müügiarveid, aruandeid ning hallata makseid, raamatupidamist ja üldist ettevõtte finantsinformatsiooni. Tugevusena, on Aktiva arvete koostamise süsteem üsnagi detailne ja võimekas – kontaktiraamatupõhine andmete sisestamine, võimalus määrata erinevaid artikleid, koostada e- ning kreditarveid ja valida sobivat valuutat. Ostu- ja müügiarvete koostamise süsteem on eraldatud vaadetena, kuigi ülesehitus on mõlemal puhul analoogne (vt. Lisa 4).

Miinusküljena leiab autor, et kuigi tasuta versiooni raames võimaldatakse märgatavas koguses funktsionaalsusi, siis katseaja lõppedes, sõlmides „Standard“ paketi, kaotab ettevõtja ligipääsu pea pooltele nendest funktsionaalsustest. Seetõttu võib esineda olukordasid, kus sõlmitakse kuumakse põhised lepingud, ilma kursis olemata, et kaotatakse ligipääs vajalikele tegevustele. Samuti on Aktiva kasutajaliides kohati segane

ja ülekoormatud – nt teadetetahvliit kuvatakse pealehe all parempoolses nurgas ning selle vaatamiseks tuleb kerida allapoole. Funktsionaalsuste rohkust saaks paremini kategoriseerida. Menüüriba kujundus erineb olulisel määral eelpool vaadeldud platvormidest ning kasutusmugavuse poolest jääb paraku nendele alla. Viimaseks, leiab autor, et laialdaste funktsionaalsuste puhul on puuduliku kogemusega keeruline esmapilgul hoomata tööriistade käitumist ning platvorm nõuab pikemat süvenemist.

Banqup [7] – *arved.ee* keskkonna edasiarendus, mis pakub e-arvete põhist arveldusteenust rahvusvahelisel tasandil. Selle veebikeskkonna analüüsi autor detailsemalt ei teosta, sest juba kasutajakonto loomisel esinevad arusaamatused. Nimelt ei leia autor võimalust eestipõhist kasutajat luua. Keskkond nõuab registreerimisel mitmekihilist autentimist, mis ei tundu töötavat korrektselt ja selleks vajaliku protsessi läbimiseks puudub juhend, mistõttu jätkata ei ole võimalik.

Lisaks veebipõhiste platvormidele, eksisteerib ka tabelarvutussüsteeme, nagu näiteks Microsoft Excel, mis võimaldavad majandustehinguid hallata. Microsoft Excel on võimas tööriist, mis pakub laial hulgal funktsionaalsusi, mis aitavad kasutajatel andmeid erinevatel viisidel hallata, analüüsida ja visualiseerida. Üheks peamiseks tugevuseks on selle võime tekitada detailseid arvutustabeleid, mida saab kohandada valemite ja vormingutega, et luua dünaamilisi interaktiivseid tabeleid [8]. Kuigi Exceli mitmekülgus, paindlikkus ja lai funktsioonide valik muudab selle võimsaks tööriistaks nii ettevõtetele, kui ka üksikisikutele, siis eelneva kogemuse puudumisel nõuab täispotentsiaali kasutamine rohkelt aega ja oskuseid. Arvete koostamine Excelis hõlmab tavaliselt andmete käsitsi sisestamist eelvormindatud arvutustabelisse, mis võib olla tüütu ning ohualdis, kuna tegemist ei ole spetsiaalselt arveldusülesannete jaoks mõeldud tarkvaraga. Lisaks puudub Excelis kasutajaliides ning tuleb kasutada erinevaid tabelilehtede vaateid. [9]

3.2 Turuanalüüsi järeldused

Eelnevas peatükis teostatud turuanalüüsi põhjal on kogutud väärtuslikke teadmisi alternatiivsete platvormide tugevuste ja nõrkuste kohta. Nende andmete põhjal luuakse prototüübi kohta *SWOT*-analüüs (ingl. k - strengths, weaknesses, opportunities, threats) tabel (vt. Tabel 3). *SWOT*-analüüs on viis, kuidas kuvada struktureeritud ülevaade kogutud sisendist, mis aitab hinnata rakenduste edukust [10]. Hinnates konkurentide

tugevaid ning nõrkasid külgesid, kaardistatakse enda jaoks potentsiaalsed võimalused, mida loodavasse prototüüpi kaasata. Tabelis antakse selge ja kokkuvõtlik ülevaade, mis tuleb hilisemas faasis rakenduse strateegilisel ülesehitusel kasuks.

Tabel 3. Prototüübi SWOT-analüüs.

TUGEVUSED +	NÕRKUSED –
<ul style="list-style-type: none"> • tasuta platvorm • skaleeritavus – edasiarendused • autor ise on teadlik rakenduse turvalisusest ja kasutajaandmete hoiustamisest • piiramatu arvete loomise võimalus • reklaamid puuduvad • luua enda kliendibaas 	<ul style="list-style-type: none"> • oskustest tingitud piiratud funktsionaalsused • piiratud kasutajabaas - väikeettevõtjad • ressursipuudus aktiivseks haldamiseks ja uuendamiseks • ajapuudusest tingitud defektid • esmasesse skooopi ei plaanita e-arvete loomist • võivad lisanduda serverite ülalpidamistasud
VÕIMALUSED +	OHUD –
<ul style="list-style-type: none"> • luua hästi teostatud kasutajaliides • vabastada kasutaja maksekohustusest • kasutajatel lubatud omavoliline isiklike andmete muutmine ja uuendamine • mitmekesine käibemaksu väärtuse sisestamise võimalus • sujuv lehekülgede vaheline suunamine ühtse kasutajaliidese välimusega • kontaktide automaatne salvestamine arvete loomisel 	<ul style="list-style-type: none"> • defektid • turvaaugud • kasutajate andmete lekkimine • seadusemuudatused

3.3 Lähtetingimused ja nõuded

Lõputöö koostamise raames tuleb autoril arvestada seadustega, mis puudutavad arveldust, et tagada korrektne viis teenuse osutamiseks. Samuti mängib rakenduse nõuete seadmisel suurt rolli eelnevalt teostatud turuanalüüs.

3.3.1 Eeltingimused

Tuginedes käibemaksuseaduse §37 lg 7 nõuetele, peavad arved olema kooskõlas riigi poolt määratud arvestandardiga [11], ehk korrektsuse mõttes peab platvorm olema võimeline arvele sisestama järgnevat informatsiooni:

- Arve järjekorranumber ja väljastamise kuupäev.
- Maksukohuslase nimi ja aadress, kauba soetaja või teenuse saaja nimi ja aadress.
- Kauba soetaja või teenuse saaja maksukohustuslasena registreerimise number, kui tal on maksukohustus kauba soetamisel või teenuse saamisel.
- Kauba või teenuse nimetus või kirjeldus ning kauba kogus või teenuse maht.
- Kauba väljastamise või teenuse osutamise kuupäev või kauba või teenuse eest osalise või täieliku makse laekumise kuupäev, kui see on kindlaksmääratav ja erinev arve väljastamise kuupäevast.
- Kauba või teenuse hind ilma käibemaksuta ning allahindlus, kui see pole hinna sisse arvatud.
- Maksustatav summa käibemaksumäärade kaupa koos kohaldatavate käibemaksumääradega või maksuvaba käibe summa.
- Tasumisele kuuluv käibemaksusumma.

Kuna arvele peab olema võimalik rakendada käibemaksu- ning allahindlusprotsente, tuleb lõppsumma arvutamise teostamisel arvestada mõlema väärtusega.

Kuigi prototüübi eesmärk on luua ettevõtjatele isiklik abivahend, mis tuleb kasuks raamatupidamisprotsesside haldamisel ja jälgimisel, ei kinnita autor, et tegemist on ametliku raamatupidamisteenust osutava veebikeskkonna või raamatupidamisregistriga. Programm pöörab tähelepanu vormistatud arvete ülesehituse korrektsusele.

3.3.2 Arveldusrakenduse prototüübi funktsionaalsed nõuded

Selle alampeatüki raames toob autor välja rakenduse planeeritavad funktsionaalsed nõuded, millele tuginetakse lõputöö praktilise osa loomisel. Nõuded on esitatud kasutajalugudena ning jaotatakse erinevatesse kategooriatesse: kliendipõhised ning üldised veebikeskkonna funktsionaalsused. Nõuded tulenevad autori nägemusest, milliseid funktsionaalsuseid rakendus kasutajale pakkuda võiks, tuginedes samas ka teiste sarnaste rakenduste analüüsile.

Kliendipõhised funktsionaalsed nõuded on järgnevad:

- Kliendina soovin ma luua endale isiklikku kontot, et saaksin oma ettevõtte andmeid sisestada.
- Kliendina soovin ma veebiplatvormile sisse ja välja logida.
- Kliendina soovin ma kasutada kontaktiraamatu vaadet, et lisada uusi ja vaadata ning muuta olemasolevaid kontakte.
- Kliendina soovin ma koostada *PDF*-formaadis ettevõttepõhiseid arveid.
- Kliendina soovin ma võimalust arvete koostamisel määrata klientide andmed automaatselt, kontaktiraamatusse eelnevalt sisestatud informatsiooni põhjal.
- Kliendina soovin ma eraldi vaates näha ja hallata eelnevalt koostatud arveid.
- Kliendina soovin ma filtreerida eelnevalt koostatud arveid kuupäevade või arve numbrite kaudu.
- Kliendina soovin ma muuta eelnevalt koostatud arvete staatuseid.
- Kliendina soovin ma käsitsi sisse kanda/dokumenteeri majandustehingute rahalisi väärtusi.
- Kliendina soovin ma näha ülevaadet eelnevalt dokumenteeritud majandustehingute summadest.
- Kliendina soovin ma filtreerida majandustehinguid kuupäevade või summade kaudu.

3.3.3 Arveldusrakenduse prototüübi mittefunktsionaalsed nõuded

Järgneva alampeatüki raames toob autor välja rakenduse mittefunktsionaalsed nõuded, milleks on:

- Veebirakendus peab olema võimeline protsesse ja päringuid täitma piisava kiirusega, kui kasutajad süsteemi kasutavad.
- Veebirakenduse kasutajaliides peab olema usaldusväärse ja ametliku välimusega.
- Veebirakendus peab olema hõlpsasti kasutatav ja intuitiivne ning ei tohi nõuda uuelt kliendilt palju süvenemist.
- Veebirakendus käitub analoogselt erinevate veebibrauserite vaadetes.
- Veebirakenduse ülesehitust peab olema selline, et seda on lihtne hooldada ja tulevikus uuendada.
- Veebirakendus on turvaline ning kaitseb kasutajaandmeid volitamata juurdepääsude eest. Sisselogimisel kasutatakse autentimisviisi.

- Veebirakendus keelab ligipääsu teatud lehtedele, kui autentimissüsteem ei tuvasta kehtivat autentimistunnust.

3.4 Projekti ettevalmistused

Tõhus planeerimine on tarkvaraarenduse oluline aspekt ning projekti haldustarkvara mängib selles võtmerolli. Enne projekti alustamist, on oluline määratleda selle ulatus, eesmärgid ja plaanitav ajakava. Edukalt planeeritud projekti abil saab arendaja oma eesmärkide nimel tõhusamalt ja tulemuslikumalt töötada. Kasutades projekti haldustarkvarasid, on võimalik jagada projekti arenduse etapid väiksemateks paremini juhitavateks ülesanneteks. Eelplaneerimine aitab arendusprotsessi vältel keskenduda ning organiseeritust hoida. [2], [12]

Tänapäeval on turul mitmeid erinevaid projekti haldustarkvarasid, nagu näiteks Trello, Asana ja Jira, mille abil on võimalik arenduse etappe ning esinevaid defekte kaardistada, tähtaegasid määrata ja jälgida, et projekti suund püsib õigel teel [13]. Autor valib enda lõputöö raames abivahendiks Asana, kuna on eelnevalt seda kasutanud ning tunneb end selles veebikeskkonnas mugavalt.

Asana on võimekas ning populaarne tööriist, mis pakub laialdasi funktsionaalsusi, et aidata kasutajatel ülesandeid tõhusalt kaardistada, määrata ja jälgida [14]. Asana mitmekülgsus võimaldab meeskondadel luua kohandatud projekte, malle ja töövooge ning see integreerub sujuvalt teiste populaarsete tööriistadega, nagu Slack ja Google Drive [15]. Asana kasutab lihtsasti kasutatavat kasutajasõbralikku liidest, mis võimaldab tarbijal vastavalt enda soovidele koostada tahvlitele ülesandeid, määrata prioriteetsust ning suunata neid meeskonnaliikmetele. Samuti toetab Asana ka populaarseid projektijuhtimise meetodikaid, nagu Kanban, Scrum ja koskmudel [14].

Kokkuvõttes on Asana mitmekülgne projekti haldustööriist, mis toetab laialdasi meetodikaid ning võimaldab kasutajal töövoogusid kohandada vastavalt enda konkreetsetele vajadustele ja eelistele. [14]

3.5 Tagarakendus

Arenduse jaoks õige tehnoloogia valimine on tähtis otsus, mis võib oluliselt mõjutada arendusprotsessi ja selle käigus loodavat tulemust. Lisaks programmeerimiskeele

valikule, tuleb autoril hinnata ka erinevaid saadaval olevaid raamistikke ja tööriistasid, kuna need võivad arendamist oluliselt lihtsustada ja koodikvaliteeti parandada. Eesmärk on leida sobiv ning hästi ühilduv kooslus raamistiku ja programmeerimiskeele vahel. [16]

3.5.1 Tagarakenduse programmeerimiskeele valik

Käesoleva lõputöö praktilise osa eesmärk on luua hajusalt ülesehitatud veebirakendus. Taolise projekti loomisel, tuleb autoril jõudluse ning funktsionaalsuse tagamiseks esmalt valida sobiv tagarakenduse tehnoloogia. Kuigi tänapäeval on saadaval arvukas koguses võimekaid programmeerimiskeeli, nagu näiteks Python, PHP, Java ja Ruby, on igal keelel omad eelised ning täpsem valik sõltub projekti spetsiifilistest vajadustest ja arendaja oskustest. Valik põhineb suuresti ka autori eelneval kogemusel, kuna täiesti uue programmeerimiskeele süntaksi õppimisele võib kuluda liigselt aega ning oodatav tulemus ei pruugi sugugi nii kvaliteetne olla. Seetõttu on fookusesse määratud C#, Java ning PHP programmeerimiskeeled ja nende raamistikud. [17], [18]

C#

C# on Microsofti poolt väljatöötatud populaarne ja võimekas programmeerimiskeel, mis on viimastel aastatel aina laialdasemalt kasutusele võetud. Selle keele üks suurimaid populaarsuse põhjuseid on kasutusmugavus ja loetavus. Lisaks toetab C# tugevalt objektorienteeritud programmeerimist, muutes selle paindlikuks keeleks keerukate rakenduste loomisel. [19]

C# on ka suurepärase keel hajutatud süsteemide arendamiseks, kuna sellel on sisseehitatud asünkroonse programmeerimise tugi, mis võimaldab arendajal kirjutada koodi, mis saab samaaegselt töötada ilma teisi protsesse blokeerimata. Hajusate süsteemide puhul on see eriti oluline väärtus, sest erinevates masinates võib töötada mitu protsessi kus on vaja reaajas üksteisega suhelda. [19]

C# programmeerimiskeelel on ka suur tugi nii *REST*-ful, kui ka *SOAP*-põhiste veebiteenustele [20]. Need funktsionaalsused koos teekide ja raamistikega muudavad selle keele ideaalseks valikuks, kui arendaja sooviks on ehitada skaleeruvaid, tõhusaid ja turvalisi veebisüsteeme [19].

C# puhul on üheks populaarseimaks raamistikuks kujunenud .NET, mis pakub arendajatele suure jõudluse juures mitmesuguseid eelehitatud teeke ja tööriistasid

veebirakenduste loomiseks. Tänu sisseehitatud raamistike paketi haldurile NuGet, on versioonihaldus ning kolmanda osapoolse teekide ja sõltuvuste lisamine üsna lihtne [21]. Lisaks sisaldab .NET mitmeid sisseehitatud turvafunktsionaalsusi, nagu nt rolli- ja autentimispõhine kasutajate süsteem [21], [22].

PHP

PHP on laialdaselt kasutatav serveripoolne skriptikeel, mida kasutatakse peamiselt veebiarendusteks. See on tuntud oma lihtsuse, tõhususe ja kasutusmugavuse poolest, mistõttu on see levinud valik veebiarenduste seas [23].

PHP sobib hästi olukordadesse, mis eeldavad dünaamilise ja interaktiivse sisu tekitamist ning andmebaasidega ühilduvust [24]. Suure populaarsuse taga on ka keele avatud lähtekoodi ligipääs, tänu millele on tänapäevaks kujunenud aktiivne arendajate kogukond, kes töötavad keele täiustamise ja laiendamise nimel [25]. Paraku on avatud lähtekoodist tingituna inimestel otsene ligipääs keele koodibaasile, mistõttu võib PHP-ga loodud veebirakenduste haavatavuste tuvastamine olla lihtsam [24]. PHP raamistike seas paistab silma Laravel oma intuitiivse süntaksi poolest. Laravel sisaldab selliseid funktsionaalsusi, nagu autentimine, veebivaheline marsruutimine ja andmebaasihaldus. Raamistikku on sisseehitatud ka Blade mallimootor, mille abiga saavad arendajad veebirakendusi luues tekitada korduvkasutatavaid malle. Tänu sellele saab hõlpsasti hoida rakenduse vaateid luues ühtset välimust. [26], [27]

JAVA

Java on mitmekülgne ja laialdaselt kasutatav programmeerimiskeel, mis töötati välja ettevõtte Sun Microsystems, Inc. poolt 1995. aastal [28]. Java positiivseteks omadusteks on selle tugev tugi objektorienteeritud programmeerimisele, mis võimaldab luua modulaarset ja lihtsasti hooldatavat koodibaasi. Lisaks pakub Java turvalisust ning võimet käsitleda suuri andmemahutusi. Kuna tegemist on platvormist sõltumatu programmeerimiskeelega, millele on loodud hulgaliselt väliseid teekide ja raamistikke, on Java ideaalne valik veebiarenduste loomiseks. Java populaarsus on osaliselt tingitud ka selle jõudlusest, sest Java virtuaalmasin (JVM) võimaldab koodi kompileerimist ja optimeerimist koodibaasi käivitamisajal. Seda funktsionaalsust saab lugeda eeliseks mõningate konkurentskeelte ees. [29], [30]

Java puhul on kõige populaarsemaks raamistikuks kujunenud Spring, mis pakub infrastruktuuri tuge Java rakenduste arendamiseks [31]. Springi tugevuseks on nt. sõltuvuste süstimine, mille abil saavad arendajad luua omavahel seotud modulaarseid rakendusi ning neid lihtsasti hallata ja testida. Samuti on võimalik Springiga siduda ka Spring MVC (*Model-View-Controller*) mustri kasutamine või Spring Boot tööriist, mis võimaldab eraldiseisvaid rakendusi automaatselt konfigureerida ja turvalisemaks teha. [32], [33] Springi JDBC mall pakub ka lihtsat ja järjepidevat viisi erinevate andmebaasidega suhtlemiseks [33].

Tagarakenduse arendamiseks sobiva programmeerimiskeele valimiseks koostab autor kriteeriumite põhise tabeli (vt. Tabel 4) eelnevalt vaadeldud keelte kohta. Kriteeriumid koosnevad programmeerimiskeelte populaarsusest, autori eelnevast kogemusest ning keelte tugevustest veebipõhiste rakenduste loomisel.

Tabel 4. Programmeerimiskeelte võrdlus [19], [24], [30], [34].

Keel	TIOBE indeks	Jõudlus	Raamistike ja kommuuni olemasolu	Autori eelnev kogemus
C#	5. koht - 7,17%	Hea	Hea	Keskmine
PHP	9. koht - 1,61%	Hea	Hea	Kasin
JAVA	3. koht - 13,56%	Hea	Hea	Hea

Esmalt hinnatakse populaarsust TIOBE indeksi põhjal. Tegemist on programmeerimiskeelte populaarsuse näitajaga, mida uuendatakse igakuiselt. Hinnangud põhinevad keelte õppimiseks mõeldud kursuste, arendajate ning kolmanda osapoole tarnijate hulgal. Reitingute arvutamiseks kasutatakse selliseid populaarseid otsingumootoreid ja veebikeskkondasid, nagu Google, Bing, Yahoo!, Amazon ja Youtube. Siinkohal tuleb arvestada, et TIOBE indeksipõhine tulemus ei tunnusta keelte paremust teiste üle, vaid annab arusaama, millises keeles on üleüldisest tulemprotsendist kõige rohkem koodiridasid kirjutatud. [34]

Nii jõudluse, kui ka raamistike ja kommuuni olemasolu võrdlus on kõikidel eelmainitud programmeerimiskeeltele võrdväärset tasemel, mistõttu on sobiva keele valik väga individuaalne, sest olenevalt rakenduse ülesehitusest ja vajadustest, tõusevad esile ka eelised teiste valikutega võrreldes.

Autori oskuste võrdlemine on suhteliselt subjektiivne kriteerium, kuid omab suurt osakaalu. Kogemuse mõttes on esikohal Java ning tugevuselt leiab autor, et Java poolt võimaldatav Spring raamistik täidab kõik vajalikud eeldused, et edukas veebirakendus luua. Seetõttu valitakse tagarakenduse programmeerimiskeeleks Java ning raamistikuks Spring.

3.6 Eesrakendus

Eesrakendus viitab rakenduse visuaalsele aspektile, millega kasutaja otseselt kokku puutub. See sisaldab tavaliselt kõiki elemente, mida ekraanile kuvatakse – nupud, menüüd, vormid, päised, tekstilahtrid ning pildid jpt elemendid. Kui tagarakendus keskendub rohkem serveripoolsele suhtlusele, siis koos eesrakendusega võimaldatakse luua komplektne tulemus. Tuleb meeles pidada, et olenemata sellest kui võimas on arendaja poolt loodud tagarakendus, ei pruugi veebikeskkond olla modernne ja kasutuskõlblik ilma hästi teostatud eesrakenduseta. [35]

3.6.1 Eesrakenduse programmeerimiskeele valik

Tänapäeval on JavaScript ja TypeScript ühed enimlevinud eesrakenduse programmeerimiskeeled, mida kasutatakse veebirakenduse arendamiseks [36].

JavaScript on objektorienteeritud skriptimiskeel, mida kasutatakse dünaamiliste ja interaktiivsete veebilehtede või -rakenduste ja isegi mängude loomiseks. JavaScript ühildub hästi CSS-i ja HTML-elementide vormindamisel [36]. Seda toetavad kõik suurimad veebibrauserid.

TypeScripti on täiustatud JavaScripti edasiarendusega, mille otstarve sarnaneb JavaScriptile, kuid omab rangemat süntaksit [37]. Erinevalt JavaScriptist, tuleb TypeScriptis muutujaid ja parameetreid selgesõnaliselt määrata ning kirjutatud kood kompileeritakse enne brauserites käivitumist, mistõttu on vigade märkamine ja koodisilumine lihtsustatud. Lisaks pakub Typescript mitmeid erinevaid mooduleid ja väliseid klasse, mis ei pruugi JavaScriptis saadaval olla. Seetõttu võib arenduste loomine olla tõhusam ja kvaliteetsem. [38]

Programmeerimiskeele valikul otsustab autor TypeScripti kasuks, tänu selle võimekusele ja struktureeritusele.

3.6.2 Esirakenduse raamistiku valik

Erinevalt programmeerimiskeele valikust, on autor suures osas kokku puutunud ainult ühe eesrakenduse raamistikuga, milleks on React.js. Seetõttu detailsemat analüüsi ei teostata, vaid otsustatakse selle valiku kasuks.

Kuigi React.js on avatud lähtekoodiga JavaScripti raamistik, töötab see analoogselt ka TypeScriptis [39]. React on kõrge jõudlusega raamistik, mida kasutatakse kasutajaliideste loomisel ning mille abil on lihtne arendada korduvkasutavaid kasutajaliideste komponente. React on äärmiselt tõhus ja skaleeritav ja kasutab virtuaalset DOM (*Document Object Model*) meetodit, mis tähendab, et Reactil võimaldatakse värsekendada eesrakenduses ainult muudetud koodjuppe, mille tulemuseks on lehevärskendused kiiremad ja tõhusamad. [40]

3.7 Rakenduse andmebaas

Andmebaas on andmete teabekogum – tavaliselt sisaldab see endas kõiki andmeid, mis on seotud ühe rakenduse või rakenduste rühmaga. Andmebaas hoiustab informatsiooni tabelite ja indeksite põhjal. Tabelitel on loogiline struktuur, mis koosneb ridadest, millel puudub kindel järjestus ning veergudest, mis on määratud tabeli loomisel autori poolt. Andmebaasi juhitakse andmebaasihaldussüsteemi (*DataBase Management System*) poolt – üldjuhul nt. mingisuguse kasutajaliidese kujul. [41]

3.7.1 Andmebaasitehnoloogia

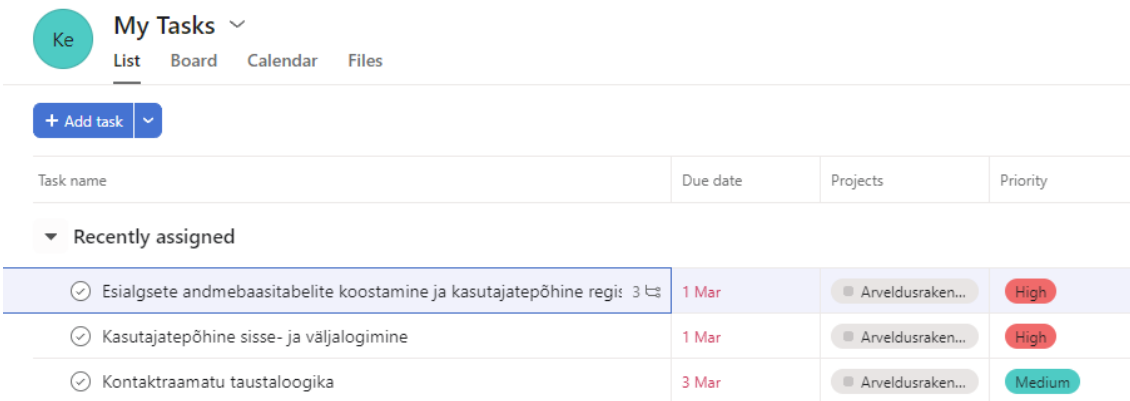
Prototüübi raames on andmebaasitehnoloogiaks valitud H2 relatsiooniandmebaasi mootor. Tegemist on lihtsasti paigaldatava avatud lähtekoodiga tööriistaga, mis on kogunud populaarsust kiiruse, väikese jalajälje ja hõlpsasti kasutatava kasutajaliidese tõttu. H2 toetab erinevaid *SQL*-i standardeid ja ühildub hästi Java rakenduste ning *JDBC* (*Java Database Connectivity*) API-dega. Samuti pakub tööriist veebilehitsejapõhist kasutajaliidest (vt. Lisa 5.) Ajutise lahendusena kasutab autor prototüübi loomisel H2 andmebaasimootorit, sest seda saab sõltuvuste süstimisega mugavalt püsti seada ning esmaste andmete koostamisel kasutada. [42], [43]

4 Lahenduse loomine

Prototüübi arendamine jaotatakse erinevateks etappideks: kasutajalugude kaardistamine Asana keskkonnas, andmebaasi ja -mudeli vormindamine ning tagarakenduse ja eesrakenduse loomine. Selles peatükis annab autor detailse ülevaate eelpool nimetatud etappidest.

4.1 Eeltöö planeering

Peatükis 3.3 tõi autor välja kasutajalood, millele loodava prototüübi raames funktsionaalsuste loomisel tugineda. Projekti realiseerimisel kaardistab autor esmalt vastavad kasutajalood, et arenduse käigus neid hõlpsamini jälgida (vt. joonis 1).



Task name	Due date	Projects	Priority
▼ Recently assigned			
✓ Esialgsete andmebaasitabelite koostamine ja kasutajatepõhine regi: 3 t	1 Mar	Arveldusraken...	High
✓ Kasutajatepõhine sisse- ja väljalogimine	1 Mar	Arveldusraken...	High
✓ Kontaktraamatu taustaloogika	3 Mar	Arveldusraken...	Medium

Joonis 1. Näide kasutajalugude kaardistamisest Asana keskkonnas.

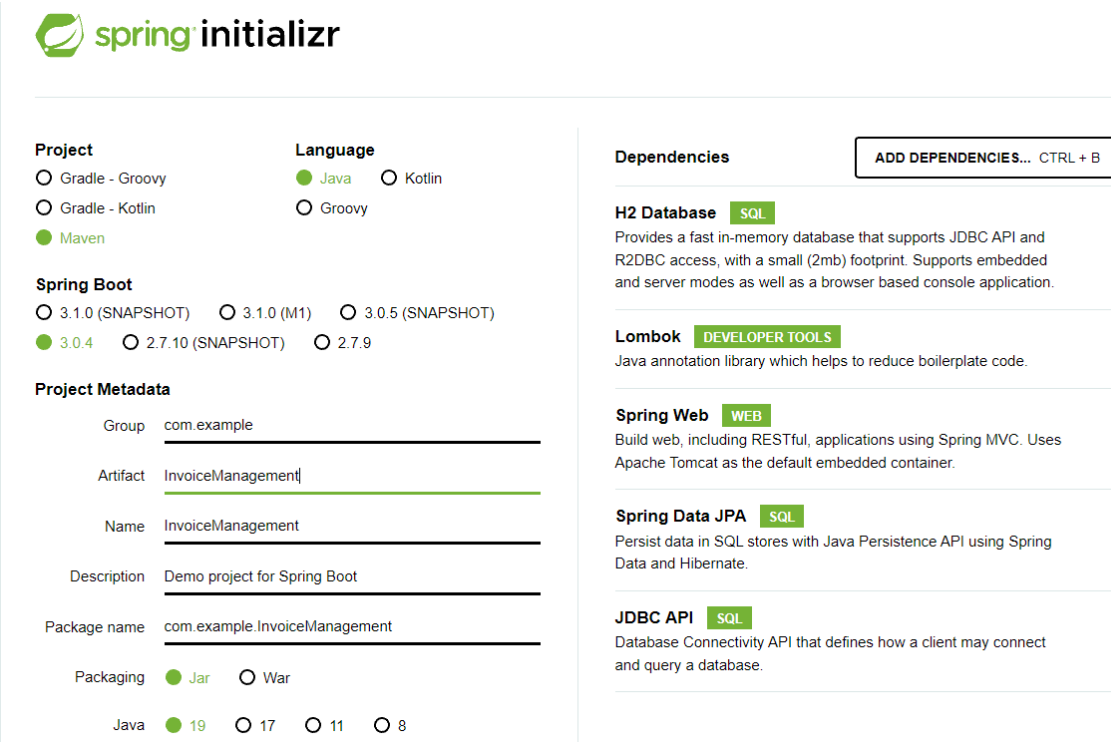
Kasutajalugude teostamisel täiendatakse jooksvalt ka nende kirjeldusi ning alamülesandeid, et arendamise käigus oleks vajalik informatsioon alati uuendatud (vt. lisa 7).

4.2 Andmebaas

Andmebaasi loomine saab alguse mudeli joonistamisest. Antud projekti raames kasutab autor Lucidchart keskkonda, et tekitada andmebaasiskeemi tabelid ja määratleda esialgsete salvestatavate andmete struktuur. Skeem koosneb olemitest, atribuutidest ja nendevahelistest seostest. Iga olem sisaldab isiksustatud primaarväärtuse võtit, mis on ainulaadne identifitseerimisvõti, et vastavas tabelis hoiustatud kirjeid märgistada [44]. Andmebaasimudeli ERD skeem on esitatud töö lisades (vt. Lisa 6).

4.3 Tagarakenduse seadistamine

Tagarakenduse seadistamise esimene etapp toimub Spring Initializr veebirakenduse abil. Spring Initializr koostab vastavalt eelvalitud seadistustele Spring Boot projekti skeleti koos esialgsete sõltuvuste ja seadistatud konfiguratsioonifailidega. See võimaldab arendajal aega kokku hoida projekti manuaalse sätestamise arvelt. Tarvis on vaid loodud projekt ZIP-failina alla laadida ning enda arenduskeskkonda importida. [45] Joonisel 2 on esitatud projekti esmane seadistus.



The screenshot shows the Spring Initializr configuration interface. It is divided into several sections:

- Project:** Radio buttons for Gradle - Groovy, Gradle - Kotlin, and Maven (selected).
- Language:** Radio buttons for Java (selected), Kotlin, and Groovy.
- Spring Boot:** Radio buttons for 3.1.0 (SNAPSHOT), 3.1.0 (M1), 3.0.5 (SNAPSHOT), 3.0.4 (selected), 2.7.10 (SNAPSHOT), and 2.7.9.
- Project Metadata:** Fields for Group (com.example), Artifact (InvoiceManagement), Name (InvoiceManagement), Description (Demo project for Spring Boot), and Package name (com.example.InvoiceManagement). Packaging is set to Jar (selected) and War.
- Dependencies:** A list of dependencies with a button "ADD DEPENDENCIES... CTRL + B". The listed dependencies are:
 - H2 Database (SQL):** Provides a fast in-memory database that supports JDBC API and R2DBC access, with a small (2mb) footprint. Supports embedded and server modes as well as a browser based console application.
 - Lombok (DEVELOPER TOOLS):** Java annotation library which helps to reduce boilerplate code.
 - Spring Web (WEB):** Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.
 - Spring Data JPA (SQL):** Persist data in SQL stores with Java Persistence API using Spring Data and Hibernate.
 - JDBC API (SQL):** Database Connectivity API that defines how a client may connect and query a database.

Joonis 2. Spring Initializr seadistamine.

4.3.1 Tagarakenduse ja andmebaasi ühildamine

Tagarakenduse ja andmebaaside vaheliseks ühenduse loomiseks koostatakse H2 andmebaasimootori konfiguratsioon Spring Boot projekti *application.properties* faili. Selle käigus täpsustatakse andmebaasi veebiaadress lokaalseks käivitamiseks, konsooli olemasolu, unikaalne baasi nimi, skeemi värskendamise strateegia, keeledialekt ning H2 konsooli kasutajanimi ja -parool. H2 serveri konfiguratsioon Spring Boot projektis on esitatud joonisel 3.

```
spring.h2.console.enabled=true
spring.datasource.generate-unique-name=false

spring.datasource.url= jdbc:h2:~/H2db/InvoiceManagementDb
spring.datasource.name=InvoiceManagementDb
spring.datasource.driver-class-name=org.h2.Driver
spring.datasource.username=sa
spring.datasource.password=password
spring.jpa.database-platform=org.hibernate.dialect.H2Dialect
spring.jpa.hibernate.ddl-auto=update
```

Joonis 3. H2 serveri konfiguratsioon Spring Boot projektis.

4.4 Tagarakenduse ülesehitus

Tagarakenduse arendamiseks on rakendatud *controller-service-repository* metoodika mustrit, ehk loogika ja arhitektuuri ülesehitus ja jaotus on mitmekihiline. Taolisel tarkvaraarhitektuuril on mitmeid tugevaid külgesid ning erinevate astmetega loogika eraldamine võimaldab rakenduse koodibaasi lihtsamini hallata ja hooldada. Seeläbi on ka tarkvara infrastruktuuri edasiste uuenduste teostamine tõhusam ning vähem ulatuslikum. Mitmekihiline arhitektuur võimaldab kasutada abstraktsiooni ehk kontseptsiooni, kus objektorienteerituse käigus koostatakse ja kuvatakse pealmistes kihtides objektide olulisi atribuute ja meetodeid, mille suurem osa ärioloogikast on varjatud alamliideste sisse. [46], [47]

Kontrollerikihis, ehk kõige pealmises kasutajapoolsemas kihis, tegeletakse esirakendusest sissetulevate REST API päringute käsitlemise ja veebipõhiste vastuste tagasisaatmisega. Kontrolleris kogutakse sisend kliendi või kasutaja poolt teostatud päringutelt, kinnitatakse sissetulevad andmed ja ligipääsu tunnusmärk ning kutsutakse välja teenuse kihis loodud meetodid. Tulemusena viiakse täide nõutud toimingud ning tulemused suunatakse esirakendusse kuvamiseks. Spring Bootis on kontrolleriklassid ära märgistatud *@RestController* annotatsioonidega. [48]

Keskmine tasand, ehk teenusekiht sisaldab peamist rakenduse ärioloogikat ja suunab informatsiooni ja toiminguid kontrolleri ja repositooriumi kihi vahelt. Siin kasutatakse andmeid, mis kontrollerikihist edasi suunatakse ning täiendava ärioloogika põhjal teostatakse vastavad valideerimised, arvutused ja kontrollid. Koos rakenduse kõige madalama, ehk repositooriumi kihiga, kutsutakse esile ka põhilised CRUD toimingud – kirjete tekitamine, lugemine, värskendamine, eemaldamine. [49] Teenusekihis on igale andmebaasiolemile loodud vastav teenuseklass (*service*), mis selle olemiga seonduvaid

tegevusi täidab. Siin hoiustatakse näiteks *iTextPdf* sõltuvuse poolt pakutavat *PDF* failiformaatide loomist ja edastamist, et sisendandmete põhjal arveformaati tekitada.

Kõige madalam kiht, ehk DAL (*data-access-layer*) võimaldab teenusekihile abstraktsiooni andmete salvestamiseks andmebaasi. Siin hoitakse meetodeid CRUD toimingute tegemiseks andmebaasiväljades. [50] Projekti raames kasutab autor selleks *JpaRepository* liidest. Samuti võimaldatakse siin ka nt. informatsiooni filtreerimist ning ligipääsu Spring Booti ORM (*object-relational mapping*) toimingute väljakutsumiseks [51]. ORM-i puhul on tegemist lihtsustatud viisiga, et andmebaasisuhtlust teostada. See tähendab, et süsteem pakub ise arendajale tööriistaid käskude esilekutsumiseks, mida muidu peaks manuaalselt täitma. ORM-i kasutamine aitab vähendada käsitsi SQL päringute kirjutamist, et andmebaasist informatsiooni pärida või sisestada – (vt. Joonis 4). [51]

```
@Repository
public interface CompanyContactInfoRepository extends
    JpaRepository<CompanyContactInfo, Long> {

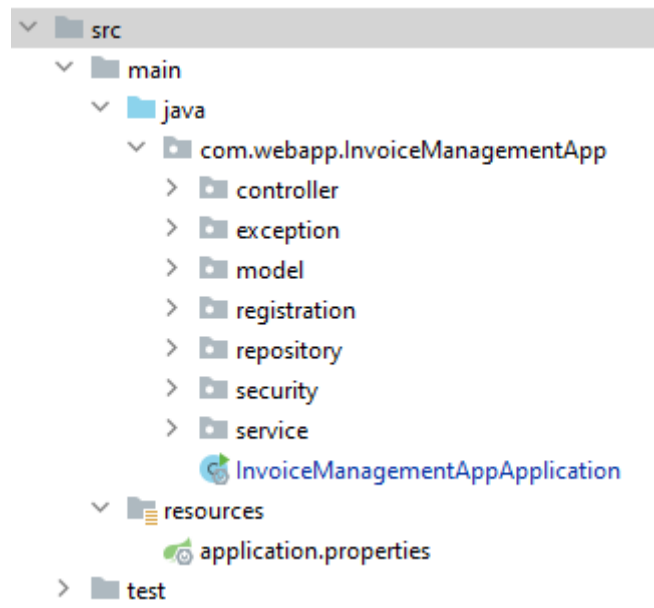
    List<CompanyContactInfo> findByCustomerCustomerId(Long
customerId);
}
```

Joonis 4. DAL ja ORM kihtide vahelise suhtluse näide *customerId* põhiste andmete pärimiseks.

Lisaks on loodud ka olemikiht (*entity layer*), mille raames luuakse klassid, mis esindavad rakenduse andmebaasimudeli põhiseid olemeid. Nende klasside siseselt määratakse atribuudid ja tüübid, mis esindavad olemitega seotud andmeid ning nendevahelisi seoseid.

Eelmainitud mitmekihilise arhitektuuri puhul rakendatakse tihti peale ka DTO (*data-transfer-object*) disainimustrit, et rakenduse skaleeritavust ja hallatavust veelgi enam täiustada, kuid antud prototüübi loomise raames otsustas autor seda mustrit mitte kasutada. Otsus tuleneb sellest, et tegemist on esmase lihtsa prototüübiga, mille arenduse ajaraam on suhteliselt kitsas. Autor leiab, et ajakulu, mis on tingitud DTO kasutamise keerukusest, ei ole tasakaalus kaasnevate positiivsete omadustega. Negatiivse küljena, võib sellises olukorras tekkida situatsioone, kus kasutajatele kuvatakse eesrakenduse kaudu üleliigseid domeeniobjektide väljasid. [52]

Tagarakenduse projekti ülesehitus arenduskeskkonnas on kujutatud joonisel 5.



Joonis 5. Projekti ülesehitus failide tasandil.

4.4.1 Rakendusliidese lõpp-punktid

Esi- ja tagarakenduste vaheline suhtlus toimub läbi rakendusliidese lõpp-punktide ehk REST API kontrollereite kaudu. Lõpp-punktide kujul on tegemist kindlatele URL-idele määratud aadressidega, mille pihta esirakenduses funktsionaalsuseid sooritades päringud teostatakse ning tulemus tagastatakse. REST-ful süsteemides vastab üldiselt iga lõpp-punkt konkreetsele HTTP-meetodile – nt. GET, POST, PUT, DELETE. [53]

Springis eristatakse päringuid annotatsioonide abil. See tähendab, et iga HTTP-meetodi puhul määratakse kontrolleri klassi siseselt ära konkreetne tüüp - *@GetMapping()*, *@PostMapping()*, *@PutMapping()* või *@DeleteMapping()*, mille kaudu täpsustatakse URL-i suunamist. Baasaadressina kasutatakse klassi enda tasandil *@RequestMapping(„/<veebiaadress>“)* annotatsiooni. Tulemusena luuakse nimeliselt algeline arusaam lõpp-punktide eesmärgi ja funktsionaalsuse kohta. [49]

Käesoleva lõputöö raames tulenevad REST API lõpp-punktide veebiaadressid doomeniobjektide nimetusest ning täpsustused ettenähtud informatsioonitöötlemise tegevusest (vt. Joonis 6).

```
@GetMapping()  
@ResponseBody  
public List<CompanyContactInfo> getAllCompanyContactInfo() {  
    return companyContactInfoService.getCompanyContactInfo();  
}
```

Joonis 6. @GetMapping()-uga tähistatud päring, CompanyContactInfo tabelist andmete kuvamiseks.

4.4.2 Turvameetmed

Turvalisus on iga rakenduse puhul kriitilise tähtsusega, et kaitsta andmeid ja ressursse volitamata juurdepääsu ja käskude eest. Kasutades Spring Bootis saadaval olevat Spring Security raamistikku, saavad arendajad kergelt integreerida turbeaspekte oma rakenduse erinevatesse kihtidesse. Autentimiseks võimaldab Spring Security erinevaid mehhanisme, nagu näiteks URL-i põhised sisselogimislehed, paroolide räsimine ja rollipõhine ligipääs. Ühtlasi saab rakendust paremini kaitsta ka XSS või CSRF rünnakute ja SQL-i süstimiste (*injection*) eest. [54]

Käesoleva lõputöö raames rakendatakse eraldi kasutajate autentimiseks ja ressursside ligipääsu võimaldamiseks ka JSON Web Tokenit (*JWT*). Tegemist on JSON-objekti (*JavaScript Object Notation*) põhimõtetel informatsiooni edastamiseks mõeldud kompaktsel meetodil. JWT hoiustatakse eduka sisselogimise järgselt veebilehitseja kohalikku mälu või küpsistesse ning sisaldab informatsiooni erinevatest võtmeväärtuse paaridest. Iga järgnev kasutaja poolt serverile suunatud päring sisaldab endas autoriseerimispaises JWT väärtust, mida server omakorda kontrollib, et ettenähtud juurdepääsu lubada. JWT on krüptograafiliselt allkirjastatud, ehk ainult saatjal ja ettenähtud saajal on ligipääs sisule. Seetõttu on keeruline võtit volitamata osapooltel võltsida. [55]

4.4.3 Arvelehtede loomine

Arvete loomise loogika asub tagarakenduse *PdfFileService* klassis. Arvete koostamiseks kasutatakse Java laiendust nimega *iTextPDF*, mille abil saab mugavalt PDF-dokumente tekitada ning töödelda. Tegemist on võimsa laiendusega, mis pakub lihtsal viisil PDF-dokumentidele teksti, piltide, tabelite ning muude elementide lisamist (vt. joonis 7). [56] Lõputöös tekitatud *PDF*-formaadis arved luuakse kõik eelmainitud laienduse abil. Esmalt

luuakse uus dokumendipõhi, millele järgneb järjekorrapõhine elementide lisamine – st, et esmalt koostatakse arve päis, mis sisaldab arve numbrit, arve saaja informatsiooni ning arvega seonduvat põhiinformatsiooni - esitamise kuupäev, maksetähtaeg jms. Teisena koostatakse tabel, mis sisaldab informatsiooni arve ridade sisust – nt. toote kogus, hind, käibemaks jne. Kolmandana kuvatakse kogu arve hindasid – koondsumma, summa ilma ja koos käibemaksuta ning soodustuse hinnalist väärtust. Viimaseks koostatakse dokumendi jalus, mis sisaldab informatsiooni arve esitaja kohta.

Joonisel 7 on esitatud lühendatud koodinäide *generateInvoicePdf* meetodist, millele väljakutsumisel esmalt edastatakse sisendina vastav arve, et *PDF* faili loomist alustada. Seejärel luuakse *ByteArrayOutputStream* objekt, mis sisestab baite baidimassiivi, et hiljem väljundina salvestatud *PDF* faili kuvada. Samuti koostatakse uus dokumendiobjekt ning kirjutajaobjekt, et elemente selle faili sisuna salvestada. Dokumendi avamisel kutsutakse välja *getInvoiceBodyFields*, millele edastatakse sisendina äsjaloodud dokumendiobjekt ja informatsiooni hoiustav arve. Antud töö raames koostab see meetod arvefailile arveridade päise ning sellega kaasneva sisu.

```
public byte[] generateInvoicePdf(Invoice invoice) throws
Exception {
    ...
    ByteArrayOutputStream outputStream = new
    ByteArrayOutputStream();
    Document document = new Document();
    PdfWriter writer = PdfWriter.getInstance(document,
    outputStream);
    document.open();
    ...
    getInvoiceBodyFields(document, invoice);
    ...
    document.close();
}
```

Joonis 7. iTextPDF abil PDF dokumendi loomine.

Selle meetodi siseselt eraldi tabeli paiknemist faili vaatest ei teostata, vaid määratakse ära arveridade tabeli veergude arv, tabeli ja ridade laiused, veergude sisu staatilise tekstina ning font ja taustavärv. Analoogselt töödeldakse sama meetodi raames ka sissetulev arve informatsioon, mis seejärel lisatakse tabeli veergudesse. Joonisel 8 on esitatud lõik *getInvoiceBodyFields* meetodi sisust.

```
public PdfPTable getInvoiceBodyFields(Document document, Invoice
invoice) throws DocumentException {
...
PdfPTable table = new PdfPTable(7);
table.setWidthPercentage(100.0f);

float[] inputColumnWidth = { 4/5f, 5f, 3/2f, 2f, 2f, 5/4f, 2f };
table.setWidths(inputColumnWidth);

ArrayList<String> informationRow = new ArrayList<String>();
informationRow.add("");
informationRow.add("Toode");
informationRow.add("Kogus");
informationRow.add("Hind (KM-ta)");
informationRow.add("Soodustus");
informationRow.add("KM");
informationRow.add("Summa");

for (String tile : informationRow) {
    PdfPCell cell = new PdfPCell();
    cell.setPadding(3f);
    Font textHeaderFont =
FontFactory.getFont(FontFactory.HELVETICA_BOLD,
10f, BaseColor.BLACK);
cell.setBackgroundColor(new BaseColor(160, 160, 160));
cell.setHorizontalAlignment(Element.ALIGN_LEFT);
cell.setBorderWidth(0f);
cell.setBorderWidthBottom(1/2f);
cell.setPhrase(new Phrase(tile, textHeaderFont));
table.addCell(cell);
}...
}
```

Joonis 8. *getInvoiceBodyFields* meetodi lühendatud koodinäidis.

Joonisel 9 on esitatud eelnevalt kirjeldatud meetodi raames loodud tulemuse näide.

	Toode	Kogus	Hind (KM-ta)	Soodustus	KM	Summa
1	Testtoode - lõputöö Kent-Marcus Pluutus	5	100.00€	0.00%	20.00%	600.00€
2	Testtoode2 - lõputöö Kent-Marcus Pluutus	1	150.00€	20.00%	0.00%	120.00€

Summa (ilma KM): 620.00€
 Käibemaks: 100.00€
 Soodustus: 30.00€
 Kokku tasuda: 720.00€

Joonis 9. *getInvoiceBodyFields* meetodi poolt loodud arveridade tabeli päise näide.

Arve terviklik ülesehitus on peamiselt jaotatud mitmeks omaette tabeliks, mis paiknevad dokumendi erinevates osades, mistõttu on nende loomise loogika valdavalt sarnane.

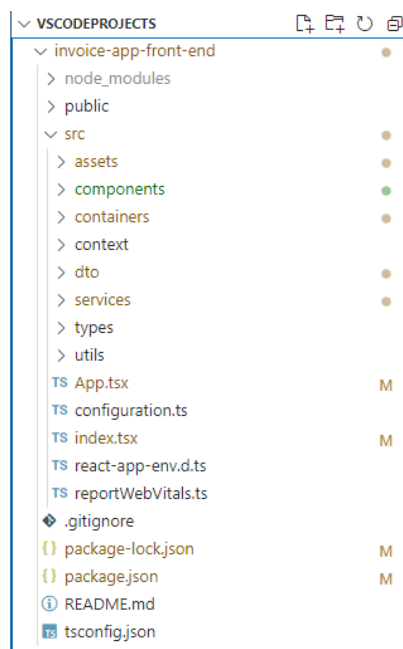
4.5 Eesrakendus

Prototüübi eesrakenduse ja kasutajaliidese loomiseks kasutatakse React.js raamistikku ning TypeScripti programmeerimiskeelt. Elementide kujundamisel on rakendatud lisavahenditena ka Bootstrap.js raamistikku ning font-awesome ikooniteeki, mis võimaldavad eesrakenduse kujundust vormistada ja taaskasutada.

Eesrakenduse põhi luuakse npm-i (Node Package Manager) abil, mille kaudu tekitatakse automaatselt React projekt, mis sisaldab esmaseid vajalikke faile ja kaustasid, nagu näiteks: *App-*, *Index.tsx*, *tsconfig.json* ning *node_modules*. Samuti on hilisemas faasis võimalik npm-i kaudu paigaldada projekti ka ülejäänud vajalikud tööriistad või vahendid.

4.5.1 Eesrakenduse ülesehitus

Eesrakenduses ülesehitusel on kasutatud alamkaustasid, et hõlpsamini vaateid ja nendega seonduvat loogikat organiseeritult hallata. Näiteks *components* kaustas hoitakse kalendrielemendi, päise ja jaluse *.tsx* faile ning *containers* kaust sisaldab alamkaustasid, kus hoiustatakse konkreetse domeeniobjektiga seotud eesrakenduse vaateid ja loogikat (vt. Joonis 10).

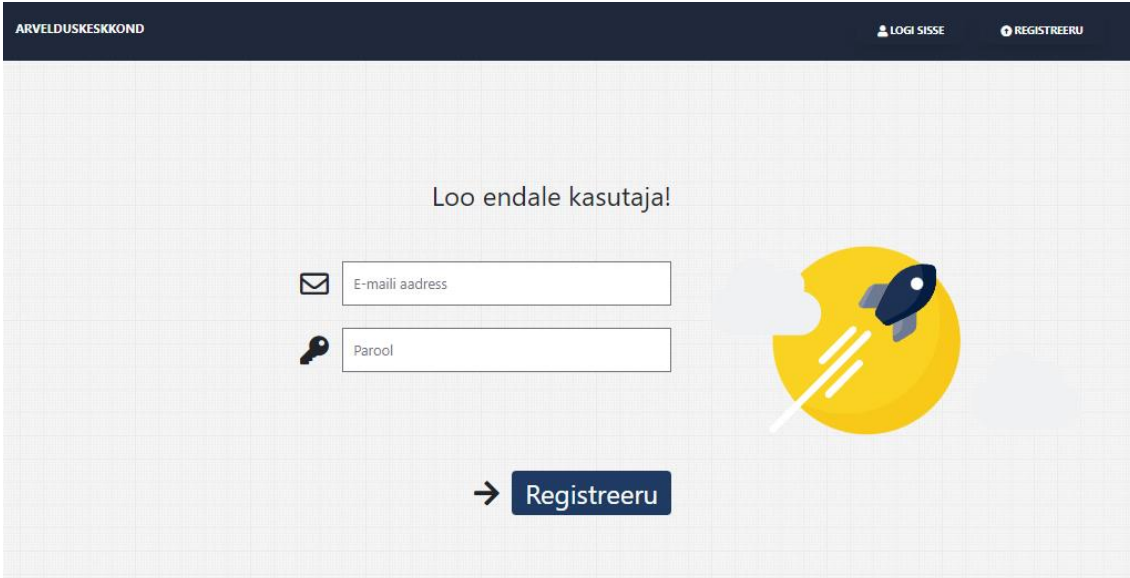


Joonis 10. Eesrakenduse ülesehitus failide tasandil.

Eesrakenduse käivitamisel on peamiseks failiks „*Index.tsx*“, mis esmalt välja kutsutakse. Sellele järgnevalt käivitatakse „*App.tsx*“, mis vastutab omakorda kasutajaliidest moodustavate komponentide laadimise ja konfigureerimise eest. Siin komponendis kutsutakse esile ning luuakse URL-i põhine marsruutimise põhi kõikide vaadete komponentide jaoks. Eelpool kirjeldatud fail on <Router> elemendi komponendi sisse mähitud, et vastavat marsruutimisahelat märgistada ja konkreetseid laetavaid komponente kuvada. Alamkomponentide laadimisel kontrollitakse veebibrauseri kohalikku mällu salvestatud kasutajapõhist autentimistunnuse olemasolu. Selle puudumisel vaate kuvamine on piiratud ning ligipääs puudub.

4.5.2 Kasutajaliides

Esmasel navigeerimisel veebikeskkonda, kuvatakse kasutajale rakenduse esileht, mis annab kiire ülevaate ja tutvustuse platvormist. Menüü navigatsiooniribal on väljatoodud kaks valikut – sisselogimise ning registreerimise nupud, millele klikates suunatakse kasutaja edasi vastavasse etappi. Mõlemal puhul on vaade välimuselt analoogne, omades minimaalseid sisestusväljade põhiseid erinevusi (Joonis 11).



Joonis 11. Platvormi registreerimisvaade.

Esmase registreerimise puhul suunatakse kasutaja edasi ettevõtteandmete sisestamise faasi (Joonis 12). Siinkohal tuleb edasiste funktsionaalsuste mugavdatud kasutamise tarbeks konto haldajal sisestada ettevõtte peamised kontaktandmed. Hilisemates

etappides on võimalik vastavaid andmeid uuendada ja muuta.

ARVELDUSKESKKOND LOGI VÄLJA

SINU ETTEVÖTTE KONTAKTANDMED

* Ettevõtte nimi: * E-maili aadress:

* Ettevõtte juriidiline aadress:

* Registrikood: * Telefoninumber:

* KMKR number: Pangakonto IBAN:

Pank (i.e. Swedbank, SEB)

[Salvesta](#)

Joonis 12. Registreerimisvaatele järgnev kontaktandmete vormi vaade.

Kontaktandmete edukal sisestamisel suunatakse kasutaja edasi veebikeskkonna esilehele, mis erineb pisut sisselogimisele eelnenud vaatest. Lehel on kuvatud infoväljad ja navigatsiooniriba valikud, mille kaudu on võimalik platvormi funktsionaalsustele ligi pääseda.

Esimene valik menüüribas on „**Kontaktid**“ – selle vaate kaudu (Joonis 13), on võimalik kasutajal hallata nii enda kontaktandmeid, kui ka lisada modaalvaate (vt. Joonis 14) kaudu teisi ettevõtteid ning neid eemaldada.

ARVELDUSKESKKOND KONTAKTID ARVED SISSEKANDED LOGI VÄLJA

[LISA UUS KONTAKT](#) [HALDA ENDA KONTAKTANDMED](#)

Kontakti nimi	E-maili aadress	Aadress	Registrikood	Telefoninumber	KMKR number
Tallinna Tehnikaülikool	ttü@ttü.ee	Ehitajate tee 5, 19086 Tallinn	EESTEKOOD123	620 2002	EEstestkmkr ✕

Joonis 13. Rakenduse kontaktiraamatu vaade.

Joonis 14. Modaalvaade kontaktide lisamiseks.

Teine valik menüüribal, „**Arved**“, suunab kasutaja otse arvete arhiivivaatesse. Tabeli formaadis kuvatakse kasutaja eelnevalt koostatud arved ja nende baasinformatsioon–arvenumber, viitenumber, kuupäevad, saaja jms, ning võimalus arvet *PDF* kujul alla laadida – (vt. Joonis 15).

Arve number	Saaja	Esitamise kuupäev	Maksetähtaeg	Kirjeldus	Viitenumber	Olek
BT-0123	Tallinna Tehnikaülikool	01/05/2023	26/05/2023	Tegemist on test arvega	VT-0123	Kinnitatud
AB-112	Tallinna Tehnikaülikool	02/05/2023	31/05/2023	Test arve selgitus - demo	VT-912	Kinnita

Joonis 15. Prototüübi arvete arhiivivaade.

Samuti asetseb tabelvaate kohal ka nupp „**Lisa uus arve**“, millele klikates suunatakse kasutaja arvete koostamise leheküljele.

Arvete koostamisel päritakse esmalt kasutaja käest arve saaja informatsioon (vt. **Error! Reference source not found.**) – siinkohal on rakendatud rippmenüüd, mis hoiab endas kõikide eelnevalt salvestatud kontaktide informatsiooni.

Arve saaja nimi Otsi loetelust salvestatud kontakte	Arve saaja nimi Tallinna Tehnikaülikool
Tallinna Tehnikaülikool	Ettevõtte juriidiline aadress Ehitajate tee 5, 19086 Tallinn
TestContact	Saaja registrikood EETESTKOOD123
* i.e 0000000	Saaja KMKR number 11111111
Saaja KMKR number	
* i.e EE0000000	

Joonis 16. Prototüüpikenduse arve saaja rippmenüü ning valikujärgne andmeväljade täitumise vaade.

Peale valiku teostamist täidetakse kohustuslikud andmeväljad koheselt andmebaasis hoiustatud informatsiooniga. Sellele järgneb arvele kuuluva kohustusliku informatsiooni sisestamine – arve number, viitenumber, kuupäevad, maksetingimused, arveread jne. Täpsem ülevaade terviklikust arvete loomise kasutajaliidesest on esitatud lisan 8.

Lisafunktsionaalsustena saab kasutaja otsinguriba kaudu märksõnade kaupa vajalikke arveid üles otsida ning tabelimenüü sorteerimisnuppude abil valikuid sorteerida. Tabeli viimases lahtris kuvatakse iga arve kohta allalaadimisnupp, millele vajutades avaneb tagarakenduse poolt vormistatud *PDF*-formaadis arvefail. Arve näidis on väljatoodud lisan 9.

Viimaseks peavaateks on koostatud sissekannete lehekülg, kus võimaldatakse kasutajal jälgida müügi- ning ostuarvete ülevaadet (vt. Joonis 17).

Arve number	Arve saaja	Esitamise kuupäev	Kirjeldus	KM-vaba summa	KM	Arve summa (KM)
BT-0123	Tallinna Tehnikaülikool	01/05/2023	Tegemist on testi arvega	620.00 €	100.00 €	720.00 €
AB-112	Tallinna Tehnikaülikool	02/05/2023	Test arve selgitus - demo	200.98 €	18.08 €	219.06 €
Summa (KM-ta):	820.98 €					
Kokku summa:	939.06 €					

Joonis 17. Prototüübi sissekannete vaade

Müügiarvete rippmenüü all kuvatakse kõik arved, mis on eelnevalt „Arved“ vaatest kasutaja poolt „Kinnitatud“ saatuse saanud. Eraldi müügiarvete sisestamise funktsionaalsust siinkohal ei võimaldata. Rippmenüü viimastel ridadel kuvatakse kõikide koostatud arvete summad koos- ning ilma käibemaksuta, et anda kiire arusaam esitatud arvete summadest.

Ostuarvete rippmenüü omab analoogset loogikat, kuid sinna kuvatakse kasutajale temapoolt eraldi sisestatud ostuarvete ülevaade ning koondsummad. Ostuarvete sisestamine toimub eraldi „Sisesta ostuarve“ nupu kaudu. Nupule vajutades avaneb modaalvaade uue ostuarve sisse kandmiseks (vt. joonis 18.).

Ostuarve lisamine: X

* Otsi loetelust salvestatud kontakte

* Arve nr: 05/16/2023

* Toode: * Kogus:

* Hind (KM-ta): * KM: * Summa:

* Kirjeldus:

Käibemaksuta summa: 0.00€
Käibemaks: 0.00€
Koondsumma: 0.00€

Lisa

Joonis 18. Modaalvaade ostuarve lisamiseks.

Andmeväljadel, mis eelmainitud joonistel andmesisestusetappides kasutajale kuvatakse, teostatakse enne jätkamist kontroll, et sisestatav informatsioon oleks formaadi poolest kooskõlas eeldustega. Formaadi vastu eksimisel, või andmeväljade tühjaks jätmisel kuvatakse eesrakenduses kasutajale punases kirjas vastav veateade. Mõningaid erandeid esineb andmeväljadel, mis ei ole kohustuslikud – nt arve selgitus, mis puhul sobib andmevälja tühjaks jätmine.

5 Töö tulemused ja edasiarendused

Käesolevas peatükis tutvustatakse valminud rakenduse prototüüpi ning autori poolt väljatoodud edasiarenduste võimalusi.

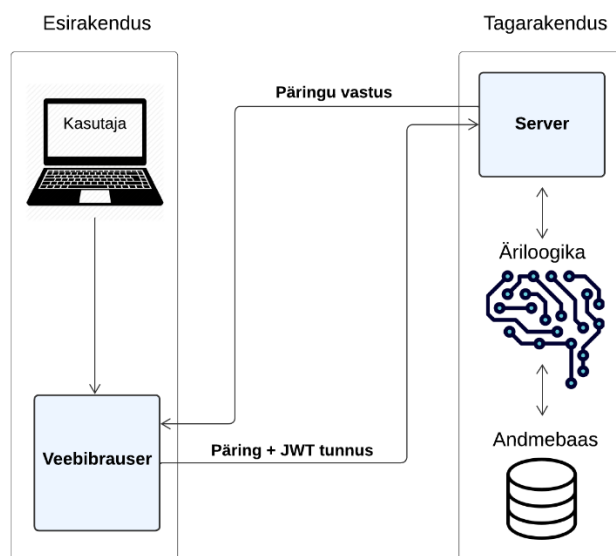
5.1 Töö tulemused

Lõputöö tulemusena arendati kõikidele eeltingimustele ning nõuetele vastav REST-api liidestuse põhine arveldusrakenduse prototüüp. Prototüüp koosneb eesrakendusest ning selle alamvaadetest, mitmekihiliselt ülesehitatud tagarakendusest ja andmebaasist.

Analüüsi käigus seatud eeltingimused ja nõuded said täidetud ning rakenduse poolt vormistatav arveformaad on kooskõlas riigi poolt sätestatud arvetingimustega.

Rakenduse arendamisele eelnevas etapis analüüsiti sobivaid programmeerimiskeeli ning raamistikke. Kasutajaliidese koostamisel koguti inspiratsiooni turul leiduvatest alternatiivplatvormide analüüsi kaudu ning pakuti lahendust esinenud kitsaskohtadele.

Loodud rakenduse andmesuhtlus saab alguse kasutajaliideseast. Funktsionaalsuste esile kutsumisel (nt. uue kontakti sisestamine), edastatakse sisestatud andmetega andmekogum esirakenduse kaudu päringuna tagarakenduse serverile. Iga päringuga kaasatakse vastava kasutaja autentimistunnus. Seejärel töödeldakse päringu sisu tagarakenduse ärikihtides, millele autentimistunnuse kontrollile järgnevalt salvestatakse, või muudetakse väljasid andmebaasis. Tänu sellele välditakse olukordasid, kus kasutaja saab edukalt pärida mõne teise kasutaja andmeid. Päringu töötlemisel tagastab teenusekiht esirakendusele vastuspäringu. Eduka tulemuse puhul teostatakse muudatused esirakenduses vaates, et kasutajaliidese informatsiooni kuvada. Andmesuhtluse skeem on väljatoodud joonisel 19.



Joonis 19. Rakenduse andmesuhtluse skeem.

Tulemusena valmis platvorm, mille abil on võimalik uuel kasutajal registreerida, lisada kontakte ning koostada arve paari minuti jooksul. Välimuselt on kasutatud ühtset stiilimustrit ning kasutajaliidese vaated omavad sarnast välimust, mistõttu on äsja liitunud kasutajal kerge keskkonda sisse elada.

5.2 Edasiarendusvõimalused

Prototüübile on võimalus luua hulgaliselt edasiarendusi, mis esialgsesse skooopi ning ajaraami ei mahtunud, kuid parendaks kasutajakogemust ning rakenduse võimekust.

Oluliseks prioriteediks, et turul asetsevate platvormidega konkureerida, tuleks prototüübile arendada e-arvete loomise võimekus. Seeläbi omavad arved masinloetavuse võimekust. Kasutajakogemuse parendamiseks, saab rakendada ka automaatset e-maili saatmist, et nupuvajutusel oleks võimalik kasutajal koostatud arveid otse klientidele esitada. Majandustehingute haldamisvaatesse on võimalus tekitada lisaks üldine väljatrüki allalaadimise funktsionaalsus, et rakenduseväliselt sisestatud andmeid eraldi hallata ja nt. aastaaruannete koostamisel andmetöötlust veelgi mugavdada. Lisaks, tuleks arvestada erinevate nüanssidega, mis võivad arvete koostamisel osapoolte vahel esineda ning neid arvete koostamisel kajastada. Kuna hetkel hoiustatakse prototüübi andmeid lokaalse serverihalduri andmebaasis, tuleks andmestik ning sellega seonduv juurutada ümber välisesse andmebaasi. Kihtidevahelises suhtluses tuleks abiks ka DTO mustri

kasutamine. Kindlasti tuleb rakenduse stabiilsuse ning funktsionaalsuste töökindluse tagamiseks arendada prototüübile detailsemad testklassid ning veateadete kavad. Seeläbi tagatakse kvaliteetsem lõpptulemus ning säästetakse pikemas perspektiivis aega, et võimalikke defekte ja vigasid lihtsamini tuvastada.

6 Kokkuvõte

Käesoleva lõputöö eesmärk oli luua kasutajapõhine arveldusrakenduse prototüüp arvete, kontaktide ning majandustehingute haldamiseks.

Lõputöö käigus uuriti alternatiivseid platvorme ning nende tugevaid ja nõrku külgi, mille põhjal koostati SWOT analüüs ja seati paika prototüübi lähtetingimused koos nõuetega. Lahenduse teostamiseks võrreldi erinevaid andmebaasi, serveripoolse- ja klientrakenduse tehnoloogiaid. Esirakenduse tehnoloogiaks valiti TypeScript programmeerimiskeel ning Reacti raamistik. Tagarakenduse tehnoloogiaks osutusid Java programmeerimiskeel koos Spring Boot raamistiku ning H2 andmebaasimootoriga.

Praktilises osas kirjeldati esmalt arendusele eelnevat eeltööd ning andmebaasimudelit. Ühtlasi toodi välja ka esi- ning tagarakenduse seadistamine ja arhitektuuriline struktuur, et ülesehitusest parem arusaam luua.

Lõputöö tulemusena loodi kõikidele lähtetingimusele ja nõuetele vastav veebirakenduse prototüüp. Valminud lahendus võimaldab väikeettevõtjatel teiste ettevõtete kontaktandmeid salvestada, nende andmetega arveid koostada ning ostu- ja müügiga seonduvaid majandustehinguid hallata. Iga eelmainitud funktsionaalsuse haldamise jaoks loodi iseseisev kasutajaliidese vaade.

Lõputöö esitamise hetkel ei ole rakendus avalikult kasutajaliidesena ligipääsetav, kuid tulevikus on plaanis platvorm avalikuks muuta ning valminud toodet huvilistele pakkuda. Samuti jätkub rakenduse edasine arendamine. Lõputöö raames arendatud prototüübi esi- ning tagarakenduse koodibaas on väljatoodud lisa 10.

Kasutatud kirjandus

- [1] Riigikogu, „Riigiteataja,“ 2022. [Võrgumaterjal]. Available: <https://www.riigiteataja.ee/akt/RPS#para7>. [Kasutatud 3 2023].
- [2] Adam, „Studiosoftware,“ Studiosoftware, [Võrgumaterjal]. Available: <https://studiosoftware.com/blog/how-to-start-a-successful-it-project-six-essential-steps/>. [Kasutatud 03 2023].
- [3] „Sumup,“ Sumup, [Võrgumaterjal]. Available: <https://www.sumup.com/en-gb/invoices/invoicing-essentials/is-it-mandatory-to-use-invoicing-software/>. [Kasutatud 03 2022].
- [4] I. Oy, „isolta,“ [Võrgumaterjal]. Available: <https://secure.isolta.com/fi/login>.
- [5] F. BV, „onArvel,“ [Võrgumaterjal]. Available: <https://www.onarvel.ee/>.
- [6] „Merit Aktiva,“ Merit, [Võrgumaterjal]. Available: <https://www.merit.ee/>. [Kasutatud 20 3 2023].
- [7] „Banqup,“ unifiedpostgroup, [Võrgumaterjal]. Available: <https://www.banqup.com/ee/et-ee>.
- [8] Microsoft, „Microsoft,“ Microsoft, [Võrgumaterjal]. Available: <https://www.microsoft.com/en-us/microsoft-365/excel>. [Kasutatud 3 2023].
- [9] Javatpoint, „Javatpoint,“ Javatpoint, [Võrgumaterjal]. Available: <https://www.javatpoint.com/10-disadvantages-of-microsoft-excel>. [Kasutatud 3 2023].
- [10] M. T. C. Team, „Mindtools,“ [Võrgumaterjal]. Available: <https://www.mindtools.com/ambtj63/swot-analysis>. [Kasutatud 3 2023].
- [11] Riigikogu, „Riigiteataja,“ Justiitsministeerium, 2022. [Võrgumaterjal]. Available: <https://www.riigiteataja.ee/akt/KMS#para371g7>. [Kasutatud 3 2023].
- [12] Atlassian, „Atlassian,“ Atlassian, [Võrgumaterjal]. Available: <https://www.atlassian.com/work-management/project-management/software>. [Kasutatud 03 2023].
- [13] Clickup, „Clickup,“ Clickup, [Võrgumaterjal]. Available: <https://clickup.com/blog/10-best-project-management-tools/>. [Kasutatud 3 2023].
- [14] Asana, „Asana,“ Asana, Inc., [Võrgumaterjal]. Available: <https://asana.com/uses>. [Kasutatud 3 2023].
- [15] Asana, „Asana,“ Asana, Inc., [Võrgumaterjal]. Available: <https://asana.com/apps>. [Kasutatud 3 2023].
- [16] „Tpptechology,“ TPP technology, 27 9 2021. [Võrgumaterjal]. Available: <https://www.tpptechology.com/blog/software-development-choosing-the-right-technology-for-your-project/#:~:text=Choosing%20the%20right%20technologies%20is,well%20as%20update%20and%20maintenance..> [Kasutatud 16 3 2023].

- [17] „Riders,“ Riders.ai, [Võrgumaterjal]. Available: <https://riders.ai/en-blog/most-used-programming-languages-in-the-world>. [Kasutatud 3 2023].
- [18] D. Ó. Tuama, „Codeinstitute,“ Codeinstitute, [Võrgumaterjal]. Available: <https://codeinstitute.net/global/blog/what-is-a-programming-language/>. [Kasutatud 3 2023].
- [19] Microsoft, „Microsoft,“ Microsoft, 13 2 2023. [Võrgumaterjal]. Available: <https://learn.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/>. [Kasutatud 3 2023].
- [20] Microsoft, „Microsoft,“ Microsoft, 13 2 2023. [Võrgumaterjal]. Available: <https://learn.microsoft.com/en-us/dotnet/csharp/asynchronous-programming/async-scenarios>. [Kasutatud 3 2023].
- [21] „Microsoft,“ Microsoft Corporation, 2023. [Võrgumaterjal]. Available: <https://dotnet.microsoft.com/en-us/>. [Kasutatud 18 3 2023].
- [22] „TechMango,“ 1 6 2022. [Võrgumaterjal]. Available: <https://www.techmango.net/what-are-the-benefits-of-upgrading-to-the-net-framework>. [Kasutatud 18 3 2023].
- [23] L. SINGLA, „netsolutions,“ 16 11 2022. [Võrgumaterjal]. Available: <https://www.netsolutions.com/insights/what-is-php/#:~:text=PHP%20is%20a%20server%20side,in%20the%20web%20development%20world..> [Kasutatud 3 2023].
- [24] „Geeksforgeeks,“ Geeksforgeeks, 6 2022. [Võrgumaterjal]. Available: <https://www.geeksforgeeks.org/advantages-and-disadvantages-of-php/>. [Kasutatud 3 2023].
- [25] „dev,“ 3 2023. [Võrgumaterjal]. Available: <https://dev.to/t/php>. [Kasutatud 3 2023].
- [26] „Laravel,“ 2023. [Võrgumaterjal]. Available: <https://laravel.com/>. [Kasutatud 18 3 2023].
- [27] T. Otwell, „Laravel,“ 2023. [Võrgumaterjal]. Available: <https://laravel.com/docs/10.x/blade>. [Kasutatud 18 3 2023].
- [28] „Oracle,“ Oracle, [Võrgumaterjal]. Available: <https://www.oracle.com/java/moved-by-java/timeline/>. [Kasutatud 09 04 2022].
- [29] „IBM,“ International Business Machines Corporation, 20 1 2023. [Võrgumaterjal]. Available: <https://www.ibm.com/docs/en/aix/7.1?topic=monitoring-advantages-java>. [Kasutatud 3 2023].
- [30] „Javatpoint,“ Javatpoint, [Võrgumaterjal]. Available: <https://www.javatpoint.com/jvm-java-virtual-machine>. [Kasutatud 15 3 2023].
- [31] „Geeksforgeeks,“ 9 2022. [Võrgumaterjal]. Available: <https://www.geeksforgeeks.org/top-10-most-popular-java-frameworks-for-web-development/>. [Kasutatud 18 3 2023].
- [32] „Javatpoint,“ [Võrgumaterjal]. Available: <https://www.javatpoint.com/spring-mvc-tutorial>. [Kasutatud 18 3 2023].
- [33] „IBM,“ IBM, [Võrgumaterjal]. Available: <https://www.ibm.com/docs/en/aix/7.1?topic=monitoring-advantages-java>. [Kasutatud 09 04 2022].

- [34] „Tiobe,“ Tiobe Software, 3 2023. [Võrgumaterjal]. Available: <https://www.tiobe.com/tiobe-index/>. [Kasutatud 16 3 2023].
- [35] Airfocus, [Võrgumaterjal]. Available: <https://airfocus.com/glossary/what-is-a-front-end/>. [Kasutatud 19 3 2023].
- [36] „hostinger,“ [Võrgumaterjal]. Available: hostinger.com/tutorials/what-is-javascript. [Kasutatud 19 3 2023].
- [37] „typescriptlang,“ Microsoft, [Võrgumaterjal]. Available: <https://www.typescriptlang.org/>. [Kasutatud 19 3 2023].
- [38] J. Hartman, „guru99,“ 4 3 2023. [Võrgumaterjal]. Available: <https://www.guru99.com/typescript-vs-javascript.html>. [Kasutatud 19 3 2023].
- [39] „emizentech,“ 29 7 2022. [Võrgumaterjal]. Available: <https://www.emizentech.com/blog/react-with-typescript-or-javascript.html>. [Kasutatud 19 3 2023].
- [40] „React,“ Meta Platforms, Inc., 2023. [Võrgumaterjal]. Available: <https://react.dev/>. [Kasutatud 19 3 2023].
- [41] „IBM,“ International Business Machines Corporation, [Võrgumaterjal]. Available: <https://www.ibm.com/docs/en/zos-basic-skills?topic=zos-what-is-database-management-system>. [Kasutatud 19 3 2023].
- [42] „H2database,“ 2023. [Võrgumaterjal]. Available: <https://www.h2database.com/html/main.html>. [Kasutatud 19 3 2023].
- [43] „Tutorialspoint,“ [Võrgumaterjal]. Available: https://www.tutorialspoint.com/h2_database/h2_database_introduction.htm. [Kasutatud 19 3 2023].
- [44] „IBM,“ International Business Machines Corporation, 1 1 2023. [Võrgumaterjal]. Available: <https://www.ibm.com/docs/he/iodg/11.3?topic=reference-primary-keys>. [Kasutatud 19 3 2023].
- [45] C. Walls, „Manning.com,“ [Võrgumaterjal]. Available: <https://freecontent.manning.com/wp-content/uploads/initializing-a-spring-boot-project-with-spring-initializr.pdf>. [Kasutatud 20 3 2023].
- [46] P. E. Staff, „PacktPub,“ 17 5 2018. [Võrgumaterjal]. Available: <https://hub.packtpub.com/what-is-multi-layered-software-architecture/>. [Kasutatud 3 4 2023].
- [47] J. Hartman, „Guru99,“ 4 3 2023. [Võrgumaterjal]. Available: <https://www.guru99.com/java-data-abstraction.html>. [Kasutatud 3 4 2023].
- [48] R. Y. Esplanada, „Stackabuse,“ [Võrgumaterjal]. Available: <https://stackabuse.com/controller-and-restcontroller-annotations-in-spring-boot/>. [Kasutatud 3 4 2023].
- [49] baeldung, „baeldung,“ 9 11 2022. [Võrgumaterjal]. Available: <https://www.baeldung.com/spring-service-layer-validation#:~:text=A%20service%20layer%20is%20a,the%20controller%20and%20service%20layers..> [Kasutatud 3 4 2023].
- [50] „GeeksForGeeks,“ 24 1 2023. [Võrgumaterjal]. Available: <https://www.geeksforgeeks.org/data-access-layer/>. [Kasutatud 3 4 2023].

- [51] D. Ó. Tuama, „Codeinstitute,“ [Võrgumaterjal]. Available: <https://codeinstitute.net/global/blog/object-relational-mapping/>. [Kasutatud 3 4 2023].
- [52] baeldung, „Baeldung,“ 22 12 2022. [Võrgumaterjal]. Available: <https://www.baeldung.com/java-dto-pattern#:~:text=DTOs%20come%20in%20handy%20in,roundtrip%20between%20client%20and%20server..> [Kasutatud 3 4 2023].
- [53] „Redhat,“ 8 5 2020. [Võrgumaterjal]. Available: <https://www.redhat.com/en/topics/api/what-is-a-rest-api>. [Kasutatud 4 4 2023].
- [54] „Spring,“ 2023. [Võrgumaterjal]. Available: <https://docs.spring.io/spring-security/reference/index.html>. [Kasutatud 4 4 2023].
- [55] „jwt,“ auth0, 2023. [Võrgumaterjal]. Available: <https://jwt.io/introduction>. [Kasutatud 4 4 2023].
- [56] „itextpdf,“ apryse, [Võrgumaterjal]. Available: <https://itextpdf.com/>. [Kasutatud 1 5 2023].
- [57] „Oracle,“ Oracle Corporation, 2023. [Võrgumaterjal]. Available: <https://www.oracle.com/java/moved-by-java/timeline/>. [Kasutatud 15 3 2023].
- [58] „IBM,“ International Business Machines Corporation, [Võrgumaterjal]. Available: [https://www.ibm.com/topics/java-spring-boot#:~:text=Java%20Spring%20Boot%20\(Spring%20Boot,ability%20to%20create%20standalone%20applications](https://www.ibm.com/topics/java-spring-boot#:~:text=Java%20Spring%20Boot%20(Spring%20Boot,ability%20to%20create%20standalone%20applications). [Kasutatud 18 3 2023].
- [59] I. VMware, „Spring,“ [Võrgumaterjal]. Available: <https://spring.io/>. [Kasutatud 3 4 2023].

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks¹

Mina, Kent-Marcus Pluutus

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

15.05.2023

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 punktile 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.

Lisa 2 – Isolta arveldusrakenduse arvete koostamise eelvaade


OÜ Teflon
Õie 2/1-4
11615 Tallinn

Sisesta arve saaja nimi

Arve number	19
Viitenumber	194
Arve kuupäev	14.03.2023
Maksetingimus	14 päeva
Maksetähtaeg	28.03.2023
Arve väljastas	
Selgitus	

[Näita arve lisainfot](#)

Kirjeldus (enne arve ridasid)

Toode nr	Kirjeldus	Ühiku hind	Kogus ja ühik	Soodus	Kokku
 test	test sisestamine	100,00 € KM 20% - 120,00 €	1 1		100,00 € 120,00 €

[+ Lisa uus rida](#) Arve read 1 Ühikuid 1 Valuuta EUR

Käibemaksuta summa	100,00 €
KM	20,00 €
Kokku	120,00 €

Tekst pärast arve ridasid

Lisa 3 – OnArvel veebikeskkonna poolt loodud arve näidis

Test OÜ

België

kent.marcus.pp@gmail.com

Test Ettevõte
T.a.v. Test nimi
Test tänav 1
11615 Tallinn
Estland

FACTUUR F230002

Datum: 13-03-2023 Vervaldatum: 27-03-2023 Btw-nummer klant: 123456789

Omschrijving	Aantal	Eenheidsprijs	BTW	Totaal excl.
Testteenus	1	€ 0,00	21%	€ 0,00
Totaal excl.:				€ 0,00
21% btw op € 0,00				+ € 0,00
Totaal te betalen:				€ 0,00

Bedankt voor uw vertrouwen in fups oü. Gelieve het factuurbedrag binnen de 14 dagen over te schrijven op rekeningnummer %iban% onder vermelding van het factuurnummer F230002.

Lisa 4 – Merit Aktiva arvete koostamise vaade

Klient	Kuupäev	Maksetähtpäev	Kande kuupäev					
<input type="text"/>	20.03.2023	27.03.2023	20.03.2023					
Address	Arve nr	Valuuta						
<input type="text"/>	1001	EUR						
	Viitenumber							
	10016							
Maksja	Kontakt							
<input type="text"/>	<input type="text"/>							
<input type="checkbox"/> Põhivara müük		<input type="checkbox"/> Hinnad km-ga						
Artikkel	Kirjeldus	Kog...	Ühik	Hind	Summa ...	A...	KM	Konto
		0,00		0,0000000	0,00	0,00		
Uus rida								
Märgi tasumine				Summa v.a km 0,00				
				KM kokku 0,00				
				Ümardus 0 0,00				
				KOKKU 0,00				
				Tasutud 0,00				
Lisainfo	Kopeeri	Koosta kreditarve	Salvesta	Salvesta ja lisa uus	Katkesta			

Lisa 5 - H2 andmebaasimootori veebikonsooli vaated

English ▼ Preferences Tools Help

Login

Saved Settings: Generic H2 (Embedded) ▼

Setting Name: Generic H2 (Embedded) Save Remove

Driver Class: org.h2.Driver

JDBC URL: jdbc:h2:~/H2db/InvoiceManagementDb

User Name: sa

Password:

Connect Test Connection

Auto commit Max rows: 1000 Auto complete Off Auto select On

Run Run Selected Auto complete Clear SQL statement:

- jdbc:h2:~/H2db/InvoiceManagement
- COMPANY_CONTACT_INFC
- CONTACT_TYPE
- CUSTOMER
- INVOICE
- INVOICE_ROW
- INVOICE_TYPE
- ROLE
- STATUS_TYPE
- INFORMATION_SCHEMA
- Users
- H2 2.1.214 (2022-06-13)

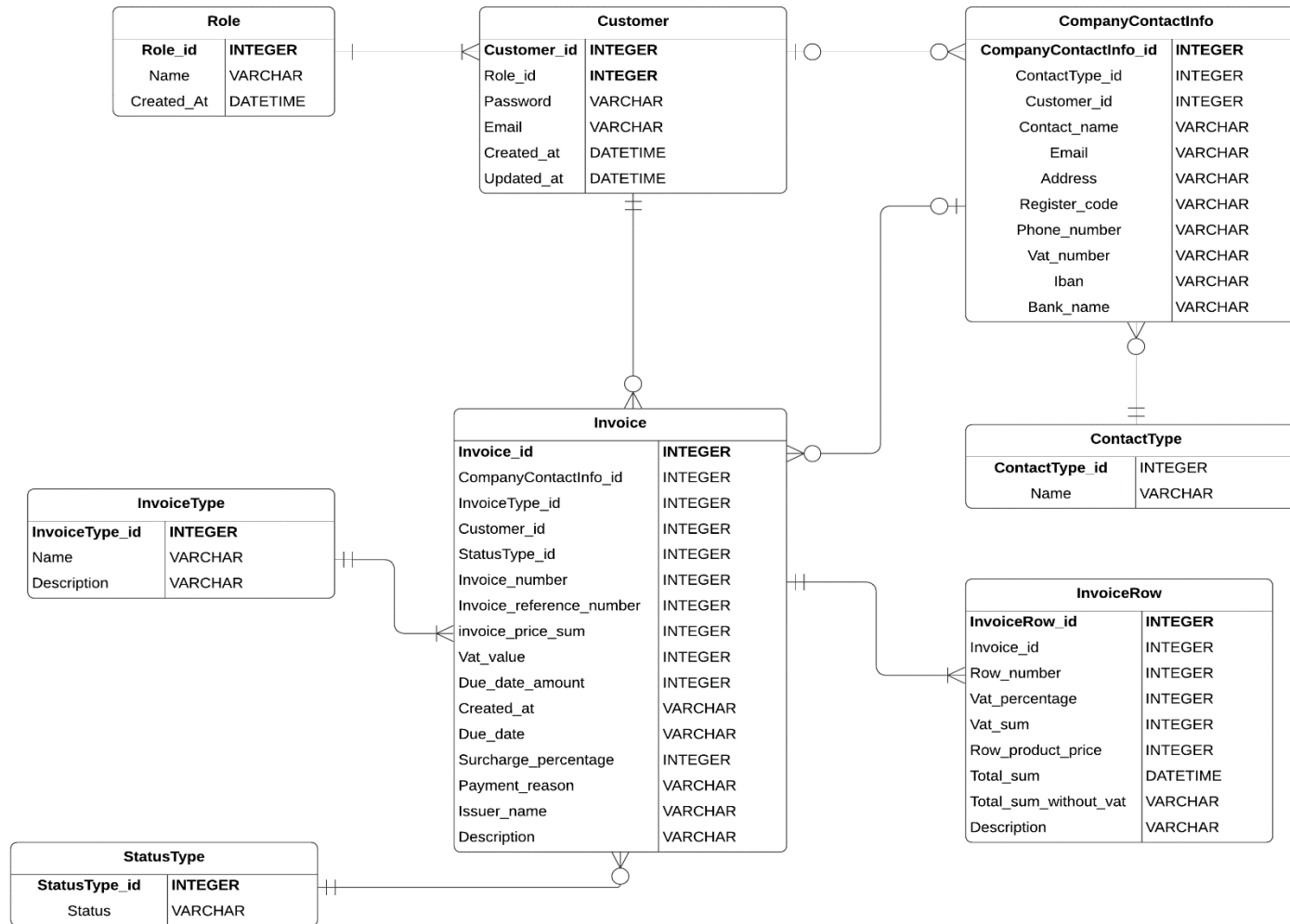
Important Commands

	Displays this Help Page
	Shows the Command History
Ctrl+Enter	Executes the current SQL statement
Shift+Enter	Executes the SQL statement defined by the text selection
Ctrl+Space	Auto complete
	Disconnects from the database

Sample SQL Script


Delete the table if it exists	DROP TABLE IF EXISTS TEST;
Create a new table with ID and NAME columns	CREATE TABLE TEST(ID INT PRIMARY KEY, NAME VARCHAR(255));
Add a new row	INSERT INTO TEST VALUES(1, 'Hello');
Add another row	INSERT INTO TEST VALUES(2, 'World');
Query the table	SELECT * FROM TEST ORDER BY ID;
Change data in a row	UPDATE TEST SET NAME='HI' WHERE ID=1;
Remove a row	DELETE FROM TEST WHERE ID=2;
Help	HELP ...


Lisa 6 – prototüübi andmebaasimudeli ERD skeem



Lisa 7 – Kasutajaloo näide

Roll + kasutaja andmebaasitabelite koostamine ja kasutajatepõhine registreerimine

Assignee  Kent × Recently assigned ▾


Due date  1 Mar ×

Projects ■ Arveldusrakenduse prototüübi are... ▾ Untitled section ▾

Add to projects

Dependencies Add dependencies

My Tasks Fields

 Priority High




Description

Kliendina soovin ma luua endale isiklikku kontot, et saaksin oma ettevõtte andmeid sisestada.

- Sisselogimine toimub e-maili põhised - kasutaja olem koosneb P-key, role f-key, password, email, created_at ja updated_at andmeväljadest.
 - Rolli tabel hoiustab kasutajatüüpe - koosneb nimiväärtusest ja created_at andmeväljadest

Sisselogitav kasutaja erineb ettevõttest endast - hiljem koostatakse ettevõtte infot sisaldav tabel (e-mailid ei pruugi alati ühtida).

Subtasks

- Koostada kasutajatüübi ja rolli andmebaasi olemid   
- Määrata tabelite atribuutide väljad
- Tekitada olemid tagarakenduses

Lisa 8 – Prototüübi arvete loomise eesrakenduse vaade

ARVELDUSKESKKOND [KONTAKTID](#) [ARVED](#) [SISSEKANDED](#) [LOGI VÄLJA](#)

Arve saaja nimi
* Otsi loetelust salvestatud kontakte

Ettevõtte juriidiline aadress
* i.e Ehitajate tee 5, 19086 Tallinn, Estonia

Saaja registrikood
* i.e 0000000

Saaja KMKR number
* i.e EE0000000

Arve number
* i.e 112

Viitenumber
* i.e 112

Arve esitamise kuupäev
05/01/2023

Maksetähtaeg
05/01/2023

Selgitus
Selgitus

Number	Toode	Toote hind KM-ta(€)	Kogus (tk)	Soodus (%)	KM(%)	Kokku (€)
#	*	*	*	*	*	0
#	*	*	*	*	*	0

+

Käibemaksuta summa: 0 €
Käibemaks: 0 €
Soodustus: 0 €
Koondsumma: 0 €

[Tagasi](#) [Salvesta](#)

Lõputöö 2023 - Kent-Marcus Piuutus

Lisa 9 - Tagarakenduses loodud arve näidis

Arve nr: BT-0123

Arve saaja

Tallinna Tehnikaülikool
Ehitajate tee 5, 19086 Tallinn
ttü@ttü.ee
620 2002

Viitenumber: VT-0123

Kuupäev: 01/05/2023

Maksetähtaeg: 26/05/2023

Arve väljastaja: Test esitaja nimi

	Toode	Kogus	Hind (KM-ta)	Soodustus	KM	Summa
1	Testtoode - lõputöö Kent-Marcus Pluutus	5	100.00€	0.00%	20.00%	600.00€
2	Testtoode2 - lõputöö Kent-Marcus Pluutus	1	150.00€	20.00%	0.00%	120.00€

Summa (ilma KM): 620.00€
Käibemaks: 100.00€
Soodustus: 30.00€
Kokku tasuda: 720.00€

Aadress: Valdeku 13, 11621 Tallinn
Telefon: 55555555
Reg nr: 11111111
KMKR nr: EE123123123

test@mail.ee

IBAN: EExxxxxxxxxxxxxx
Pank: Swedbank AS

Lisa 10. Prototüübi koodi repositooriumi lingid

Esirakendus:

<https://github.com/kentmarcusp/VSCoDeProjects/tree/master/invoice-app-front-end>

Tagarakendus:

<https://github.com/kentmarcusp/InvoiceManagementBackEnd/tree/main>