

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Toomas Parts 186062

Arvuti komplekteerimise veebirakendus hobikasutajatele

Bakalaureusetöö

Juhendaja: Kaido Kikkas
Tehnikateaduste
doktor

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: Toomas Parts

Annotatsioon

Käesoleva bakalaureuse lõputöö eesmärgiks on uurida arvuti komplekteerimise protsessi ning projekteerida selle lihtsustamist võimaldav veebirakendus.

Töö sisaldab endas ise arvuti komplekteerimise eeliste ja raskuste analüüsi, vanade arvutite täiendamisevõimaluste kirjeldamist. Probleemi lahendamiseks projekteeriti veebirakendus.

Töö tulemuseks on rakenduse prototüüp, mis võimaldaks kasutada projekteerimise käigus määratud funktsionaalsusi.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 29 leheküljel, 8 peatükki, 5 joonist, 0 tabelit.

Abstract

The aim of this bachelor's thesis is to study the computer assembly process and to design a web application that enables its simplification.

The main contents of this thesis are analysis of the advantages and difficulties of assembling a computer, a description of the possibilities of upgrading old computers. A web application was designed to solve the given problems.

The result of the work is a prototype of the web application, which would allow the usage of the functionalities that were specified during the design process.

The thesis is in Estonian and contains 29 pages of text, 8 chapters, 5 figures, 0 tables.

Lühendite ja mõistete sõnastik

API	<i>Application Programming Interface</i> Rakenduse programmeerimise liides
CRUD	<i>Create Read Update and Delete</i> , andmebaasi objektide baas toimingud
CPU	<i>Central Processing Unit</i> , protsessor
ECC	<i>Error Access Code</i> , vea parandus kood
EF Core	Entity Framework Core, Microsoft ORM Dotnet raamistikus kasutamiseks
GPU	<i>Graphics Processing Unit</i> , videokaart
JSON	<i>Javascript Object Notation</i> , avatud faili vormingu ning andmevahetus standard
LED	<i>Light emitting diode</i> , valgust kiirgav diod
ORM	Object Relational Mapper, relatsiooniline kaardistaja
PCIe	Peripheral Component Interconnect express, laia kasutusega arvuti laiendamise pesa standard
SATA	Serial AT Attachment, arvuti liides ühissiini ühendamiseks massmäluseadmetega
SQL	Structure Query Language, andmebaasi päringukeel

Sisukord

1	Sissejuhatus	8
2	Valmis arvutid versus enda arvuti ehitamine	9
2.1	Ise komplekteerimise eelised	9
2.1.1	Parem kontroll hinna üle	9
2.1.2	Parem kvaliteet ja täiustamisvõimalused	10
2.1.3	Oma disain	10
2.1.4	Tarkvara puhtus	10
2.2	Arvuti osade valikuga seotud raskused	11
2.2.1	Ühilduvus	11
2.2.2	Koormusele vastavus	12
3	Vanade arvutite täiendamine	13
4	Olemasolevad võimalused	14
4.1	PC Building Simulator	14
4.2	PCPartPicker	14
5	Rakenduse projekteerimine	14
5.1	Kihiline arhitektuur	16
5.2	Kasutajahaldus	16
5.3	Rakenduste majutus ja organiseerimine	17
6	Kasutatav tehnoloogia	17
6.1	Andmebaas – PostGreSQL	18
6.2	Tagarakendus - ASP.net REST rakendusliides	19
6.3	Eesrakendus - Vue.js	19
7	Veebiteenuse prototüüp	20
7.1	Rakenduse dokumentatsioon	20
7.1.1	Andmebaas	21
7.1.2	Rakendusliides	21
7.1.3	Eesrakendus	22
7.2	Edasised arendamisvõimalused	25
8	Kokkuvõte	26
	Kasutatud kirjandus	27

Jooniste loetelu

Joonis 1 - Andmebaasi olemi-suhte diagramm	21
Joonis 2 - Veebilehe põhivaade	23
Joonis 3 - "Minu arvutid"-vaade.....	24
Joonis 4 - "Avalikud arvutid"-vaade	24
Joonis 5 - Administratiivne vaade	25

1 Sissejuhatus

Käesoleva bakalaureusetöö eesmärgiks on luua veebirakendus, mis lihtsustaks ise arvuti kokkupaneku protsessi, täpsemalt arvuti täpsemalt parimate võimalike osade valikut ja ühilduvuse kontrolli. Ise sellise informatsiooni otsimine võib olla tülikas ning aeganõudev, kuna on olemas palju erinevaid komponente, mis omakorda sobivad ainult teatud süsteemidesse. Sellel põhjusel otsustas autor luua eelnevast probleemist ning oma huvist lähtudes rakenduse, mis võimaldaks arvuti komplekteerimise protsessi lihtsustada. Samuti pakub projekteeritav rakendus lisafunktsionaalsusi, näiteks hinna ja teiste filtrite järgi automaatset arvutiosade valimist või komponentide olemasolu puhul mõnes Eesti e-poes, viide otse poe toote lehele.

2 Valmis arvutid versus enda arvuti ehitamine

Käesolevas peatükis toob autor välja ise arvuti ehitamise erinevused võrreldes valmis arvuti ostmisega.

2.1 Ise komplekteerimise eelised

Ise arvuti komplekteerimisel on mitmeid funktsionaalseid ja esteetilisi eeliseid. Selles peatükis loetleb autor nende peamisi ning toob välja nende kasulikkuse arvuti komplekteerijale.

2.1.1 Parem kontroll hinna üle

Ise arvuti komplekteerimisel saab ise valida, mis osade peale läheb suurem osa rahast ja kus tuleks selle arvelt vähem kulutada.

Juhul kui komplekteeritav süsteem läheb raskesti ligipääsetavasse või muul moel mitte eriti nähtavasse kohta, siis ei ole eriti hea mõte keskenduda ilusale valgustusele, mida keegi nii kui nii näha ei saa. Selle asemel tuleks jälgida süsteemi õhutust, et kinnises ruumis oleva süsteemi sisemus liiga kuumaks ei läheks, mis võib tekitada jõudluse kadu.

Olenevalt tulevase arvuti kasutusala tasub mõelda selle peale, mis osad süsteemist hakkavad kõige rohkem väärtust tootma. Kui tegu on mänguri arvutiga, siis kõige esimesel kohal peaks olema graafikakaart ning sinna tuleks suunata suur osa süsteemi eelarvest. Vastupidisel juhul, kui tegu on puhtalt tööarvutiga, mis ei nõua eriti suurt graafikajõudlust, peaks keskenduma protsessori võimsusele ja jahutusele. Kolmandal juhul, kui tegu on kontoriarvutiga, mille põhiliseks kasutusala on kontoritarkvara, tasub pigem jälgida erinevaid pikaajalist töövõimekust ja töökindlust pakkuvaid osi, näiteks kõrge kvaliteediga kõvakettaid ja vooluallikaid. Eelnimetatud on samuti tähtsad ka muudes süsteemides. (1 lk 23-24)

2.1.2 Parem kvaliteet ja täiustamisvõimalused

Arvuti komponente on turul palju, mõned neist on näiliselt väga sarnased, kuid erinevad spetsifikatsioonist tuleneva kvaliteedi poolest.

Parem kvaliteet ei tähendada ainult seda, et toode töötab kauem ja paremini, vaid suurendab ka toote tulevikuväärtust, muutes seda taaskasutatavaks ja uute toodetega ühilduvaks. Hea näide sellest on emaplaadid. Kõrgema kvaliteediga emaplaadid pakuvad tüüpiliselt märkimisväärselt rohkem erinevaid omadusi, näiteks suur hulk muutmälu pesi või laia protsessori toega kiibistik. Sama kehtib lisaks ka vooluallikate puhul, kus hea vooluallikas võib kesta mitmeid aastaid, enne kui tuleb välja vahetada.

2.1.3 Oma disain

Kuigi funktsionaalsete arvuti baasosade valik võib üldjuhul mitme arvuti puhul sarnaneda, on siiski võimalik luua süsteem, mis on tõeliselt unikaalse välimusega.

Tüüpiliselt alustatakse korpuse disainiga. Tüüpiliselt on arvuti korpuseks püstine või lamav risttahukas. Funktsionaalsuse poolest tagab see head paigaldamisvõimalused ning lihtsa õhuvoolu süsteemi. Selliseid korpuseid on palju, kuid samuti on olemas küllaldaselt ebatraditsioonilisi korpuseid. Näiteks korpus võib olla püramiidikujuline, kosmoselaeva kujuga või üldse võib arvuti paigutada otse oma töölaua sisse (2). Selle teemaga tegeleb rohkem *casemod* ehk korpuste ümberkujundamise kogukond, kes on loonud palju erinevaid ning huvitavaid arvuti korpuse disaine. (3)

Alati ei pea võtma ette ulmelist korpust, milles võib olla keeruline süsteemi kokku panna, on võimalik ka tüüpkorpust isikupärastada. Alustades komponentide toonide valimisega ja lõpetades ilusate LED-lampidega, mis illumineerivad klaasist külgedega korpust ilusa vaate jaoks, ei ole ilusa sisemuse saavutamiseks vaja väga keerulisi meetodeid kasutada. (2)

2.1.4 Tarkvara puhtus

Valmis arvutid, eelkõige need, mis töötavad Windowsi platvormil, saavad ostmisel tüüpiliselt kaasa eelpaigaldatud tarkvara, mis võivad olla mingil määral kasulikud, aga on enamjaolt ebavajalikud või isegi häirivad. Tüüpiliselt on nendeks programmideks erinevad limiteeritud litsentsiga anti-viirused või „arvuti puhastajad ja haldajad“. Nende programmide eemaldamine on üks esimesi asju, mida arvuti ostja teeb. (4)

Ise arvuti ehitamisel saab täpselt ise ära valida, mis tarkvara süsteemis on, alates operatsioonisüsteemist kuni veebibrauserini.

2.2 Arvuti osade valikuga seotud raskused

Arvuti komplekteerimine algab osade valikust, kuid millised valida ja kuidas neid üksteisest eristada? Iga komponendil on oma spetsifikatsioon, mis määrab ära tema sobivuse ehitatavasse süsteemi. Samuti tuleb võtta arvesse süsteemi tulevast kasutusala, et paremini selgitada välja vaja minevad komponendid.

2.2.1 Ühilduvus

Selleks, et arvuti komponendid optimaalselt või üldse töötaksid, peavad nad olema üksteisega ühilduvad. Halb ühilduvus süsteemi osade vahel võib heal juhul viia kerge jõudluse või mõne lisafunktsionaalsuse kadumiseni, halval juhul aga võib süsteem mitte käivituda või rikneda. Seega oleks arvutiosade omavaheline ühilduvus kõige tähtsam asi, mida komplekteerimisel tasub jälgida.

Näiteks käesoleval ajal toetavad suur osa uuemaid graafikakaarte PCIe-4 standardit. Kuna see standard kasutab sama pesa kui eelmise generatsiooni PCIe-3, on need kaardid igati ühilduvad vanemate arvutitega, kuid kasutades vanemat tehnoloogiat, töötavad nad oma võimalikust kiirusest aeglasemalt. (5)

Emaplaadi põhifunktsionaalsuse määrab tüüpiliselt ära tema kiibistik(ing. keeles chipset). Sellest oleneb nii emaplaadiga sobilike protsessorite nimekiri kui ka kasutatavate muutmälu moodulite spetsifikatsioon. (6 lk 192-194)

Töö kirjutamise ajal on kõige enam levinud muutmälu tüüp lauarvutite puhul DDR4. Uus DDR5 standard ei ole veel tavakasutusse läinud ja DDR3 standardit kasutavad väga vähesed arvutid. Sellise turu seisu tõttu on hetkel DDR3 oma madala nõudluse tõttu isegi kallim kui DDR4 mälu, olles samal ajal halvema jõudlusega.

Muutmälu valimine oma arvutisse oleneb esiteks eeltoodud standarditest, mis on kasutusel arvuti emaplaadil. Kui tegu on üsna vana arvutiga, siis on sinna sobilik DDR3 mälu. Uue süsteemi ehitamise puhul on praegusel hetkel parim valik DDR4. Olukord võib muutuda, kuna DDR5 standard ei veel küll tavakasutajateni jõudnud, samas on see juba välja töötamisel ja selle kohta on olemas palju informatsiooni, mis viitab suurele jõudluse

kasvule ning paljudele muudele lisafunktsionaalsustele, mida varasemates generatsioonides ei olnud, näiteks sisseehitatud ECC(Error Correction Code). (7)

Lisaks eelmainitule tasub muutmälu valimisel jälgida ka mälu kiirust ning ajastust. Üldjuhul ei ole see ühilduvuse suhtes väga tähtis, kuna emaplaadid toetavad suurel hulgal erinevaid kiiruseid ning vajadusel saab liiga kiire mälu puhul selle kiirust vähendada. See tähendaks võimaliku jõudluse kaotust, mida tasuks vältida mälu soetamisel.

Arvuti komponendid on voolu kasutuse suhtes muutunud märkimisväärselt efektiivsemaks, kui varem. Efektiivsem voolukasutus tähendab väiksemat soojuste tekkimist. Vaatamata sellele on olemas teatud komponendid, mis kasutavad selle lisa efektiivsuse ära, et suurendada veel enam komponendi jõudlust, selle pärast peaks igal süsteemil olema korralik jahutus, mis komponentide poolt õhatava kuumuse süsteemist välja.

Kõige lihtsam ja kasutatavam meetod selle saavutamiseks on õhkjahutus ventilaatoritega. Ventilaatorid puhuvad kuuma õhu välja ja toovad külma õhu sisse. On olemas palju erinevaid konfiguratsioone, kuidas õhkjahutus üles sättida, millest enamus olenevad süsteemis kasutusel olevast korpusest. Osad korpused võimaldavad õhuvoolu nii ülevalt-alla, kui eest-taha, kuid mõnel on olemas näiteks ainult tagant õhutamise võimalus ning õhu sisse tõmme toimub passiivselt või läbi külgevade. (8)

Tänapäeval on lauarvutitele võimalik lisaks traditsioonilistele komponentidele lisada juurde ka erinevaid lisafunktsionaalsusi pakkuvaid mooduleid. Nendeks võivad olla näiteks lisa USB-pistikuid sisaldavad kaardid või RAID kontrollid. Peamiseks arvutisiseseks laiendamiseks kasutatakse emaplaadil olevat PCIe-pesa, ehk siis võimalike lisaseadmete kogus oleneb peamiselt nende kogusest.

2.2.2 Koormusele vastavus

Arvutiosade laia valiku tõttu võib olla keeruline aru saada, milline kindel komponent vastavalt ehitatava süsteemi tulevase suunitlusele sobilik on.

Kodu arvuti on erinevate võimekuste keskte, üritades olla piisavalt hea igasuguste kodutoimingute jaoks, kuid samal ajal olles ka taskukohane. Selliste arvutite puhul tuleks peamiselt keskenduda süsteemide lihtsusele. Ei ole vajalik seadistada vesijahutust, kuna sellist tüüpi arvutis olevad muud komponendid ei tooda nii palju kuumust.

Mänguri arvuti keskendub enamasti 3D graafilisele edukusele. Sellise arvuti puhul on esimesel kohal graafikakaart, mis määrab põhiliselt ära arvuti võimekuse 3D programmides. Samuti tasub arvesse võtta ka protsessorit, et see ei oleks liiga nõrk ega tekitaks süsteemis pudelikaela.

Tööarvuti keskendub produktiivsusele ja töökindlusele. Sellise arvuti puhul oleneb graafikakaardi ja protsessori valik nõutavast töö spetsiifikast, ülejäänud osad peaksid tuleviku täiendamise ja jätkusuutlikkuse jaoks olema kõrge kvaliteediga.

3 Vanade arvutite täiendamine

Tänapäeval ei toimu võrreldes minevikuga enam igal aastal murrangulisi muutusi riistvara võimsuse suurenemisel. Hea arvuti võib kesta mitu aastat, enne kui ta muutub kasutamatuks. Mis aga pidevalt muutub, on tarkvara maailm, kus pidevalt tulevad juurde uued programmid või vanade uuendused, mille tõttu võib pidevalt tõusta nõutav süsteemi jõudlustase. Seega on isegi head arvutit vaja ühel või teisel hetkel riistvaraliselt uuendada, selleks et kasutada uuemaid funktsionaalsusi.

Tänu tänapäevaste arvutikomponentide suurele ühilduvusnimekirjale on võimalik täiesti uue arvuti ostmise asemel täiendada vana arvutit ja muuta see taas mugavalt kasutatavaks.
(9)

Esimene komponent, mille ühilduvus ei olene eriti teistest ja mida kõige tihedamini ka välja vahetatakse, on graafikakaart. Olles osaks nii mängurite arvutis kui ka krüptoraha kaevanduses näeb graafikakaarte süsteemides pidevalt vahetumas.

Muutmälu puhul oleneb süsteemi täiendatavus emaplaadil olevatest mälupesade kogusest ja toetatavatest kiirustest. Vabade kohtade olemasolul ja vastava ühilduva mälu leidmisel saab muutmälu kogust lihtsasti suurendada. Kõvaketaste lisamisel on tähtsal kohal emaplaadi SATA pesade kogus ja arvuti toiteallika SATA ühenduste kogus. Uue kõvaketta ühendamisel läheb vaja neist mõlemat.

4 Olemasolevad võimalused

4.1 PC Building Simulator

PC Building Simulator on videomäng, mis vastavalt oma nimele simuleerib arvuti ehitamist. Selles mängus saab diagnoosida, remontida ning komplekteerida oma arvuteid. Kasutusel olevad komponendid on täpsed koopiad päriselust ning seega on mängus kokkupandavad süsteemid ka reaalselt võimalik komplekteerida. Samuti täiendatakse komponentide valikut pidevate uuendustega. Käesolev mäng on heaks tutvustuseks inimestele, kes soovivad ise arvutit komplekteerima õppida. (10)

4.2 PCPartPicker

PCPartPicker on kõige kasutatum arvuti komplekteerimise veebirakendus. Samuti oli see ka suureks inspiratsiooniks käesoleva töö loomisel.

Sellel leheküljel saab komplekteerida oma arvuti suurest valikust komponentidest, omavahel jagada komplekteeritud arvuteid ning vaadata teiste omi. (11)

5 Rakenduse projekteerimine

Käesolevas töös vaadeldavate probleemide lahendamiseks otsustas autor kavandada veebirakenduse. Probleemi lahendamiseks valis autor veebirakenduse, kuna see on kasutaja jaoks lihtsasti ligipääsetav ning ei nõua kasutamiseks rohkemat kui brauserit. Süsteem peab vastama järgnevatele tingimustele:

- Kihiline arhitektuur
- Kasutajahaldus
- Hoitavate andmete nõuded

Tagarakenduse funktsionaalsused:

- Kõikide andmebaasi objektide CRUD – administratiivne funktsionaalsus
- Komponentide võrdlus üldhinnangu või spetsifikatsiooni järgi
- Parema komponendi otsing identifikaatori järgi
- Komponentide ühilduvuse kontroll
- Kindlad komponentide tüübid
- Andmete olemasolu käivitamisel – testandmed
- Arvuti osade kokku panek – arvuti loomine
- Üldine hinnang / kategoriseerimine komponentidele
- Oma arvuti ehitamine “mall” arvutist – koopia tegemine
- Olemasoleva arvuti otsing mudeli järgi
- Andmete importimine
- Päringud JSON vormingus
- Swagger API tugi – lihtsam tagarakenduse dokumentatsioon
- Sessiooni haldus läbi JSON Web Token-i
- Avalikud mall arvutid

Eesrakenduse funktsionaalsused:

- Kõikide andmebaasi objektide CRUD lehed – administratiivne paneel
- Komponentide võrdluse leht
- Komponentide otsing/filtreerimine – mudel, identifikaator, tüüp, spetsifikatsioon
- Arvuti kokkupaneku leht
- Mallide leht
- Teiste poolt kokku pandud avalikud arvutid
- Kasutajahaldus – sisselogimine, väljalogimine, andmete haldus
- Automaatne arvuti komplekteerimine leht:
 - Testide tulemuste / üldhinnangu / kategooria järgi ehitamine
 - Hinna järgi ehitamine
 - Suuna valimine - Mänguri / Kodu / Töö

5.1 Kihiline arhitektuur

Veebiteenus jaguneb kolmeks kihiks:

- Andmebaas
- Tagarakendus
- Eesrakendus

Andmebaasis hoitakse enamikku püsiandmeid, näiteks toodete andmed või kasutajate andmed. Sellele peaks ligi pääsema ainult läbi tagarakenduse või erandjuhul administratiivsete õigustega isik otse.

Tagarakendus tegeleb andmebaasist andmete küsimisega, nende sisestamisega ning eesrakenduse jaoks serveritavate andmete arvutamisega. Tagarakendus võib töötada ka ainult andmebaasiga, olles API rollis ka teiste seda nõudvate teenuste jaoks.

Eesrakendus on peamine tavakasutaja poolt kasutatav vaade. Selle kaudu pääseb ligi tagarakenduse funktsionaalsustele ning samuti hõlbustab ta ka andmebaasiga seotud administratiivseid tegevusi, pakkudes selle jaoks mugava kasutajaliidese.

5.2 Kasutajahaldus

Veebiteenus saab olema kasutajapõhine. Kasutajad saavad seostada teatud andmeid ainult enda kasutaja kontoga. Vaatamata sellele peaks süsteemis olema võimalik tegutseda ka anonüümse kasutajana, kuid sellisel juhul ei salvestata kasutaja tegevusi.

Vastavalt seadustele peab kasutaja saama näha ja vajadusel kustutada temaga seonduvaid andmeid veebiteenusest. Selle jaoks tuleb luua kasutajahaldussüsteem, mis liidetakse läbivalt olemasolevatesse kihtidesse. Andmebaasis hoitakse kasutajatunnus, parooli räsi ja teised kasutaja poolt loodavad andmed. Tagarakendus verifitseerib kasutaja ning kontrollib ligipääsuõigusi. Eesrakenduses saab luua oma konto ning edastada andmeid teistesse kihtidesse, samuti peab eesrakenduses olema selge andmete või kogu kasutajakonto kustutamise võimalus.

Peamisteks hoitavateks andmeteks on:

- Komponentide kasutamised komplekteerimisel
- Poodidega seotud andmed – komponentide hinnad, saadavus

5.3 Rakenduste majutus ja organiseerimine

Loodaval süsteemil on kihiline ülesehitus ning seega peab olema võimalus igal kihil üksteisega suhelda. Kui kõik rakendused ühe arvuti peal töötavad, ei ole see probleemiks, kuid selle jaoks, et kogu tarkvara ka erinevatel aadressitel omavahel korralikult suhelda saaksid, tuleks nendele soetada veebimajutus.

Selleks, et rakendused omavahel ka üle veebi suhelda saaksid, peab igaüks neist asuma kindlal aadressil. On olemas palju erinevaid kohti, kuhu saaks oma rakendust majutada.

Prototüübi ajutiseks majutamiseks kasutab autor Microsoft Azure'i veebimajutusteenust. Samuti on rakendused mugava majutuse jaoks pakitud Dockeri tarkvarapaketti, mis võimaldab rakendusi lihtsamalt käivitada ja hallata.

Docker on ühendatud versioonihaldusega, nii et muudatuste tekkimisel kompileerib Dockeri pakett ennast uuesti ning veebiteenus uuendab ennast automaatselt. Tingimusteks saab lisada näiteks testide läbimist.

6 Kasutatav tehnoloogia

Projekteeritava rakenduse loomiseks on võimalik kasutada palju erinevaid tööriistu ning raamistikke. Siin peatükis toob autor välja oma rakenduse loomisel kasutatavaid tehnoloogiaid ja raamistikke ning põhjendab nende kasutust.

6.1 Andmebaas – PostGreSQL

Tagarakenduses kasutatakse andmebaasiga seotud tegevusteks *Entity Framework Core* ORM-i.

ORM ehk *Object Relational Mapper* on abitööriist, mis laseb kogu andmebaasi suhtluse kirjutada enda soovitud programmeerimiskeeles ning teeb selle andmebaasi saates ise SQL-iks ümber. See on hea viis teha päringuid andmebaasi juhul kui ise nende käsitsi kirjutamine on liiga tülikas, aeganõudev või puudub vajalik kogemus. Lisaks pakuvad mõned ORM-id ka lisafunktsionaalsusi, näiteks migratsioonide automaatne tegemine, mis kiirendavad arendamisprotsessi. (12)

Peamiseks negatiivseks küljeks on ORM-i kasutusel selle keeruline paigaldus ning arendaja jaoks abstrahereeritud funktsionaalsus, ehk juhul kui arendaja ei ole hea SQL-i päringute kogemusega, on raske ette näha, kuidas ORM täpselt töötab. (12)

Entity Framework Core on Microsoft-i poolt loodud ORM, mis on ühilduv paljude erinevate andmebaasidega. (13)

Selles töös sai kasutusele võtta EF Core, kuna see teeb arendusprotsessi lihtsamaks ning autori varasema kokkupuute tõttu SQL-i kirjutamisega ei ole probleemi rakenduse funktsionaalsustest aru saamisega. Samuti oli kasulik selle ORM-i mitme erineva andmebaasi tugi.

Käesolevas töös on andmete hoidmiseks kasutusel PostGreSQL andmebaas. Samuti on lisatud tugi lokaalse SQLite andmebaasi kasutamiseks, milleks tuleb vastavalt juhistele muuta rakenduse konfiguratsiooni.

Töö projekteerimisel olid andmebaasi valimisel valikus PostGreSQL ja MySQL, mis on mõlemad töö kirjutamise ajal kõige enam levinud andmebaasid.

Algseks plaaniks oli kasutada töö loomisel MySQL andmebaasi. See on Oracle Corporation-i poolt arendatav populaarne andmebaasi süsteem, mida kasutavad lisaks teistele ka suuremad firmad näiteks Google, Facebook ja Adobe. (14)

Sarnaselt MySQL andmebaasile on PostgreSQL tasuta ja avatud lähtekoodiga.

PostgreSQL-i peamisteks eelisteks, mille tõttu autor otsustas just seda oma töös kasutada, on suurem tugi erinevatele funktsionaalsustele, mis võivad tulla kasuks tulevases arendamises. Samuti tegi PostgreSQL-i alane varasem praktiline kogemus selle andmebaasi kasutusele võtu veebirakenduses palju lihtsamaks.

SQLite-i eristab teistest andmebaasides peamiselt selle väike maht ning portatiivsus, kuna kogu andmebaas hoitakse ühes failis.

SQLite andmebaasi tugi on lisatud peamiselt lihtsaks kohalikus seadmes funktsionaalsuse testimiseks, suurem osa testitavatest osadest töötavad ka selle andmebaasiga, ning tänu EF Core-i abstraktsioonikihile ei pea andmebaaside vahetamisel rakenduse koodi muutma.

Kuni andmebaas on EF Core-i poolt toetatav, on tulevases arenduses võimalik andmebaasi vahetada.

6.2 Tagarakendus - ASP.net REST rakendusliides

Loodava veebiteenuse tagarakendus on kirjutatud C# programmeerimiskeeles ning kasutades .Net Core ja ASP.net raamistikke. Andmebaasiga suhtlemiseks on kasutusel Entity Framework Core.

C# on Microsofti poolt toodetud tugevalt tüübitud, objektorienteeritud tarkvaraarenduse keel. Selle programmeerimiskeele valik põhines peamiselt autori varasemal kogemusel C# keele kasutuses tagarakenduste kirjutamisel ning korraliku ja natiivselt toetatud ORM-i olemasolu antud platvormil.

6.3 Eesrakendus - Vue.js

Javascript on projekti kirjutamise hetkel kõige populaarsem programmeerimiskeel veebirakenduste loomiseks. (15)

Selle programmeerimiskeele valis autor lihtsa paigaldamise, hea õpitavuse ning võimekuse poolest. Samuti saab kõnealus keeles kasutada palju erinevaid laiendusi ja lisapakette, mis kergendavad arendamisprotsessi.

Selles projektis on veebilehe loomisel kasutusel Vue.js raamistik. Vue.js on Typescript-i toetav tarkvaralahendus modulaarsete ja skaleeritavate üheleheliste rakenduste loomiseks. Vue.js on lihtsasti paigaldatav ning ei nõua suurel määral konfigureerimist hästi töötamiseks.

Üheks Vue.js eripäradest on veebilehe komponentide hoidmine mitte eraldi mitmes failis, üks fail koodi jaoks, teine fail kujunduse jaoks jne, vaid ühes kompaktses .vue failis.

Vue.js on populaarsuse kasutamispopulaarsuse poolest hetkel esikolmikus, olles üks vanematest raamistikest, mis siiaaani tõusva populaarsusega on. (16)

7 Veebiteenuse prototüüp

Töös käsitletava probleemi lahenduse esitamiseks tegi autor eeltoodud funktsionaalsustest lähtuvalt veebirakenduse.

7.1 Rakenduse dokumentatsioon

Rakenduse prototüüp on arendamise protsessi ajal majutatud järgmistele veebilehekülgedele:

- Tagarakendus: <https://tparts-pcbuilder-back.azurewebsites.net>
- Eesrakendus: <https://tparts-pcbuilder-front.azurewebsites.net/>

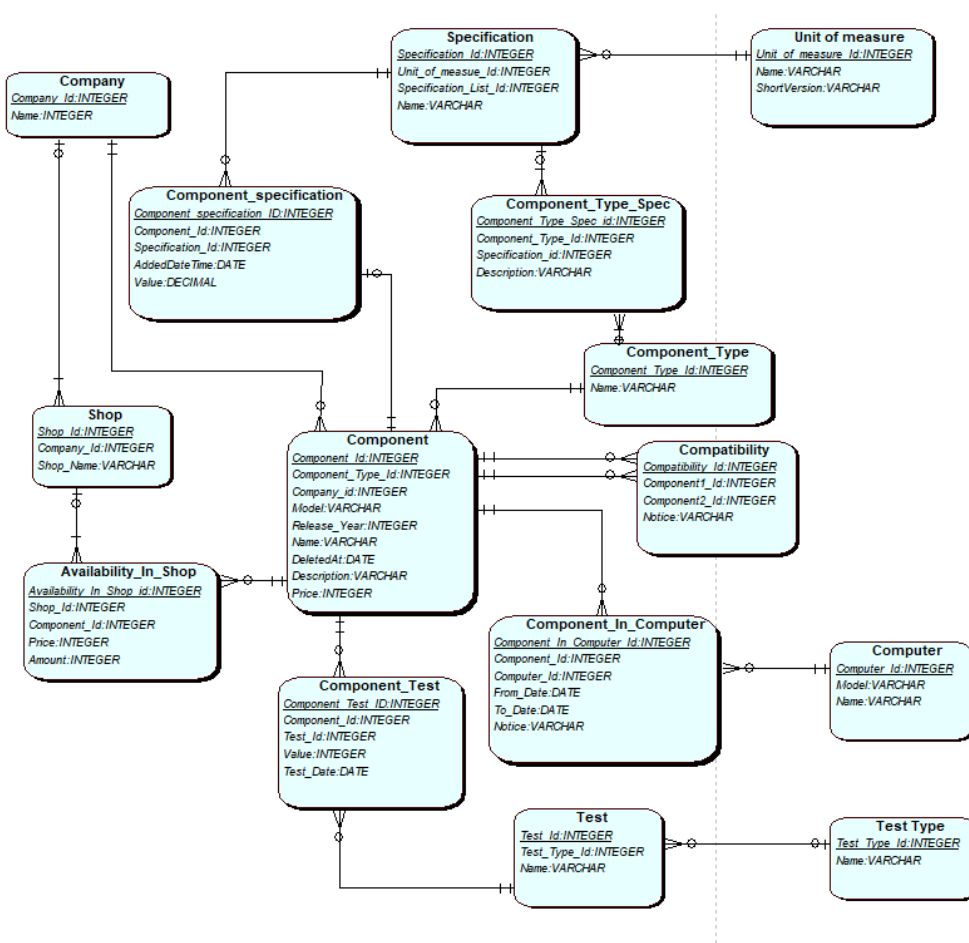
Rakenduse lähtekood on leitav Tallinna Tehnikaülikooli Gitlab-i järgmistelt lehekülgedelt:

- Tagarakendus: <https://gitlab.cs.ttu.ee/tparts/diplom-back>
- Eesrakendus: <https://gitlab.cs.ttu.ee/tparts/diplom-front>

7.1.1 Andmebaas

Andmebaasis on põhikomponentideks arvutiosad(Joonis 1 *Component*) ja arvutid(Joonis 1 *Computer*). Ülejäänud osad on staatilised või vähemuudetavad andmekogumid, näiteks komponendi tüübid(Joonis 1 *Component Type*), ja mitu-mitmele ühendavad objektid, näiteks komponent arvutid(Joonis 1 *Component In Computer*).

Komponentide vahelise ühilduvuse jaoks on loodud eraldi objekt(Joonis 1 *Compatibility*), kus hoitakse ühilduvate objektide paare, kuid samuti võib ühilduvuse kontrolliks kasutada komponentide spetsifikatsioone(Joonis 1 *Component Specification*).



Joonis 1 - Andmebaasi olemi-suhte diagramm

7.1.2 Rakendusliides

Projekti rakenduseliidese dokumenteerimiseks kasutatakse Swagger-it, mis on leitav järgmiselt aadressilt:

<https://tparts-pcbuilder-back.azurewebsites.net/swagger>

Swagger on professionaalne tööriistakomplekt, mis laseb automaatselt genereerida ning ülal hoida rakendusliideste dokumentatsiooni. (17)

Rakendusliideses toimub komponentide ühilduvuse kontroll kahel viisil:

- Andmebaasi ühilduvuse tabeli kaudu
- Spetsifikatsiooni põhiselt

Esimene valik on primaarselt kasutatav meetod. Andmebaasi tehakse päring, kas on olemas väli ühilduvuste tabelis, kus on olemas antud komponentide identifikaatorid.

Teine valik on keerulisem ning on seega varuvariandiks, juhul kui esimene tulemust ei anna. Igal komponendi tüübil on ette nähtud spetsifikatsioonid, mis on tähtsad ühilduvuse kontrollis. Juhul, kui kõik vajalikud spetsifikatsioonid on kahe komponendi vahel täidetud, näiteks protsessori ja emaplaadi puhul oleks nendeks protsessori pesa tugi ja emaplaadi kiibistik, on komponendid ühilduvad ning luuakse ka vastav rida tabelisse ühilduvused. Selle meetodi paljude variatsioonide tõttu on antud töö prototüübis kasutatud peamiselt esimest meetodit. Ühilduvusi saab ka eesrakenduses käsitsi lisada.

Rakendusliides võtab vastu informatsiooni JSON-i kujul ning konverteerib selle endale sobilikeks objektideks. Samuti saadab rakendusliides enda informatsiooni klientidele JSON vormingus.

Tagarakenduses on olemas ka andmete importimise võimalus, mida kasutatakse hetkel hea informatsiooni allika olemasolu puudumise tõttu vaid staatiliste ning käsitsi kirjutatud testandmete importimiseks.

7.1.3 Eesrakendus

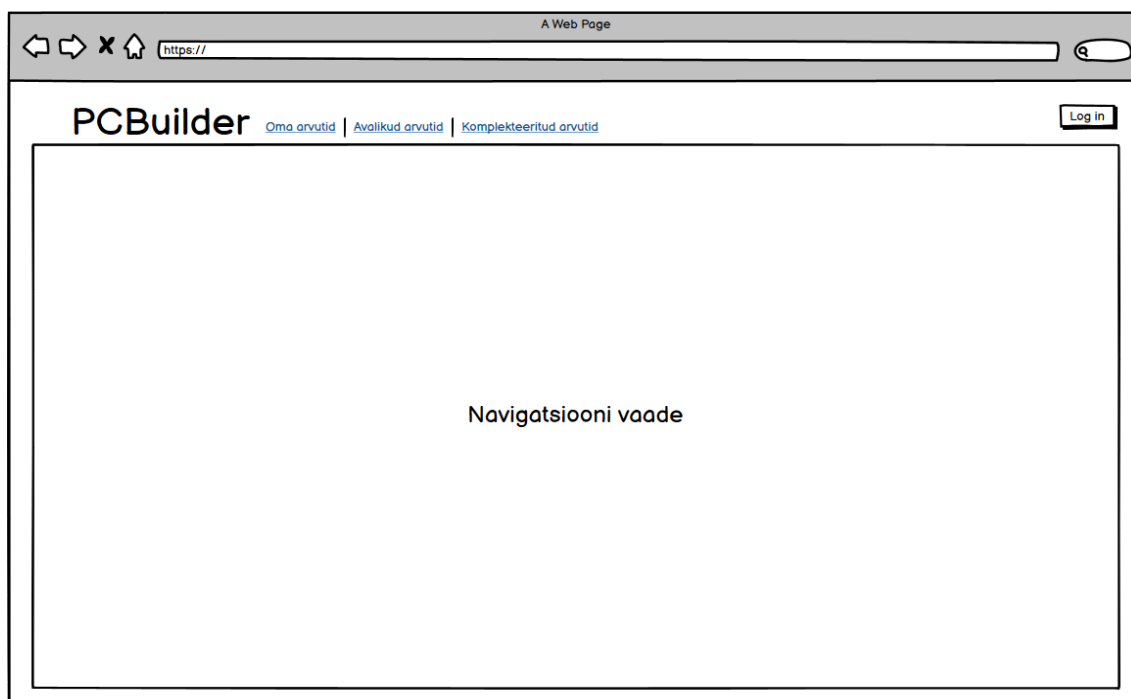
Kasutades Vue.js raamistikku, on loodav veebiteenus üheleheline, ehk tegu on ühe modulaarse põhilehena, mille komponendid muutuvad vastavalt kasutaja soovile. Enamus navigeerimist toimub läbi põhinavigatsiooniakna, mis asub lehekülje ülaosas. Samuti pääseb sealt kaudu ligi ka kasutajahalduse vaatele.

Põhilisteks mooduliteks veebilehel on:

- Oma arvutite vaade/nimekiri
- Kindla arvuti vaade/ehitamise vaade
- Arvuti automaatse komplekteerimise konfigureerimise vaade
- Administratiivne paneel andmete halduseks
- Komponentide nimekiri
- Komponenti vaade
- Kasutaja vaade

Eelnimetatutest on põhinavigatsioonis arvutite nimekiri, automaatne arvuti komplekteerimine, kasutaja vaade, administratiivne paneel. Teiste moodulite juurde navigeerimine toimub läbi eelnevate.

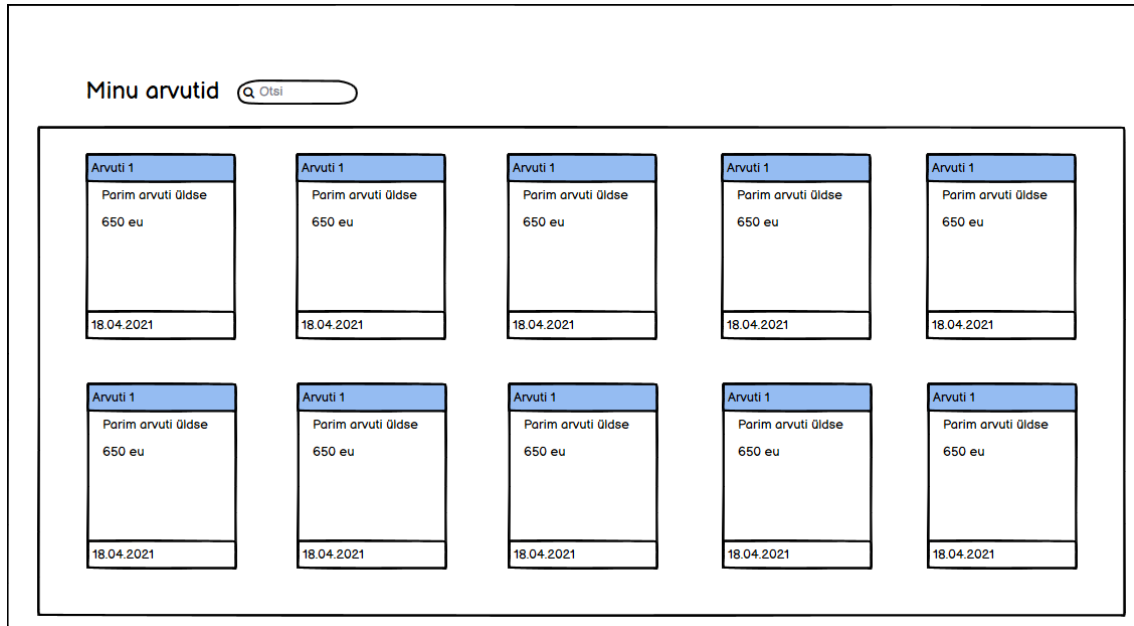
Põhivaade:



Joonis 2 - Veebilehe põhivaade

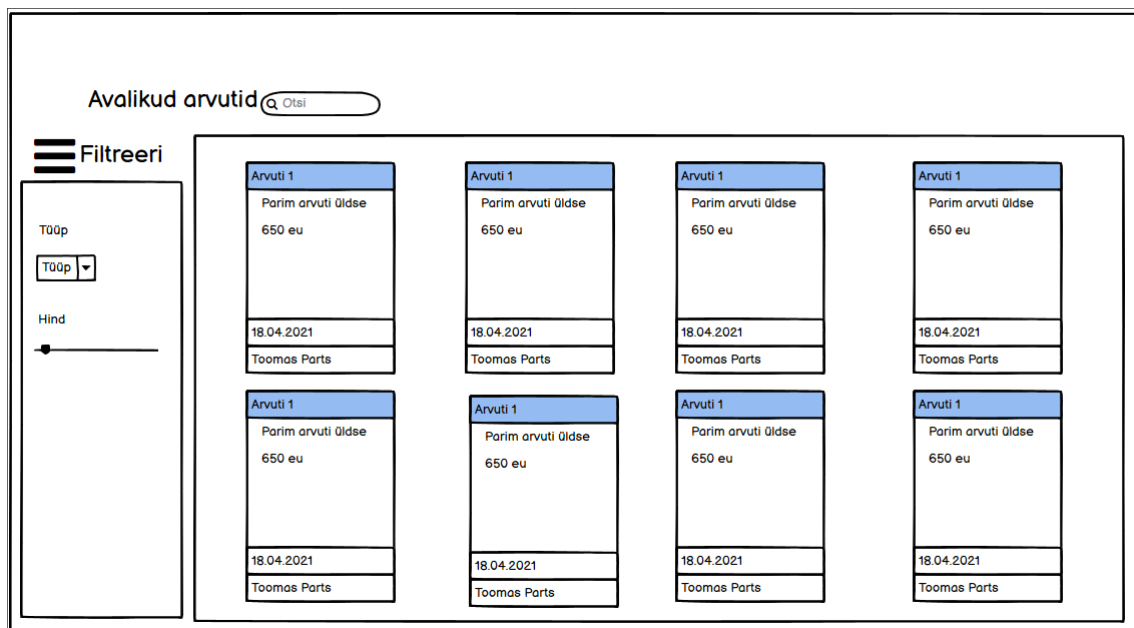
Rakenduse põhinavigatsiooni menüü on pidevalt nähtav ning navigatsiooni vaade oleneb hetkel valitud komponendist.

Oma arvutite nimekiri:



Joonis 3 - "Minu arvutid"-vaade

Avalikud arvutid:



Joonis 4 - "Avalikud arvutid"-vaade

Administratiivne vaade:

Komponendi nimi	Mudel	Valikud
Ryzen 5 3600	x000000	
Ryzen 5 3600	x000000	
Ryzen 5 3600	x000000	
Ryzen 5 3600	x000000	

Joonis 5 - Administratiivne vaade

7.2 Edasised arendamisvõimalused

Loodud rakenduse peamine puudus on rakenduste automaatse pärimise võimaluse puudumine, mis jäi hetke projekti skoobist välja liigse mahu ning keerukuse tõttu. Peamiseks informatsiooniks, mida antud rakendus pidevalt uuena nõuaks on andmed komponentide kohta ning nende saadavuse kohta poodides. Selle loomiseks on mitmeid erinevaid võimalusi.

Üheks võimaluseks on luua „veebikaabitsa“ rakendus, mis liigub mööda ettenähtud e-poodi ning „kraabib“ sealt vajaliku informatsiooni, mis edastatakse seejärel rakendusliidesesse andmebaasi sisestamiseks/uuendamiseks. Siinkohal ei tasu segamini ajada veebikaabitsat ja „veebiämblikut“. Nendest esimese eesmärgiks on leheküljelt informatsiooni korjamine, aga teist kasutatakse peamiselt informatsiooni indekseerimiseks parema otsitavuse jaoks, näiteks Google otsingumootor (18)

Sellist lahendust ei ole üldjuhul hea mõte kasutada ilma antud veebilehtede omanike nõusolekuta, kuna vastasel juhul võivad nad blokeerida veebikaabitsa ligipääsu e-poele. Samuti on selline tegevus ilma eelnevalt luba küsimata juriidiliselt vaieldava seaduslikkusega. (19)

8 Kokkuvõte

Käesoleva töö eesmärgiks oli selgitada arvutite ise komplekteerimise võimalusi ja luua rakendus, mis aitaks lihtsustada arvuti komplekteerimise protsessi. Samuti on siin uuritud ka arvuti komplekteerimisel komponentide valikuga seotud raskusi ning analüüsitud vanade arvutite täiustamisvõimalusi.

Rakenduse arendamisprotsessi käigus kirjeldas autor loodava rakenduse nõudeid ning esitas ka valitud programmeerimiskeeled ning raamistikud. Rakenduse sai jagada kolme kihti: PostgreSQL-il põhinev andmebaas, ASP.Net tagarakendus ning Vue.js eesrakendus.

Arendamise käigus loodud prototüüp vastab ainult väikesele osale projekteeritavast rakendusest. Peamisteks puudusteks on automaatse andmete pärimise ning arvuti komplekteerimise võimaluse puudumine. Need puudused tekkisid peamiselt autori vähesest kogemusest ORM-iga töötamisel ning sellest tekkinud suurte ajakulude tõttu. Lisaks oleks rakenduse olemasolevale funktsionaalsusele vaja rakendada automaattestid, mis edasise arenduse kindlamaks teeksid.

Kasutatud kirjandus

1. **Chambers, Mark L.** *Build Your Own PC Do-It-Yourself For Dummies*. s.l. : Wiley Publishing, Inc., 2009.
2. **Ross, Kaelum.** The 15 Most Unique PC Cases You Can Actually Buy in 2021. *What in tech*. [Võrgumaterjal] WHAT IN TECH, 10. Jaanuar 2021. a. [Tsiteeritud: 26. aprill 2021. a.] <https://whatintech.com/unique-pc-case/>.
3. **Murphy, David.** 15 crazy PC case mods. *PC World*. [Võrgumaterjal] PC World. [Tsiteeritud: 28. aprill 2021. a.] <https://www.pcworld.idg.com.au/slideshow/351980/15-crazy-pc-case-mods/>.
4. **Griffith, Eric.** PCMag. *How to Rid a New PC of Crapware*. [Võrgumaterjal] PCMag, 1. Aprill 2020. a. [Tsiteeritud: 26. aprill 2021. a.] <https://www.pcmag.com/how-to/how-to-rid-a-new-pc-of-crapware>.
5. **George, William.** PCI-Express 4.0 vs 3.0 Video Card Performance. *Puget systems*. [Võrgumaterjal] Puget systems, 30. november 2020. a. [Tsiteeritud: 26. aprill 2021. a.] <https://www.pugetsystems.com/labs/articles/PCI-Express-4-0-vs-3-0-Video-Card-Performance-1982/>.
6. **Mueller, Scott M.** *Upgrading And Repairing Pcs*. 22. s.l. : Pearson Education, 2015. 978-0-7897-5610-7.
7. **Smith, Ryan.** DDR5 Memory Specification Released: Setting the Stage for DDR5-6400 And Beyond. *Anandtech*. [Võrgumaterjal] Anandtech, 14. juuli 2020. a. [Tsiteeritud: 26. aprill 2021. a.] <https://www.anandtech.com/show/15912/ddr5-specification-released-setting-the-stage-for-ddr56400-and-beyond>.
8. **Intel Corporation.** PC Cooling: The Importance of Keeping Your PC Cool. *Intel*. [Võrgumaterjal] Intel. [Tsiteeritud: 26. aprill 2021. a.] <https://www.intel.co.uk/content/www/uk/en/gaming/resources/pc-cooling-the-importance-of-keeping-your-pc-cool.html>.
9. **James, Dave.** The easy PC upgrade guide: everything you need to know. *PCGamer*. [Võrgumaterjal] Future US, Inc, 17. detsember 2014. a. [Tsiteeritud: 26. aprill 2021. a.] <https://www.pcgamer.com/the-easy-pc-upgrade-guide/2/>.
10. **The Irregular Corporation Limited.** PC Building simulator. *PC Building simulator*. [Võrgumaterjal] The Irregular Corporation Limited. [Tsiteeritud: 6. mai 2021. a.] <https://www.pcbuildingsim.com/>.
11. **PCPartPicker.** PcPartPicker. *PcPartPicker*. [Võrgumaterjal] 2021. a. [Tsiteeritud: 26. aprill 2021. a.] <https://pcpartpicker.com/>.
12. **Hoyos, Mario.** What is an ORM and Why You Should Use it. *Bits and Pieces*. [Võrgumaterjal] Bits and Pieces, 24. detsember 2018. a. [Tsiteeritud: 26. aprill 2021. a.] <https://blog.bitsrc.io/what-is-an-orm-and-why-you-should-use-it-b2b6f75f5e2a>.
13. **Microsoft.** Entity Framework Core. *Microsoft docs*. [Võrgumaterjal] Microsoft, 20. september 2020. a. [Tsiteeritud: 26. aprill 2021. a.] <https://docs.microsoft.com/en-us/ef/core/>.
14. **Oracle Corporation.** MySQL documentations. *MySQL*. [Võrgumaterjal] 2021. a. [Tsiteeritud: 26. aprill 2021. a.] <https://dev.mysql.com/doc/>.

15. **inVerita.** inVerita - The Best Web Application Development Languages in 2021. *Medium*. [Võrgumaterjal] inVerita, 10. märts 2021. a. [Tsiteeritud: 26. aprill 2021. a.] <https://medium.com/@inverita/the-best-web-application-development-languages-in-2021-6b6eb5944925>.
16. **Existek.** Top Front-End Frameworks in 2021. *Existek blog*. [Võrgumaterjal] 14. jaanuar 2020. a. <https://existek.com/blog/top-front-end-frameworks-2021/>.
17. **SmartBear Software.** Swagger. *Api-documentation*. [Võrgumaterjal] [Tsiteeritud: 26. aprill 2021. a.] <https://swagger.io/solutions/api-documentation/>.
18. **Imperva.** Web Scraping. *Imperva*. [Võrgumaterjal] Imperva. [Tsiteeritud: 26. aprill 2021. a.] <https://www.imperva.com/learn/application-security/web-scraping-attack/>.
19. **Roberts, Edward.** Is Web Scraping Illegal? Depends on What the Meaning of the Word Is. *Imperva*. [Võrgumaterjal] Imperva, 17. september 2018. a. [Tsiteeritud: 26. aprill 2021. a.] <https://www.imperva.com/blog/is-web-scraping-illegal/>.

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks¹

Mina, Toomas Parts

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Arvuti komplekteerimise veebirakendus hobikasutajatele“, mille juhendaja on Kaido Kikkas
 - 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.

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 loomingulise 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.