

TALLINNA TEHNIKAÜLIKOOL  
Infotehnoloogia teaduskond

Holger Roosioja 184944IADB

**Nasdaq Tallinna börsil noteeritud ettevõtete  
aktsiaraamatute väljavõtteks vajaliku  
rakenduse arendamine**

Bakalaureusetöö

Juhendaja: Lembit Viilup  
Doktorikraad

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: Holger Roosioja

15.05.2023

## **Annotatsioon**

Bakalaureusetöö eesmärgiks on luua rakendus Nasdaq Tallinna börsil noteeritud ettevõtete aktsiaraamatute mugavaks väljavõtteks.

Töö esimeses osas toimub loodava rakenduse analüüs, vaadeldakse hetkeolukorda ning probleemi. Sellele järgnevas osas kaardistatakse funktsionaalsed ja mittefunktsionaalsed nõuded. Seejärel valitakse sobivaimad tehnoloogiad ja tööriistad ning luuakse töötav rakendus. Lõputöö viimases peatükis analüüsib autor käesoleva lõputöö tulemusena valminud rakendust ning toob välja võimalikud edasiarendused.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 28 leheküljel, 8 peatükki, 5 joonist.

## **Abstract**

# **Development of Application for Extracting Share Books of Companies Listed on the Nasdaq Tallinn Stock Exchange**

The aim of the bachelor's thesis is to develop an application for a convenient extract of the share books of companies listed on the Nasdaq Tallinn stock exchange.

In the first chapter of the thesis, the application is being analyzed, the current situation and the problem will be examined. The following chapter maps functional and non-functional requirements. Following that, the most suitable technologies and tools are selected and the working application is developed. In the final chapter of the thesis, the author analyses the application that was completed as a result of this thesis and points out possibilities for further development.

The thesis is in Estonian and contains 28 pages of text, 8 chapters, 5 figures.

## Lühendite ja mõistete sõnastik

.NET	Microsofti arendatud tarkvaraplatvorm.
Aktsia	Väärtpaber.
Aktsiaraamat	Nimekiri ettevõtte aktsionäridest.
Aktsionär	Füüsiline või juriidiline isik, kes omab ettevõtte aktsiaid.
API	<i>Application Programming Interface</i> , Rakendusliides.
Captcha	<i>Completely Automated Public Turing test to tell Computers and Humans Apart</i> , Väljakutse tuvastamiseks kas kasutaja on inimene.
Cron	UNIX-i käsk, ettemääratud tegumite perioodiliseks täitmiseks ettemääratud aja.
Emitent	Juriidiline isik, kes emiteerib aktsiaid.
EVKS	Väärtpaberite registri pidamise seadus.
HTML	<i>HyperText Markup Language</i> , Hüperteksti märgistuskeel.
IPO	<i>Initial Public Offering</i> , Aktsiate avalik esmapakkumine (aktsiaemissioon).
JSON	<i>JavaScript Object Notation</i> , JavaScriptil põhinev andmevahetusvorming.
LINQ	<i>Language Integrated Query</i> , .NET komponent andmebaasi päringuks.
Nasdaq	Kauplemise, börsitehnoloogia, kliiringu, noteerimise, teabe ja börsiettevõtetega seotud teenuste pakkuja.
Razor	Märgistuskeel veebilehe arenduseks.
SQL	<i>Structured Query Language</i> , Struktuurpäringukeel.
WebAssembly	Binaarne käsuvorming programmide käivitamiseks veebibrauseris.

# Sisukord

1. Sissejuhatus .....	9
1.1. Metoodika.....	9
2. Analüüs.....	11
2.1. Probleem.....	11
2.2. Tallinna börs.....	12
2.3. Olukord Tallinna börsil .....	13
2.4. Olukord maailmas .....	14
2.5. Andmekaitse .....	15
2.6. Riskid.....	16
3. Nõuded .....	17
3.1. Funktsionaalsed nõuded .....	17
3.2. Mittefunktsionaalsed nõuded.....	17
4. Arendusvahendid .....	18
4.1. Serveripoolsete tehnoloogiate valik.....	18
4.1.1. Python.....	18
4.1.2. TensorFlow.....	19
4.1.3. Selenium WebDriver.....	19
4.1.4. Apache Tika .....	19
4.2. Kliendipoolsete tehnoloogiate valik .....	20
4.2.1. Blazor .....	20
4.2.2. Entity Framework Core.....	20
4.3. Andmebaasi valik .....	21
4.4. Arenduskeskkonnad ja koodi hoidla .....	23
4.4.1. Arenduskeskkonna valik .....	23
4.4.2. Koodihaldustarkvara valik .....	24

5. Rakenduse arendus .....	25
5.1. Andmebaas .....	25
5.2. Serverrakenduse arendus .....	26
5.2.1. Struktuur.....	26
5.2.2. Captcha lahendus.....	28
5.2.3. Serverrakenduse haldus.....	30
5.3. Klientrakenduse arendus.....	30
5.3.1. Server või WebAssembly.....	30
5.3.2. Struktuur.....	31
5.3.3. Teek.....	32
5.3.4. Klientrakenduse haldus .....	33
5.4. Testimine .....	33
6. Tulemused .....	34
7. Edasiarendused .....	35
8. Kokkuvõte .....	36
Kasutatud kirjandus .....	37
Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks .....	41
Lisa 2 – Täielik olemi-suhte diagramm.....	42
Lisa 3 – Kuvatõmmis registripidaja lehelt aktsiaraamatu väljavõtteks .....	43
Lisa 4 – Kuvatõmmis registripidaja aktsiaraamatu PDF failist.....	44

## Jooniste loetelu

Joonis 1 Lihtsustatud olemi-suhte diagramm .....	25
Joonis 2 Serverrakenduse struktuur .....	27
Joonis 3 Skripti genereeritud turvakood .....	29
Joonis 4 Registripidaja genereeritud turvakood .....	29
Joonis 5 Klientrakenduse struktuur .....	31



# 1. Sissejuhatus

Keeruline majanduslik olukord maailmas ja laialdane investeerimis võimaluste kasv on hüppeliselt kasvatanud inimeste huvi investeerimise vastu.

Tallinna börsil on läinud viimastel aastatel erakordselt hästi, mida iseloomustab 2021. aasta börsiindeksi(OMXTGI) tõus 48.92 protsenti[1]. Kasvu põhjuseks võib pidada ka rahvasuus nimetatud IPOdroomi kui 2021. aastaga lisandus börsile 7 ja 2022 aastal 6 avalikku pakkumist. Börsijuhi Kaarel Otsa sõnul ületas 2022 aastal Eesti investorite arv 100 000 maagilise piiri. Täpsemalt on tegu Eesti elanike väärtpaperikontodega, kus on vähemalt üks Tallinna börsil kaubeldav väärtpaper[2][3]. Ettevõtte aktsiad jõuavad börsile läbi avaliku esmaemissiooni ehk IPO mille korral firma müüb aktsiaid avaliku pakkumisena. Aktsiabörsil jälgitakse aktsiate pakkumise ja nõudluse vahekorda, mis määrab aktsia hinna[4].

Viimaste aastate poliitiliste otsuste tagajärjel nagu II pensionisamba lammutamine, samba maksete ajutine peatamine ja pensioni investeerimiskonto loomine peab olema igal eestlasel võimalik saada osa Eesti majanduse edust läbi kohaliku börsi. Paraku on börsil tehtavate tehingute informatsioon täna võrdlemisi raskesti kättesaadav, mis eeldab börsiinfo kasutajatelt varasemaid kasutamiskogemusi ja kõrget professionaalsust.

Käesoleva bakalaureusetöö eesmärgiks on luua rakendus Tallinna aktsiabörsi aktsiaraamatutes toimunud muutuste operatiivseks ja mugavaks väljavõtteks. Töö käigus tutvustatakse seotud probleemi, tuuakse välja olemasolevad lahendused, kirjeldatakse töö metoodikat ning määratakse funktsionaalsed ja mittefunktsionaalsed nõuded. Samuti kirjeldatakse rakenduse loomiseks kasutatud tehnoloogiad ja põhjendatakse tehtud valikuid. Lõpetuseks tuuakse välja tehtud töö tulemused ja võimalikud edasiarendused.

Töö tulemustest on kasu eelkõige Tallinna börsi tänastele ja tulevastele aktsionäridele kes soovivad läbipaistvamat, detailsemat ja kiiremat ülevaadet aktsiaraamatutes toimunud muutustest, et seeläbi teha teadlikumaid investeerimise otsuseid.

## 1.1. Metoodika

Käesolev töö on jagatud kaheks. Töö analüüsi osas kirjeldatakse hetkeolukorda, lahendatavat probleemi, olemasoleva lahenduse tugevusi ja nõrkuseid ning pakutakse

omapoolne parem lahendus. Tuuakse selgelt välja rakendusele vajalikud nõuded mis on jaotatud vastavalt funktsionaalseteks ja mittefunktsionaalseteks nõueteks. Kirjeldatakse olukorda maailmas, andmekaitse aspekte ning võimalikke riske.

Töö teises osas valitakse sobivaimad tehnoloogiad rakenduse arendamiseks vastavalt seatud nõuetele ja tehnoloogiapinu oskustele. Kirjeldatakse valminud rakenduse arhitektuuri, esinenud probleeme ja lahendusi. Lisaks nõuetele loodavale rakendusele lisandub nimekiri funktsionaalsustest, millega hiljem rakendust täiendada. Viimasena võetakse kokku tehtud töö tulemused ning antakse hinnang valminud rakendusele.

## 2. Analüüs

Eesti Väärtpaberite registri pidamise seadus (edaspidi EVKS) § 7 lõige 1 ja 2 kohaselt on Nasdaq Tallinna börsi põhi- ja lisanimekirjas kaubeldavate ettevõtete aktsionäride nimekirjad avalikud[5]. Aktsiaraamatud avaldatakse peale igat börsipäeva Eesti väärtpaberiregistri pidaja Nasdaq CSD SE Eesti filiaal (edaspidi registripidaja) kodulehel.

Börs on avatud ja läbipaistev kauplemisskeskkond, mis viib omavahel kokku raha paigutavad investorid, täiendavat kapitali kaasata soovivad ettevõtted ning börsi liikmed, kes vahendavad investorite väärtpaberitehinguid ühtses elektroonilises kauplemissüsteemis. See on üks võimalik lahendus, mille kaudu ettevõtted saavad kaasata kapitali selleks, et investeerida ettevõtte projektidesse või rahastada üldist arengut ning seeläbi luues uusi töökohti või efektiivistada tootmist. Põhjus võib olla ka lihtsas soovis tõsta ettevõtte tuntust, luua töötajate motivatsioonipakett optsoonide näol, laiendada investorite ringi või luua võimalus väljuda mõnel praegusel investoril. Tänu börsi poolt kehtestatud reeglitele väheneb ettevõtete juhtkondade võimu kuritarvitamise tõenäosus ning suureneb turu läbipaistvus ja efektiivsus.

Börs ise aktsiate hindu ei kujunda, hinna paneb paika nõudluse ja pakkumise vahekord ehk teisisõnu: hinnad kujunevad investorite poolt tehtud tehingute tulemusel. Börs on justkui baromeeter riigi majandusliku olukorra mõõtmiseks. Peamisteks funktsionaalsusteks on hinnastamise kujundamine, tehingute turvalisuse ja likviidsuse tagamine. Viimane on oluline, et investorid saavad olla kindlad, et neil on võimalik kiiresti ja lihtsalt oma aktsiad maha müüa, millal iganes neil selleks vajadus peaks tekkima. Seega on tänu likviidsele aktsiabörsile võimalus investeerida pikaajalistesse projektidesse, kuid väljuda investeeringust endale sobival ajal.[6]

### 2.1. Probleem

Aktsiaraamatuid on avaldatud alates 2001. aastast ja juba siis lubati, et on plaan pakkuda kõigile võimalust saada ettevõtte aktsiaraamatu väljavõtteid erinevatel ajahetkedel ning teha päringuid, et hõlbustada informatsiooni analüüsimist[7].

Iga investor peaks hoidma ennast kursis ettevõtete käekäigu kohta kuhu ta on oma kapitali paigutanud. Selleks on hea võimalus kaasosanike ja toimunud tehingute tundmine.

Raamatute muutuste põhjal on võimalik luua investeerimis strateegiaid ning tuntud nimed ja ettevõtted nimekirjas võivad mõjutada potentsiaalsete investorite huvi antud emitendi kohta.

Ometi on täna olukord ikkagi selline, et avalikult tasuta on võimalus näha ainult viimase tehingupäeva seisuga põhi- ja lisanimekirjas kauplevate ettevõtete täielikku nimekirja. Nimekirjad avalikustatakse PDF failide kujul, mis on küll pisike areng võrreldes perioodiga kuni 2022.a kevad kui andmed olid HTML tabelis ja suuremate nimekirjade puhul kulus kohati mitu minutit, et nimekiri täielikult avaneks.

Tasulise teenusena tellitud päring ühe ettevõtte kohta maksab eraisikutele 1.44 EUR(1.20 + KM) ja tasuta on võimalik eesti suuremate internetipankade kaudu. Täiendava tasu eest on võimalik pärida ka Läti ja Leedu väärtpaberiomaniike kohta, kui väärtpabereid hoitakse Läti ja/või Leedu kontohaldurpankade esindajakontodel. Lisaks on võimalik tellida ka automaatseid päringuid, mis maksab 12 EUR + KM kvartalis. Ettevõtte esindajale (juhatuse liige või volitatud e-Registri kasutaja) on ettevõtte enda poolt emiteeritud osade ja aktsiate kohta päring tasuta[8]. Avalikke tasulisi päringuid e-registrist kõigi aktsiaseltside hetke seisuga koosta saab teha ainult tööpäevadel ajavahemikul 07:00 kuni 19:45. Registripidaja lubab päringu vastust 1 minuti jooksul, kuid tegelik test näitab, et vastuse saamiseks kulus umbes 15 minutit mis on mõistlikuks kasutamiseks liiga pikk aeg. Päringud on saadaval HTML3, XLS3 ja CSV failina. Ainuke sisuline eelis tasulises päringus on see, et nimekirjas on lisaks isiku/ettevõtte nimele näha ka isikukood või ettevõtte puhul registrikood.

Suheldes teiste huvilistega selgus, et tasulist teenust kasutamata salvestatakse perioodiliselt andmed käsitsi excelisse ja koostades erinevaid valemeid luuakse endale aruanne toimunud muutustest aktsiaraamatus. Antud tegevus eeldab korralist ja mahukat manuaalset tegevust.

## **2.2. Tallinna börs**

Aktsiakauplemine Nasdaq Tallinna börsil toimub kauplemissüsteemis INET, mis on kasutusel kõigil Nasdaq aktsiaturgudel üle maailma. Börs on litsentseeritud Eesti Finantsinspektsiooni poolt, kes teostab börsi üle ka järelevalvet[9]. Tallina börs on osa Nasdaq Balti börsist, kuhu kuulub veel Riia ja Vilniuse börs. Need on ainukesed reguleeritud börsid Baltikumis[10]. Tänapäevane börs avati 31. mail 1996, mil oli noteeritud

11 väärtpaberit ning on alates asutamisest elektrooniline väärtpaberi turg. Regulaarne kauplemine Tallinna Väärtpaberibörsil algas samal aastal 3. juunil[11][12].

Balti turul on emitendid kolmes nimekirjas, põhinimekiri, lisanimekiri ja First North alternatiivnimekiri. Peamine erinevus seisneb ettevõtete suuruses, vanuses ning raporteerimis nõuetes. Põhinimekiri on mõeldud raha kaasamiseks ja noteerimiseks ettevõtetele, kes vastavad kõrgeimatele raporteerimise ja läbipaistvuse nõuetele. Lisanimekiri on hüppelaud börsile kui ettevõtte vanus ei ole veel põhinimekirjas nõutud kolm aastat ning turuväärtus 4 miljonit eurot. Alternatiivturg on mõeldud eelkõige uutele, kiiresti kasvavate ettevõtetele investorite raha kaasamiseks ning tuntuse tõstmiseks. Lisaks puuduvad ettevõttele vanuse, omakapitali ja ranged info avalikustamise nõuded.[13]

Tallinna börsi põhinimekirjas on 2023. aasta veebruari seisuga 18, lisanimekirjas 2 ja alternatiiv nimekirjas First North 13 emitenti. Nende turukapitalisatsioon kokku on umbes 5 miljardit eurot. Võrdluseks siis Vilniuse turg on sammuti 5, Riia turg 0,8 miljardit ning USA turg umbes 43 triljonit dollarit[14][15]. Kauplemismaht eelnimetatud Tallinna börsi nimekirjades kokku oli 2022. aastal üle 375 miljoni euro ja tehinguid ligi 950 tuhat[1]. Hetkeseisuga on ka kauplemis ja haldustasud balti aktsiatel Eesti suurimates pankades tasuta või väga soodsad mille kohta leiab täpsemat infot panga hinnakirjast. Kauplemiseks on turg avatud 5 päeva nädalas avaoksjoniga kell 10:00 ning lõpeb sulgemisoksjoniga kell 16:00[11]. Igal aastal on ka mõned kauplemispühad mis on leitav registripidaja kodulehel[16].

### **2.3. Olukord Tallinna börsil**

Hetkel on aktsiaraamatute väljavõtteks üks võimalus registripidaja kodulehel laadida alla aktsionäride nimekiri PDF faili kujul. Fail on kättesaadav ainult viimase väärtuspäeva seisuga ja vanemaid andmeid ei ole võimalik näha. Väärtuspäev on tavaliselt 2 tööpäeva pärast tehingut olev päev, mil isik saab ostetud aktsiate omanikuks või kaotab müüdüd aktsiate üle omandiõiguse.

Teine võimalus on kasutada Nasdaq elektroonilist teenuste keskkonda(e-registrit), kus saab tasulise teenusena teha taotlus ja tellida endaga seotud väärtpaberiomanike nimekirjad. Endaga seotud nimekirjad on sellised, kus päringu tegemise hetkel oled aktsionär/osanik, kuid börsil noteeritute nimekirju saavad tellida kõik huvilised. E-

registrisse saab siseneda ainult autoriseerides end ID-kaardiga, Mobiil ID-ga või internetipankade kaudu.

Nimekirju on võimalik tellida ka ajaloost iga tööpäeva 07:00-19:00 täistunnise intervalli ja arveldussüsteemi tööpäeva lõpu seisuga. Arveldussüsteemi tööpäeva lõpp oleneb üleeuroopalisest väärtpaperite arveldussüsteemist T2S ning on eelduslikult Eesti ajavööndi järgi orienteeruvalt 19:45[17]. Tasulise teenusena saab ka registripidajalt tellida hetkeseisu aktsiaseltsides üle 10% osalust omavate osanike kohta, börsiettevõtete aktsiaraamatud kuvatakse täielikena.

Lisaks eelnevale on vastavalt väärtpaperituru seadusele emitendi juhil, tema lähedasel ja emitendiga seotud juriidilisel isikul tehingutest teavitamise kohustus[18][19]. Teavituskohustus tekib alates tehingust, mille kogusumma on kalendriaastal üle 5000 euro. Selle arvutamiseks liidetakse kokku kõik asjaomased tehingud ilma tasaarvelduseta. Tehinguteavitus tuleb teha finantsinspeksioonile ja emitendile mitte hiljem kui kolmandal tööpäeval pärast tehingu tegemise päeva. Emitentidega seotud isikute tehingud on avaldatud sellisel kujul, koosseisus ja mahus nagu nad on edastatud finantsinspeksioonile[20]. Tehinguid on võimalik filtreerida emitendi, tehingu sooritaja ja kuupäevade järgi ja registrit saab eksportida CSV faili.

Lähim lahendus selles vallas on Marko Jagori poolt 2022. aastal kirjutatud magistritöö „Tallinna börsi ettevõtete insaiderite tehingute jäljendamine: analüüs ning finantsrakenduse arendamine“. Antud lahendus jälgib ainult insaiderite ehk seotud isikute tehinguid. Arvestades tehingute hulka Tallinna börsil 2022. aastal mida oli eelnevalt väljatoodud 950 tuhat ja insaiderite tehinguid mida oli kõigest 366 siis see on kaduväike osa kõigest muutustest aktsiaraamatutes[21]. Vaadeldes andmeid Finantsinspeksiooni veebilehel olevast registrist on näha, et paljudel juhtudel teavitatakse tehingust mitmeid päevi hiljem kui tegelik tehing sai tehtud, seega ei anna rakendus ka ajalist eelist.

## **2.4. Olukord maailmas**

Enamik välisbörsidel noteeritud ettevõtted avaldavad oma aruannetes ainult suuremad aktsionärid. Tavaliselt on nad kõik suured investeerimisfondid või pangad.

Üksikutel juhtudel on ka ettevõtted ise avalikustanud peamised aktsionärid oma veebilehel. Näitena võib tuua Nasdaq Helsinki börsil noteeritud Elisa Oyj ettevõtte, kelle

kodulehel on toodud 20 suurimat aktsionäri koos ajalooaga[22]. Samas Telia Company näitel kes kaupleb nii Stockholmi kui ka Helsinki börsil on oma kodulehel toonud ainult 10 suurimat aktsionäri ja seda ka ainult 2022. aasta lõpu seisuga[23].

Täielikku nimekirja ettevõtteid üldiselt ei avalda ja toodud on vaid 10 kuni 100 suurimat osanikku. See on tavaliselt reguleeritud kohaliku riigi seadustega või on jäetud ettevõtte enda otsustada kas ja kuidas ta aktsiaraamatut avaldab.[24][25]

## **2.5. Andmekaitse**

Töö teema valikul ja kirjutamisel on lähtunud väärtpaberite registri pidamise, väärtpaberi turu ja isikuandmete kaitse seadustest. Kõnealuseid andmeid on juba üle 20 aasta hoitud avalikus registris. Registri andmeid on töödeldud ja analüüsinud nii väikeinvestorid kui ka majandusteemaline väljaanne Äripäev, kes teeb regulaarseid väljavõtteid sisuga “Investorite TOP” kus on reastatud suurimad Tallinna börsi investorid. Arusaadavalt nii mõnelegi investorile selline pingeritta seadmine nende nimega pole kuigi meeltnööda. Andmete käitlemisel on lähtunud sellest, et tegemist on andmetega mis pärinevad avalikust registrist. Äripäev on vaidluses eraisikuga antud teemal riigikohtus võitnud, kui isiku arvates ei oleks tohtinud tema andmeid sellisel kujul töödelda ega avaldada. Riigikohus leidis, et isiku vastu võib eksisteerida avalik huvi ka siis, kui ta ei kvalifitseeru üldisemas plaanis avaliku elu tegelaseks.[26] Tallinna börsi likviidsus on kohati nii väike, et pole võimalik tõmmata selget piiri kust algab avalik huvi, seega pole oluline aktsiate arv ega investeeringu eesmärk.

Väljavõtte kohtulahendist 3-3-1-85-15: “Tallinna börsi toimimiseks kehtestatud reeglitest tulenevalt peavad börsil noteeritud aktsiate omanikud, sõltumata nende osaluste suurusest, taluma seega teatavat eraelu puutumatuset riivet, mis seondub nende varalist seisuga kajastavate andmete kättesaadavusega, ning seda sõltumata hinnangust, kas investeerimist tuleb lugeda eraelu või avaliku elu sfääri kuuluvaks tegevuseks.

Kolleegium leiab, et avalik huvi Tallinna börsi toimimise, sh börsiinvestorite isikute ja nende investeeringute suuruse vastu kaalub üles börsiinvestori huvi eraelu puutumatusel.”[26]

## 2.6. Riskid

Üks tõenäolisemaid riske rakenduse jätkusuutlikuks toimimiseks on registripidaja otsus muuta kasutajaliidest selliselt, et muutub ka aktsiaraamatute ligipääs või formaat. Selline risk on juba arenduse käigus realiseerunud kui muutus pisut nii registripidaja kasutajaliides kui ka aktsiaraamatu faili kujundus. Tänu ettenägelikult valitud tehnoloogiatele sai vastav muudatus rakendusse lihtsa vaevaga tehtud ning rakenduse töö jätkus.

Teine risk on olukord kui registripidaja otsustab muuta ka kõik ajaloolised andmed tasuta kõigile kättesaadavaks. Arvestades töö eesmärki oleks selline stsenaarium pigem positiivne, kuid vaadeldes hetkeolukorda on see pigem vähetõenäoline. Lisaks arvestades rakenduse plaanitud edasiarendusi mis on kirjeldatud vastavas peatükis täidaks rakendus jätkuvalt oma unikaalset eesmärki.

Halvima stsenaariumi järgi muudetakse aktsiaraamatud Tallinna börsil privaatseks või väga piiratud ligipääsuga loetavaks. Lähtudes sellest, et hetkel reguleerib seda EVKS, pole see risk kuigi suur.



### **3. Nõuded**

Rakenduse sihtgrupiks on Tallinna börsi tänased ja potentsiaalsed investorid. Uue lahenduse nõuded on koostatud sihtgrupi abiga ja on jagatud funktsionaalseteks ning mittefunktsionaalseteks nõueteks. Rakenduse nõuded on tingimused või võimalused, mis kirjeldavad seda, kuidas probleemi saaks lahendada.

#### **3.1. Funktsionaalsed nõuded**

Funktsionaalsete nõuete all peetakse silmas rakenduse kasutaja tegevusi sisaldavaid nõudeid. Funktsionaalsete nõuete abil saab hea ülevaate loodavasse rakendusse planeeritavatest funktsioonidest[27].

- Rakendus peab olema kõigile vabalt kättesaadav.
- Kasutaja peab saama otsida nii emitendi kui ka isiku nime järgi.
- Tulemus peab olema võimalikult korrektne ja ajakohane.
- Rakendus peab kasutama andmebaasi info salvestamiseks.
- Rakendus peab saama hakkama väga suure ja pidevalt uueneva andmehulgaga.
- Rakendus tuleb luua veebirakendusena ning kasutama reageerivat veebidisaini.
- Kasutajal on võimalik näha ettevõtte ja osaniku kohta ajaloolisi andmeid.

#### **3.2. Mittefunktsionaalsed nõuded**

Mittefunktsionaalsed nõuded kirjeldavad rakenduse kvaliteeti ja kuidas süsteem toimib või käitub[28].

- Rakendus peab kuvama kasutajale vajadusel mõistetavaid veateateid.
- Rakendus peab toimima enamlevinud seadmetel ja brauseritel.
- Rakendus peab olema optimeeritud ja jõudluse poolest kiire.
- Rakenduse kasutajaliides peab olema lihtne, intuitiivne ning navigeerimine kasutaja jaoks selgesti mõistetav.
- Rakendus peab olema lihtsasti hooldatav ja edasi arendatav.
- Rakendus peab olema suuteline jätkama ka vea tekkimisel.
- Kasutajaliides peab järgima kasutajasõbraliku disaini printsiipe ning olema läbivalt sarnane.

## 4. Arendusvahendid

Järgnevad peatükid vaatlevad serveri ning kliendipoolseid tehnoloogiaid lahenduse teostuseks. Rakenduse arendamiseks vajalike tööriistade valikul lähtutakse seatud nõuetest, piisavast kasutajaskonnast ning varasemast kogemusest. Lähenemine peab olema paindlik ja lihtsasti täiendatav. Tehnoloogiad peavad olema turvalised nii rakenduse kasutaja kui ka haldaja huvides, samas ei tohi olla tasulised.

Serveripoolse rakenduse eraldamine kliendirakendusest võimaldab eraldada vaated täielikult äriloogikast ning neid arendada eraldiseisvalt. Lisaks annab selline lähenemine suuremat paindlikkust nii programmeerimiskeelte kui ka raamistike valikul. Miinuspoolena võib välja tuua lisatöö tegemise andmete valideerimisel ja töötlemisel. Näiteks kliendi poolses rakenduses on vajadus uuesti defineerida kõik andmeobjektid, mis tegelikkuses juba teenuse pool on defineeritud.

### 4.1. Serveripoolsete tehnoloogiate valik

Serveripoolse valiku peamiseks kriteeriumiks on see, et rakenduse eesmärgiks on veebiandmete kaapimine (*web scraping*) dünaamiliselt veebilehelt. Olemas on mitmeid valmis tööriistu mis ei eelda eelnevat programmeerimis kogemust, kuid need kõik said välistatud selle tõttu, et vajalik on captcha pildi lahendamine ning selline võimekus on ainult suurtel tasulistel vahenditel[29]. Arvestades kriteeriumeid tõuseb populaarsematest tehnoloogiatest tugevalt esile Python ja TensorFlow kombinatsioon.

#### 4.1.1. Python

Python on 1991. aastal loodud mitme otstarbeline programmeerimiskeel, mille abil on võimalik luua nii objektorienteeritud rakendusi kui ka lühikesi skripte. Peamised rakendus valdkonnad on veebiarendus, automatiseerimine, andmeanalüüs, visualiseerimine ja masinõpe. Pythoni populaarsuse taga on kergesti loetav süntaks, avatud lähtekood, lai ja aktiivne kogukond ning suur valik teke erinevateks vajadusteks. Lisaks peetakse seda üheks lihtsamaks esimeseks programmeerimiskeeleks algajale.[30]

#### **4.1.2. TensorFlow**

TensorFlow on Google Brain tiimi poolt loodud avatud lähtekoodiga tarkvarateek masinõppe mudelite loomiseks ja kasutamiseks. Platvorm koosneb mitmekülgsatest tööriistadest, mis on mõeldud arendajatele, ettevõtetele, teadlastele kui ka lihtsalt huvilistele, kes soovivad luua masinõppel töötavaid lahendusi nii veebilehitsejatele, serveritele, pilve- kui ka mobiilisüsteemidele. Google ise kasutab TensorFlow'd paljudes enda poolt pakutavates teenustes nagu Gmail, Translate, Maps, Android ja YouTube. Lisaks kasutavad tehnoloogiat ka paljud suurimad tehnoloogiaettevõtted nagu Airbnb, Arm, Intel, Lenovo, PayPal, Spotify, Twitter ja paljud teised[31]. Antud töös kasutatakse TensorFlow'd CAPTCHA piltide lahendamiseks.

#### **4.1.3. Selenium WebDriver**

Selenium WebDriver on avatud lähtekoodiga testimistarkvara mis ei vaja vaheserveit, et suhelda brauseriga. Selenium on laialdaselt kasutatud vahend, mida kasutatakse küll peamiselt automaatsete koostamiseks, et testida kasutajaliideseid, kuid võimaldab ka edukalt veebilehtede kaapimist. Üheks olulisemaks eeliseks teiste populaarsemate teekide ees nagu BeautifulSoup ja Scrapy on dünaamiliste veebilehtede tugi mis antud probleemi juures on üks olulisemaid aspekte. Dünaamilised veebilehed on sellised, kus veebilehe sisu muutub ilma lehte uuesti laadimata.

Selenium pakub tavalise veebilehitsejaga identseid võimalusi, mis omakorda teeb veebilehe kaapimise lihtsaks ja kiireks ka olukordades kus toimuvad pidevad kasutajaliidese muudatused. See omakorda tagab väga kiire ja lihtsa halduse loodavale rakendusele. Tagavaks plussiks on tasuta litsents, tugev ja suur kogukond, kõigi populaarsemate programmeerimiskeelte ja veebilehitsejate tugi. Miinusena võrreldes konkurentidega peab mainima kindlasti kaapimis kiirust, kuid käsitletavate mahtude juures ei mängi see olulist rolli ning kasutatavus kaalub selle tugevalt üle.[32][33]

#### **4.1.4. Apache Tika**

Apache Tika on vabavaraline tööriistaraamistik tekstiliste andmetega( nt PPT, XLS ja PDF) tutvumiseks ja nende analüüsimiseks. Kõiki neid faile saab filtreerida ja töödelda vastavalt vajadusele näiteks tõlkimiseks, analüüsiks, andmekorjeks ja paljuks muuks. Tika on kirjutatud Javas, kuid tänu REST serverile ka edukalt kasutatav ka python

platvormil.[34][35] Tika küll nõuab Java 7+ olemasolu serveris, kuid võrreldes näiteks PyPDF2 ja PDFtoText teekidega python platvormil on tulemas tunduvalt parem ja lihtsamini töödeldav.

## 4.2. Kliendipoolsete tehnoloogiate valik

Kliendipoolse valiku peamiseks kriteeriumiteks sai lihtne struktuur, uudsed, kiired ja lihtsasti arendatavad tehnoloogiad.

### 4.2.1. Blazor

Blazor on osa ASP.NET ökosüsteemist C# ja HTML keelele ning Razor süntaksile toetuv raamistik. Veebilehitsejad otseselt ei toeta C# keelt, kuid enamikus moodsates brauserites on sisse ehitatud WebAssembly (WASM) standardile vastav virtuaalmasin. WASM jaoks kompileeritud kood on samade võimaluste ja õigustega kui JavaScript, kuid ei kasuta selleks peidetud teisendust JavaScriptist ja on märksa kiirem ja väiksema mahuga, kuna on kompileeritud kujul[36]. Kasutajale on võimalik serveerida Blazorile rajatud kasutajaliidest nii serveris kui otse kliendi masinas. Selle abil on võimalik serveerida baitkoodi otse kasutaja brauserisse ning sellevõrra vähendada sõltuvust massiivsetest JavaScript baasil toimivatest raamistikest.[37] Suurimateks eelisteks võib pidada võimalust kirjutada nii klient kui ka server rakendus samas tehnoloogias ning seeläbi kiiret arendusprotsessi. Samuti on toetatud programmeerimistehnikad nagu sõltuvuste süstimine(*dependency injection*), muutes olekuhalduse(*state management*) väga lihtsaks.[38]

### 4.2.2. Entity Framework Core

Üks osa rakenduse arendamises, mida on võimalik lahendada raamistike abil on suhtlus andmebaasiga. Entity Framework Core(edaspidi EF) on vabavaraline ORM(*Object Relational Mapper*), mõeldud kasutamiseks .NET platvormiga. Eesmärk on luua otseühendus koodis olevate olemite, seostatud andmete ning neid haldavate relatsiooniliste andmebaasi süsteemidega.[39]

Sarnaselt Hibernate, SQLAlchemy ja teiste ORM tarkvaradega on läbi EF võimalik teha andmebaasipäringuid otse koodi tasandil. ORM puhul on võimalus luua andmebaasimudel otse koodi tasandil kirjutatud olemite põhjal ning automatiseerida

andmebaasi migratsiooni. Samuti võimaldab EF luua täiendavaid abstraktseid andmestruktuure mille abil on võimalik defineerida asünkroonsed baasoperatsioonid kõikidele olemitele andmebaasi lisamisel, eemaldamisel, uuendamisel ja kustutamisel. Kasutades neid täiendusi on võimalik märkimisväärselt vähendada lähtekoodi täiendamist uute olemite lisamisel.[40]

EF kiirendab märgatavalt arendaja tööd optimeerides päringuid ja võimaldades objektidena kirjeldatud mudelite põhjal tabeleid genereerida. Tänu sellele, et pole otsesuhtlust andmebaasiga on parandatud ka turvalistust. Siiski on võimalus ka otse andmebaasiga suhelda, kasutades EF spetsiifilist LINQ või SQL vormingut. Kuna suhtlust vahendab raamistik, siis on lihtsam vajadusel ka andmebaasi vahetada.

### **4.3. Andmebaasi valik**

Selleks, et rakenduse andmeid talletada, on vajalik valida rakendusele sobiv andmebaas. Nii serveri kui ka kliendipoolne rakendus suhtleb aktiivselt andmebaasiga mis on kogu rakenduse aju. Seega võib andmebaasi valikut pidada olulisemaks kogu rakenduse juures. Valiku tegemisel tuleb valida andmebaasi struktuur ning seejärel vastava struktuuriga andmebaasi haldussüsteem. Andmebaasi haldussüsteemi valimisel tuleb samamoodi nagu muude tehnoloogiatega kaaluda populaarsust, materjalide kättesaadavust, jõudlust, funktsionaalsust ning hinda.

Andmebaasid jagunevad põhiliselt viieks erinevaks kategooriaks: hierarhiline, võrgumudeli põhine, objektorienteeritud mudel, mitterelatsiooniline ja relatsiooniline mudel.

- Hierarhilises andmebaasis on andmed puustruktuuriga, kus on alati kasutusel üks mitmele seosed. See ei ole kuigi paindlik olemite suhete poolest, mille tõttu ei ole see enam tänapäeval populaarne.
- Võrgumudeli põhine andmebaas on sarnane hierarhilisel mudelil põhinevale andmebaasile, kuid seda on kohandatud nii, et alamolemid saavad ennast siduda mitme vanemolemiga. Negatiivseks pooleks on selle struktuuri keerukus ja sõltuvus, mille tõttu võib osutada väga raskeks seda muuta.
- Objektorienteeritud andmebaasis on andmed esitatud objekti kujul. Kasutatakse suure jõudluse, keerukate algoritmide või kiiremate tulemuste puhul. Puuduseks on vähene tehniline tugi erinevatel programmeerimis kehtel ja raamistikel.

- Mitterelatsiooniline andmebaas ehk NoSQL andmebaas salvestab andmeid kasutades hoidmise mudelit, mis on optimeeritud spetsiaalsete nõuete jaoks tulenevalt andmestikust, mida salvestatakse. Tegemist hilisema tehnoloogiaga, mis on hoogu kogunud viimasel sajandil. Andmebaasides võivad andmed olla talletatud erineval kujul – võti-väärtus kogumina või JSON väärtusena. Tuntuim näide NoSQL andmebaasist on MongoDB.
- Relatsiooniline andmebaas ehk SQL andmebaas salvestab andmeid tabelites ning ridade kaupa. Selle tööpõhimõte seisneb informatsiooni sidumisel mitmetest tabelitest kokku, kasutades selleks võtmeväärtusi. Populaarsemad nendest on Microsoft SQL Server, Oracle, PostgreSQL ja MySQL.[41]

Käesoleva töö jaoks on kõige pädevam relatsiooniline andmebaas. Peamised põhjused valikuks on paindlikud suhted olemite vahel ja pikas perspektiivis andmemahu kasvades stabiilsuse ning töökindluse säilitamine. Rakenduse kasutushulk ei saavuta niivõrd kõrget mahtu, et tuleks kaaluda NoSQL andmebaasi, ning samuti on tegemist struktureeritud andmetega, mis teeb relatsioonilisest andmebaasist sobiva valiku.

PostgreSQL on avatud lähtekoodiga ja relatsiooniline andmebaasi haldussüsteem. See on ühilduv mitmete platvormidega ning on tugevalt optimeeritud jõudlusega, mis tavaliselt on saadaval ainult kallites kommertsilikes andmebaasides. Samuti on PostgreSQL andmebaasis tugev rõhk andmete terviklikkusel. Populaarsust soodustavad mitmed tegurid. Esiteks on sellel aktiivne avatud lähtekoodiga kogukond, mida ei kontrolli ükski sponsor ega ettevõtte. Teiseks on funktsioonide komplekt pidevas täiendamises ning on väga lihtne paigaldada ja üles seada.[42]

MySQL on vabavaraline relatsiooniline andmebaasi haldussüsteem ning seda peetakse üheks populaarseimaks arendajate seas. MySQL on töökindel ning üldkasutuseks sobilik haldussüsteem, mis jääb küll omaduste poolest teistele alla, kuid heastab selle oma kasutajalihtsuse ning andmebaasi lugemisoperatsioonide kiirusega. Tegemist on laialdaselt kasutatavata andmebaasiga, mille arendusel keskenduti kiirusele ja töökindlusele. Kuid sellest tuleneb ka selle puudus: osad SQL võimalused on jäetud välja, et muuta andmebaasimootorit kiireks (näiteks full-join). Kompromiss selle võimekuse saavutamiseks on kehvem jõudlus komplekssete päringute tegemisel. Seda on algajatel lihtne kasutada ning andmemahu voo kasvades suures mahus skaleeritav.[43]

MariaDB on samuti populaarne relatsiooniline andmebaas, mis on loodud MySQL-i arendajate poolt. See on avatud lähtekoodiga ja seda on võimalik kasutada enamlevinud operatsioonisüsteemide peal. MariaDB suudab pakkuda sama funktsionaalsust, mis MySQL ning ka uut funktsionaalsust nagu näiteks tugi JSON API-de jaoks. Lisaks sellele on MariaDB suuremas osas ühilduv MySQL'ga, mis võimaldab oma andmebaasid ilma suurema vaevata üle viia MariaDB'sse.[44]

Valiku tegemisel välistati andmebaasi süsteemid mis on tasulised, lokaalsed ja mõeldud pigem suurematele rakendustele. Autor valib andmebaasi haldussüsteemiks MySQL'i. Haldussüsteem sobib, sest rakenduse peamine koormus andmebaasile saab olema lugemisoperatsioonide teostamisel ning populaarsus ja laialdane kasutus arendajate seas tagab vajadusel abistavad materjalid. Samuti on tänu MySQL'i platvormiülesusele mugav seda lokaalse arenduse jaoks kasutada. Rakenduse ja andmemahu kasvamisel ei ole välistatud MariaDB või PostgreSQL peale kolimine.

#### **4.4. Arenduskeskkonnad ja koodi hoidla**

Arenduskeskkond on keskkond, kus on kasutatav kogu vajalik funktsionaalsus tarkvarasüsteemi arendamiseks. Käesoleva töö puhul käsitletakse selle all versioonihaldustarkvara ja arenduskeskkonda.

##### **4.4.1. Arenduskeskkonna valik**

Rakenduse arendamisel kasutatav integreeritud arenduskeskkond (*Integrated Development Environment, IDE*) peaks toetama tööd valitud raamistiku ja programmeerimiskeelega, milleks on klientrakenduse puhul C# ja .NET ning serverrakenduse puhul python.

JetBrains on viimase aastakümne jooksul tõusnud üheks juhtivamaks tarkvaraarenduse tööriistade loojaks. JetBrains Rider on nende tasuline IDE mõeldud just .NET rakenduste arendamiseks. Selle funktsionaalsus ja mugavus on võrreldav Visual Studio valikuga ning võib isegi öelda, et kohati hulga parem. JetBrains tooteid saab väga lihtsalt kohandada vastavalt maitsele. Näiteks on sisseehitatud versioonihaldus, koodi silumine, valideerimine ja refaktoreerimine, ühik testide tugi, suur laienduste valik, mugav koodi navigeerimine jpm.[45]

Peamiseks serveripoolse rakenduse arenduskeskkonnaks oli JetBrains PyCharm mis on mõeldud pythoni arenduseks. TalTech tudengitele on võimalik kasutada kõiki JetBrains Professional tooteid tasuta[46].

#### **4.4.2. Koodihaldustarkvara valik**

Versioonihaldussüsteemid pakuvad võimaluse muudatusi jälgida, mitmel arendajal korraga sama projekti kallal töötada, kasutajaliidese abil faili eri versioonidest visuaalne ülevaade saada, mitme arendaja samaaegse muudatuse tõttu tekkinud vastuolud lahendada ja rakenduse publitseerimine. Tuntumad sellised tööriistad on GitHub, GitLab ja BitBucket. Neist kõigil on eelpool loetletud võimalused olemas ning on tasuta kasutamiseks mitteärielistel eesmärkidel. Samuti on neil kõigil nii graafiline kui ka käsuraapõhine liides, mille abil eri versioonide vahel mugavalt navigeerida.[47]

Kuna ühelgi neist keskkondadest ei ole teiste ees eriti kaalukaid eeliseid ega puudusi, valitakse käesoleva töö lähtekoodi halduseks GitHub, mille kasutamise kogemus on olnud kõige suurem ning klient ja serverrakendus hoitakse erinevates repositooriumites.



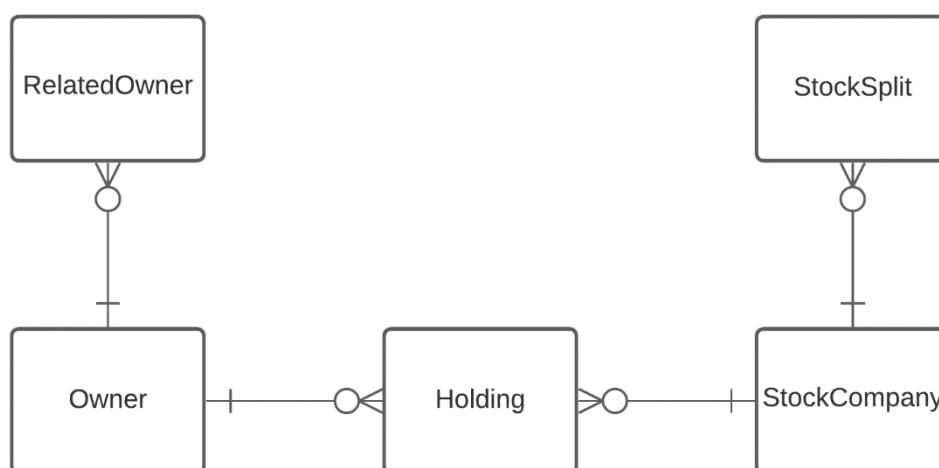
## 5. Rakenduse arendus

Projekti raames alustati andmebaasiskeemi koostamisega seejärel kirjutati valmis serverrakendus ja lõpuks klientrakendus. Ülesandele läheneti objektorienteeritud programmeerimise tavadest lähtudes mis jagab rakenduse arusaadavaks ja lihtsalt hallatavateks osadeks.

### 5.1. Andmebaas

Andmebaas on loodud „kood esmalt“ (*code first*) põhimõttel, mis tähendab, et andmebaasi tabelid, väljad, seosed ning piirangud kirjeldatakse koodi tasemel. Tabeleid ei looda käsitsi vaid selle eest hoolitseb ORM mis migratsiooniga genereerib kõik klassidesse kirjeldatud tabelid koos väljadega[48].

Relatsioonililise andmebaasi loomine iseenesest ei nõu andmeskeemi, kuid vaatama sellele on kasulik see koostada, kuna aitab varakult paika panna kõik vajalikud tabelid, väljad ja nendevahelised seosed. Samuti on iga tabeli kohta kirjeldatud selle sisemine struktuur – väljad, nende andmetüübid ning piirangud. Rakenduse diagramm (joonis 1) koosneb 5 olemist ning on loodud kasutades tööriista Lucidchart[49].



Joonis 1 Lihtsustatud olemissuhte diagramm

Täielik diagramm on leitav Lisa 2 kus on näha ka kõikide tabelite väljad koos andmetüüpidega.

Järgnevalt on kirjeldatud kõikide tabelite eesmärki ja sisu. Tabelite ja väljade nimed on ingliskeelsed, sest ka kogu kood on nii.

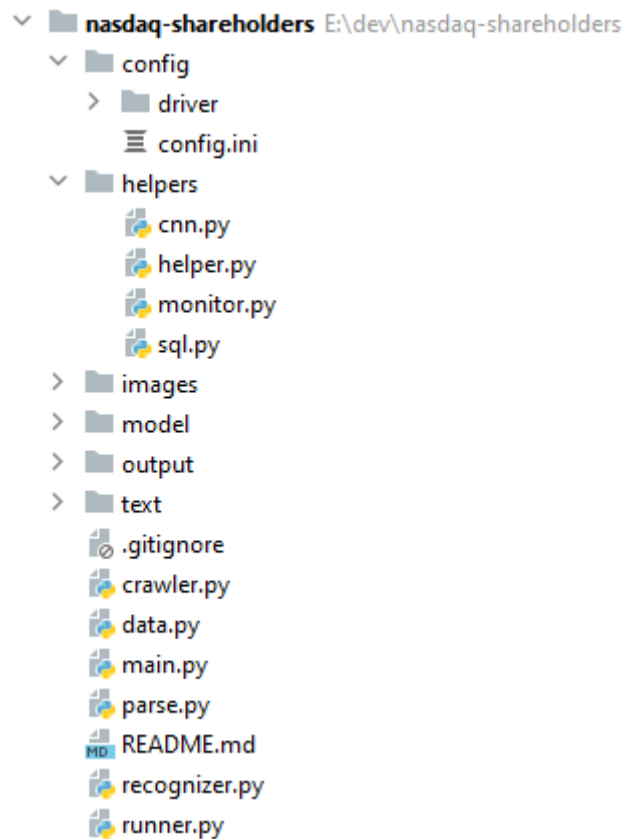
- Owner – Isik/ettevõtte. Tabelis hoitakse kõik aktsiaraamatutes leiduvate osanike unikaalsed nimed. Päevases aktsiaraamatu nimekirjas ei ole toodud isikukoodi ega ettevõtte puhul registrinumbrit, seega on kõik nimed grupeeritud.
- RelatedOwner – Seotud isik/ettevõtte. Tabelis hoitakse isikute ja ettevõtete omavahelisi võimalike seoseid. Juhul kui mõne ettevõttega on seotud isik kes on samuti osanike nimekirjas saab tõmmata nende vahele mugava seose ja seda kasutada kasutajaliideses.
- StockCompany – Emitent. Tabelis hoitakse nimekirja emitentidest kes on noteeritud Tallinna börsi nimekirjas. Salvestatud on ka kuupäev mis päeva seisuga on andmed uuendatud.
- StockSplit – Aktsiatükeldus. Tabelis hoitakse aktsia jagunemise infot, et tükelduse puhul aktsiate arvus anomaaliaid ei tekiks. Aktsiate tükeldamiseks nimetatakse aktsia nimiväärtuse vähendamist või suurendamist aktsiakapitali suurust muutmata. Näiteks kordaja 10:1 jagunemise puhul saab iga investor igale aktsiale lisaks 9 aktsiat. Eesmärk on vähendada aktsia hinda, et see muuta rohkem kättesaadavamaks.[50]
- Holding – Positsioon ettevõttes. Tabelis hoitakse aktsiaraamatute seisu börsipäeva lõpu seisuga. Aktsiapositsiooni muutuse korral salvestatakse tabelisse kirje aktsiate arvu ja kuupäevaga. Samuti on salvestatud olukorrad kus isik või ettevõtte kustutatakse aktsiaraamatust, see on märges kogu positsiooni müümise kohta.

## **5.2. Serverrakenduse arendus**

Serverrakendus vastutab registripidaja veebilehelt andmete perioodilise kogumise ja salvestamine eest.

### **5.2.1. Struktuur**

Rakenduse struktuur on jagatud kaustadeks ja objektideks. Joonisel 2 on näha serverrakenduse põhilised komponendid.



Joonis 2 Serverrakenduse struktuur

Rakendus käivitatakse *main.py* objektis olevast *Main* klassist, kus jagatakse kõik emitendid lõimete(*thread*) vahel ära, et tõsta programmi efektiivsust. Selline emitentide jagamise eesmärk on töödelda mitut emitenti korraga, et tagada kiirem andmete uuendus. Testid näitavad, et selline lähenemine on andnud umbes neljakordse ajalise võidu võrreldes sellega kui kogu protsess töötaks ainult ühes lõimes.

Igal lõimel on oma klass objektist *runner.py*, mis sisaldab endas eelnevalt jagatud emitentide nimekirja mille alusel lähtestatakse *crawler.py* objektis olev andmete kaapimis objekt. Ettevõtte valimisel Selenium teegiga mille driver asub kataloogis *config/driver/* laetakse alla turvakoodi pildifail mis saadetakse tuvastamisele *recognizer.py* objektis olevasse klassi mis tagastab neljast sümbolist koosneva sõne. Sõne sisestatakse kaapimis objektis emitendi turvakoodi väljale mille tulemusel saadakse vastuseks kas aktsiaraamatu PDF fail või veateade kujul „Turvakood on vale“. Saadud tulemuse põhjal salvestatakse ka turvakoodi pildifail kataloogi */images* mille protsess toimub objektis *helper.py*. Kui turvakoodi lahendamine ei õnnestu korratakse sama protsessi uuesti.

Turvakoodi edukal lahendamisel toimub andmete lugemine PDF failist *parse.py* objektis olevas klassis, PDF fail salvestatakse kataloogi */output*, mille näide kuvatõmmisena on

näha Lisa 4. Andmete lugemine toimub rida rea haaval kus põhiliseks probleemiks on see, et nii järjekorra number, aktsionäri nimi, väärtpaberi arv ja osaluse protsent on ühes jorus ilma selgete eraldusmärkideta. Näiteks võib aktsiaraamatus olla omaniku nimi kujul „OÜ Mina ja koer 4“. Antud probleemi lahendamiseks kasutatakse eeldust, et aktsionärid on järjestatud väärtpaberite arvu järgi ning osaluse protsent vastab väärtpaberi arvule. Väärtpaberiomani ke üle peetakse registris arvestust kontopõhiselt, seega võib isik esineda ühes raamatus oma nimega mitmeid kordi. Kõik nimed grupeeritakse ning nimekirjas kajastub omaniku summaarne osalus.

Andmete edukal lugemisel toimub andmete uuendamine objektis *data.py*. Uuendamisel kontrollitakse kõigepealt omaniku nime olemasolu andmebaasis, vajadusel lisatakse uus omanik. Seejärel võrreldakse osakute arvu andmebaasi seisuga, arvu muutumisel lisatakse uus kirje. Lõpuks toimub lahkunud aktsionäride kontroll kus võrreldakse andmebaasis oleva aktsionäride nimekirja PDF failis olevate nimedega, kui esineb nimesid keda enam failis ei leidu lisatakse neile uus kirje väärtpaberite arvuga null. *Sql.py* failis on kõik vajalikud meetoid andmebaasiga suhtlemiseks ning tabelite struktuur kasutades ORM'i SQLAlchemy.

Kogu rakenduse tegevus logitakse, et vajadusel oleks võimalik takkajärgi tuvastada võimalikke tekkinud vigu või teha optimeerimist. Selline vajadus on eriti oluline rakenduse aktiivses arendus faasis. Logimiseks kasutatakse python standard teeki *Logging*, ning failid salvestatakse kataloogi /text. Lisaks on rakendusel ka monitooringu süsteem, millega saadetakse teavitusi email kujul juhul kui tehnilistel põhjustel peaks ebaõnnestuma andmete uuendamine, et saaks vajadusel kiirelt reageerida, sest andmete uuendamiseks on reeglina aega kuni 24 tundi. Rakenduse konfiguratsioon on sätestatud failis *config.ini*, kust leiab kõik tööks vajalikud parameetrid nagu andmebaasi ühenduse detailid, kataloogid, monitooringu ja logimise sätted.

### **5.2.2. Captcha lahendus**

Aktsiaraamatu PDF faili allalaadimiseks registripidaja kodulehelt on vaja lahendada captcha pilt ehk turvakood mille näidis on nähtav kuvatõmmisel Lisa 3. Captcha on test kasutajale, et teha kindlaks kas tegemist on tõesti inimese või robotiga kes rakendust kasutab. Vaatamata sellele, et lahendus on mõeldud automaatsete robotite blokeerimiseks on captcha ülesanded koostatud automaatselt, seega ei ole neid kuigi raske ka automaatselt lahendada.[51]

Lahenduseks on kasutatud närvivõrku CNN(*Convolutional Neural Network*), mida kasutatakse, et viia sisendpilt kujule kust lugeda tähtede ja numbrite kombinatsioon[52]. Närvivõrgu treenimiseks on vaja hinnanguliselt vähemalt 1000 pilti mis on eelnevalt valideeritud. Paraku täpselt selliseid captcha pilte nagu registripidaja kasutab kuskilt vabalt saada pole. Registripidaja lehelt piltide ükshaaval salvestamine ja valideerimine võtaks liiga kaua aega, seega sai loodud python skript mis genereerib võimalikult sarnased pildid. Pilt on mustvalge ning koosneb neljast tähtede ja numbrite kombinatsioonist. Lisaks on pildil suvaliselt paigutatud jooned ja täpid ning siis veel pilti moonutatud laineliseks. Joonisel 3 on näha koostatud skripti näide mis on genereeritud sama kombinatsiooniga joonisel 4 näitel registripidaja genereeritud turvakoodist.



*Joonis 3 Skripti genereeritud turvakood*



*Joonis 4 Registripidaja genereeritud turvakood*

Genereeritud 20 000 pildiga sai treenitud CNN mudel mille täpsusklass senise registripidaja 3695 pildi näitel on 22,68%(2857 ebaõnnestunud ja 838 õnnestunud pilti). Tulemus on küll päris madal, kuid arvestades väikest ajakulu piltide genereerimisel ja eesmärki võimalikult kiiresti andmete salvestamiseks on see rakenduse arendusjärgus täiesti piisav.

Rakenduse töö käigus salvestatakse kõik pildid, mille eesmärk on registripidaja piltidega luua uus mudel mille täpsus on suurem, et ei peaks tegema nii palju päringuid ning andmete kaapimine oleks kiirem. Kogutud 20 690 pildiga sai loodud uus treening mille täpsus on hetkel 33,17%(10669 ebaõnnestunud ja 5295 õnnestunud pilti). Tulemus on täiesti rahuldav arvestades asjaolu, et piltide käsitsi valideerimist treeningu andmeteks pole tehtud. Tulevikus rohkemate piltide kogumisel saab treenida veelgi parema mudeli mille täpsus on kindlasti üle 50%. Selle jaoks peab tõenäoliselt tegema ka käsitsi piltide valideerimist.

### 5.2.3. Serverrakenduse haldus

Rakendus majutub virtuaalses privaat serveris(VPS) operatsioonisüsteemil Ubuntu. Serveril on E5-2630 protsessor millest on 4 tuuma eraldatud, 16GB mälu ja 300GB kettaruumi. Kettaruum on vajalik andmete, logide ja piltide salvestamiseks. Lisaks on server kasutusel ka teiste autori rakenduste majutuseks.

Rakendus on seadistatud töötama cron töödega peale igat argipäeva hommikul kui registripidaja on andmed uuendanud. Täiendav töö mis täidab täpselt samasugust ülesannet on seadistatud töötama mõned tunnid hiljem. Eesmärk on teha kontroll kas andmete uuendamine õnnestus ja vajadusel andmete paranduseks. Selline vajadus võib tekkida olukorras kui registripidaja pole jõudnud hommikuks andmeid uuendada või on esinenud tõrkeid serveri või rakenduse töös.

Arendus toimus windows platvormil autori arendus(DEV) keskkonnas. Lisaks toodangu(PROD) keskkonnale on ka testimise(TEST) keskkond kus arendus täiendused testitakse vähemalt päeva jagu läbi enne kui toimub arenduse tarne toodang keskkonda. Eesmärk on veenduda, et parandused on korrektsed ning ei põhjusta lisa probleeme.

## 5.3. Klientrakenduse arendus

Klientrakendus on kontaktpunktiks lõppkasutajale, kust on lihtne ja kiire otsida aktsiaraamatute seisu nii emitendi kui ka aktsionäri järgi.

### 5.3.1. Server või WebAssembly

Rakenduse arhitektuuri mõttes on oluline otsustada, kas kasutada andmete edastamisel REST põhinevaid veebiteenuseid või on soov teha funktsionaalsus kättesaadavaks ühe rakenduse siseselt kasutades andmete juurdepääsu kihti (*data access layer* ehk DAL). Viimane võimaldab säästa raha, mis kuluks eraldi rakenduste haldamisele ning aega funktsionaalsusele ligi pääsemiseks API loomisel.[53] Server rakenduse eelisteks on veel andmete pärimis kiirus andmebaasist mis on antud juhul oluline kuna andmete maht on suur ja pidevas kasvus. Puuduseks võib tuua, et server rakendus ei tööta ilma võrguühenduseta(*offline*), kuid antud juhul ei ole see kriitilise tähtsusega.[54]

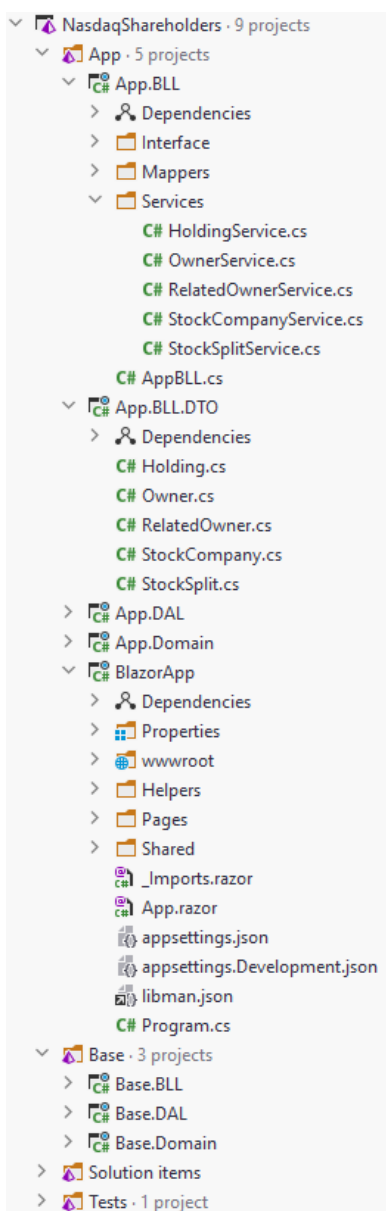
Kuna rakenduse jaoks on vajalik eelkõige suhtlus kasutajatega vaadete kaudu, mitte niivõrd andmete edastamine kolmandatele osapooltele, oleks mõistlik kasutada Blazor

server lahendust, mitte REST API põhinevat WebAssembly rakendust. Tulevikus nõudluse tekkimisel ei ole välistatud avalikud API veebiteenused.

### 5.3.2. Struktuur

Klientrakenduse struktuur on jaotatud erinevateks kihtideks, mis tagab pikemas vaates kiirema arendusprotsessi, kuna tõstab koodi kvaliteeti ning vähendab võimalikke vigu koodis ning ajakulu edasisteks arendusteks ja hooldusteks. Selline lähenemine tõstab skaleeritavust, töökindlust ja turvalisust. Igal kihil on oma ülesanne, mis teeb rakenduse erinevad osad ning seosed nende vahel mõistetavateks ja kergesti hallatavateks.[55]

Selle saavutamiseks on rakendus jaotatud joonisel 5 nähtaval kujul.



Joonis 5 Klientrakenduse struktuur

Kiht DAL(*Data Access Layer*) ehk andmetele juurdepääsu kiht on kasutusel andmebaasiga suhtlemiseks. Kihi peamisteks ülesanneteks on andmete lugemine, sisestamine, uuendamine ja kustutamine ning see võimaldab hoida kõik andmebaasiga seotud toimingud ühes kohas. Kihis on ka kasutusel hoidla muster(*repository pattern*), mille eesmärgiks on kihi abstraktsioon. Iga domeeni objekti jaoks on tehtud üks hoidla klass, mis tagab ligipääsu eelnimetatud loogikale.[56]

Sellele järgneb ärilooikakiht ehk BLL(*Business Logic Layer*), kus igale objektile vastab teenus klass(*service*) mis on justkui vahelüli andmebaasiga suhtlemisel andmete edastamiseks esitlus kihile. Kihis on leitav kogu ärilooika ning selleks, et vajadusel pääseda ligi mitmele hoidlale on kasutusel tööüksus ehk UOW(*Unit of Work*).[56]

Esitluskiht(*presentation layer*) on kõige kõrgem kiht ning leitav projektis „BlazorApp“ mille peamiseks ülesandeks on kasutajaga suhtlemine. Kasutajaliides koosneb erinevatest Blazor komponentidest mis tagab puhta ja selge struktuuri.

Andmete liigutamiseks läbi erinevate kihtide on loodud iga kihi jaoks eraldi edastusobjektid ehk DTO(*data transfer object*). Iga objekt sisaldab ainult kihile vajalikke parameetreid, see tagab, et kihis on ainult vajalikud andmed millele pääseb ligi ning ühe kihi muudatus ei mõjuta automaatselt teist. Objektide teisenduseks kihtide vahel kasutatakse *mapper* tüüpi objekte.

Kihtide vaheline sõltumatuse saavutamiseks kasutatakse sõltuvuste süstimist (*dependency injection*), mis põhineb abstraktsioonide vastu programmeerimise kontseptsioonil. Abstraktsioon saavutatakse jõustades(*implement*) madalama kihi liidest(*interface*), mis võimaldab kasutada kihi meetodeid, kuid ei avalikusta nende sisu.[57] Antud objektid on koondatud kõik vastavasse „Base“ projekti, et tagada selge struktuur.

### 5.3.3. Teek

Klientrakenduse komponentide loomiseks on otsustatud kasutada olemasolevat tasuta teeki mis võimaldab lihtsalt ja kiirelt luua vajaliku kasutajaliidese. Lähtudes selles, et rakenduse peamine sisu on suurte andmehulkade esitamine tabelite ja ruudustike(*grid*) kujul sai valitud just nende hulgaks kus on vajalik komponent juba olemas. Valituks osutus MudBlazor kuna on praktiliselt ainuke täiesti tasuta kasutatav teek millel on olemas vajalik funktsionaalsus. Teised tuntud ja võimekad teegid nagu Blazorise,



DevExpress, Syncfusion, Telerik, Infragistics ja Radzen on paraku kas osaliselt või täielikult tasulised ja langesid seetõttu valikust välja. Vaatluse all oli ka ametliku Blazor tiimi poolt arendatav „QuickGrid“, kuid kahjuks pole see veel saadaval .NET 7 versioonil ning hetkel kasutatav ainult .NET 8 eelversioonil[58].

MudBlazor on tasuta põhjalikult dokumenteeritud teek, millele kehtib MIT litsents. Teek on valmistatud spetsiaalselt Blazor raamistikule.[59]

#### **5.3.4. Klientrakenduse haldus**

Rakendus on publitseeritud kasutajatele läbi Docker konteineri mis kasutab hoidlana Docker Hubi. Konteinerid võimaldavad jooksutada palju erinevaid rakendusi samas serveris või kolida rakendust lihtsa vaevaga ühest serverist teise. Lisaks on see väga kiire ja mugav viis rakenduse publitseerimiseks ja halduseks ning annab rakenduse käituse keskkonna ühtluse nii arenduskeskkondades kui ka valmistoodangu keskkondades.

#### **5.4. Testimine**

Enamus rakenduse testimist on toimunud arenduse käigus manuaalselt testides erinevaid situatioone. Testimise eesmärgiks oli veenduda, et registripidaja väljastatud aktsiaraamatu sisu vastaks rakenduse andmebaasi seisule. Selleks on tehtud pidevaid pistelisi kontrole andmetes ning kaasatud ka teisi aktsiaraamatute huvilisi kes andsid tagasisidet nii andmetele kui ka rakendusele endale. Serverrakendusel on ka arendatud monitooringu süsteem mis saadab teavituse kui peaks tekkima probleeme andmete uuendamisel.

## 6. Tulemused

Lõputöö tulemusena valmis serverrakendus andmete korraliseks kogumiseks registripidaja veebilehelt ning kõigile vabalt ligipääsetav klientrakendus aktsiaraamatute mugavaks väljavõtteks. Serverrakendus sai arendatud Python programmeerimiskeeles, mis kasutades Selenium raamistikku kogub kokku iga börsipäeva aktsiaraamatute muudatused. Muudatused salvestatakse MySQL andmebaasi, kust need jõuavad kõigi huvilisteni läbi klientrakenduse. Rakendus sai arendatud Blazor raamistikul, mis võimaldab arendada klientrakendusi kasutades programmeerimiskeelt C#. Rakendus pakub unikaalset tasuta ja kiiret võimalust saada väljavõtte toimunud muudatustest Tallinna börsi aktsiaraamatutes mis suurendab kohaliku börsi läbipaistvust ja aitab investoreid nende investeerimisotsustel.

Töö kõige keerulisem osa oli analüüsi faas, et leida ajakohast informatsiooni aktsiaraamatute avaldamise kohta nii kohalikul kui välisturgudel. Lisaks pakkus head väljakutset kasutajaliidese disaini koostamine, sest autori sellealane kogemus on pigem tagasihoidlik. Tänu valitud Blazor tehnoloogiale valmis nõuetele igati vastav rakendus mis on jätkusuutlik ja kergelt edasiarendatav.

## 7. Edasiarendused

Peamised edasiarenduse võimalused on klientrakenduse täiendused. Suurim arendus on lisada aktsionäri vaatesse emitendi hinna graafik ja lisada graafikule tehtud tehingud. See annab hea ülevaate sellest, kuidas on hind ajas muutunud vastavalt aktsionäri tehtud otsustele. Teise olulise muudatusena tuleks klientrakendusse realiseerida aktsiate tükelduse loogika, et oleks selgelt näha olukord kui on toimunud tükeldus ja kuidas on aktsiate arv tänu sellele muutnud. Vajalik on ka täiendus, et oleks näha millises nimekirjas on emitent hetkel või olnud ka minevikus. See annaks ka võimaluse näha, et mis kuupäeval on emitent börsile tulnud või lahkunud.

Kogutud andmete põhjal on võimalik koostada ka erinevat statistikat, nagu viimase päeva/nädala/kuu suurimad liikumised, keskmine muutuste arv päevas jms. Hetkel on rakendus kasutatav vaid eesti keeles, lähiajal on lisandumas ka inglise keele tugi.

Lisaarendustena võib tekkida vajadus aktsionäri nime pseudonümiseerimiseks, API veebiteenusteks või arendusteks juhul kui registripidaja otsustab muuta kasutajaliidest või turvakoodi lahendust nii, et aktsiaraamatute uuendamine on häiritud. Arenduste nimekirjas on ka teavituste süsteem, et oleks võimalik tellida teavitus juhul kui huvipakkuva aktsionäri seis on mõnes aktsiaraamatus muutunud.

## 8. Kokkuvõte

Bakalaureusetöö eesmärgiks oli arendada rakendus Nasdaq Tallina börsi aktsiaraamatute väljavõtteks, mis võimaldab mugavalt ja kiirelt saada ülevaade nii emitentide kui ka aktsionäride muutustest läbi aja erinevates aktsiaraamatutes.

Lõputöö eesmärgi saavutamiseks kaardistati olemasolev probleem ja aktuaalsus ning pandi paika loodava rakenduse nõuded. Rakenduse nõuded kaardistati suheldes Tallinna börsi investoriga kes on olnud vaadeldava probleemiga kimpus juba aastaid. Lõputöö praktilist teostust kirjeldav peatükk andis ülevaate süsteemi struktuurist, arendusprotsessist kui ka kasutatud tehnoloogiatest.

Töö käigus valmis nõuetele vastav lahendus ja kõik püstitatud eesmärgid said täidetud. Serverrakendus kogub peale igat börsipäeva registripidaja veebilehelt aktsiaraamatute muudatused ning salvestab need andmebaasi, kust klientrakenduse kaudu need nähtavaks saavad. Loodud rakendus on hea põhi edasiarenduseks ja andmete põhjalikumaks analüüsiks.

## Kasutatud kirjandus

- [1] Nasdaq „Tallinna börsi statistika“, [Võrgumaterjal]. Available: <https://nasdaqbaltic.com/statistics/et/statistics> [Kasutatud 18 veebruar 2023].
- [2] Kaarel Ots „100 000 investorit!“, [Võrgumaterjal]. Available: <https://www.kaarelots.ee/single-post/100-000>. [Kasutatud 12 veebruar 2023].
- [3] Kaarel Ots „Börsiaasta kokkuvõte 2022“, [Võrgumaterjal]. Available: <https://www.kaarelots.ee/single-post/b%C3%B6rsiaasta-kokkuv%C3%B5te-2022> [Kasutatud 12 veebruar 2023].
- [4] Investopedia: „Getting to Know the Stock Exchanges“, [Võrgumaterjal]. Available: <http://www.investopedia.com/articles/basics/04/092404.asp> [Kasutatud 12 veebruar 2023].
- [5] Riigi Teataja, Väärtpaberite registri pidamise seadus, [Võrgumaterjal]. Available: <https://www.riigiteataja.ee/akt/104122019018?leiaKehtiv#para7lg1>. [Kasutatud 01 veebruar 2023].
- [6] „9 Most Important Functions of Stock Exchange/Secondary Market“, [Võrgumaterjal]. Available: <http://www.yourarticlelibrary.com/economics/market/9-most-important-functions-of-stock-exchangesecondary-market/8766/> [Kasutatud 12 veebruar 2023].
- [7] Nasdaq „Börsiettevõtete aktsiaraamatud Internetis“ 04.09.2001, [Võrgumaterjal]. Available: <https://nasdaqbaltic.com/et/news/borsiettevotete-aktsiaraamatud-internetis>. [Kasutatud 02 veebruar 2023].
- [8] Nasdaq CSD SE Eesti hinnakiri, [Võrgumaterjal]. Available: [https://nasdaqcsd.com/wp-content/uploads/010123-Nasdaq\\_CSD\\_hinnakiri\\_register\\_EE.pdf](https://nasdaqcsd.com/wp-content/uploads/010123-Nasdaq_CSD_hinnakiri_register_EE.pdf). [Kasutatud 04 veebruar 2023].
- [9] Nasdaq Tallinna börs, [Võrgumaterjal]. Available: <https://nasdaqbaltic.com/et/meist/nasdaq-tallinna-bors> [Kasutatud 18 veebruar 2023].
- [10] Nasdaq Balti börsid, [Võrgumaterjal]. Available: <https://nasdaqbaltic.com/et/meist/nasdaq-balti-borsid>. [Kasutatud 12 veebruar 2023].
- [11] „Investeerimise teejuht“, Äripäev, 2007 – LK 34, [Kasutatud 12 veebruar 2023].
- [12] „Development of Capital Markets and its Institutions in Estonia“, Alar Kein – LK 17, [Võrgumaterjal]. Available: <https://www.nato.int/acad/fellow/96-98/kein.pdf> [Kasutatud 12 veebruar 2023].
- [13] Nasdaq balti börsi turuinfo, [Võrgumaterjal]. Available: <https://nasdaqbaltic.com/et/turuinfo> [Kasutatud 18 veebruar 2023].
- [14] Nasdaq balti börsi turukapitalisatsioon 17.02.2023 seisuga, [Võrgumaterjal]. Available: <https://nasdaqbaltic.com/statistics/et/capitalization?filter=1&preset=d&start=2023-02-17&end=2023-02-17> [Kasutatud 18 veebruar 2023].
- [15] USA börsi turukapitalisatsioon 17.02.2023 seisuga, [Võrgumaterjal]. Available: <https://companiesmarketcap.com/usa/largest-companies-in-the-usa-by-market-cap> [Kasutatud 18 veebruar 2023].

- [16] Tallinna börsi kauplemispühad, [Võrgumaterjal]. Available: <https://nasdaqbaltic.com/statistics/et/calendar?filter=1&period=&from=2023-01-01&to=2023-12-31&category=3&issuer=&exchanges%5B%5D=XTAL> [Kasutatud 18 veebruar 2023].
- [17] Nasdaq CSD SE Eesti filiaali e-Register, [Võrgumaterjal]. Available: <https://ereg.nasdaqcsd.com>. [Kasutatud 04 veebruar 2023].
- [18] Juhend väärtpaberituru seaduse §-s 188-13 sätestatud tehingutest teavitamise kohustuse täitmiseks, [Võrgumaterjal]. Available: <https://www.fi.ee/et/juhendid/investment-market/juhend-vaartpaberituru-seaduse-ss-s-188-13-satestatud-tehingutest-teavitamise-kohustuse-taitmiseks>. [Kasutatud 05 veebruar 2023].
- [19] Riigi Teataja, Väärtpaberituru seadus, [Võrgumaterjal]. Available: <https://www.riigiteataja.ee/akt/13148804?leiaKehtiv>. [Kasutatud 05 veebruar 2023].
- [20] Juhtimiskohustusi täitvate isikute ja nendega lähedalt seotud isikute tehingutest teavitamine, [Võrgumaterjal]. Available: <https://www.fi.ee/et/investeerimine/registrid-ja-nimekirjad/juhtimiskohustusi-taitvate-isikute-ja-nendega-lahedalt-seotud-isikute-tehingutest-teavitamine>. [Kasutatud 05 veebruar 2023].
- [21] Finantsinspektsioon „Börsiemitentidega seotud isikute tehingute register“ [Võrgumaterjal]. Available: <http://oam.fi.ee/et/transaction-list> [Kasutatud 22 aprill 2023].
- [22] Elisa Corporation aktsionäride nimekiri [Võrgumaterjal]. Available: <https://elisa.com/corporate/investors/shareholders> [Kasutatud 03 aprill 2023].
- [23] Telia Company aktsionäride nimekiri [Võrgumaterjal]. Available: <https://www.teliacompany.com/en/investors/share-related-information/shareholdings> [Kasutatud 03 aprill 2023].
- [24] „Access to Shareholder Lists: Who and How?“ [Võrgumaterjal]. Available: <https://www.upcounsel.com/lectl-access-to-shareholder-lists-who-and-how> [Kasutatud 03 aprill 2023].
- [25] U.S. Securities and Exchange Commission „Shareholder Lists, When You Can Get Them“ [Võrgumaterjal]. Available: <https://www.investor.gov/introduction-investing/investing-basics/glossary/shareholder-lists-when-you-can-get-them> [Kasutatud 03 aprill 2023].
- [26] Riigikohtu kohtulahend 3-3-1-85-15, [Võrgumaterjal]. Available: <https://www.riigiteataja.ee/kohtulahendid/fail.html?id=206132758>. [Kasutatud 06 veebruar 2023].
- [27] “Functional vs Non Functional Requirements”, [Võrgumaterjal]. Available: <https://www.geeksforgeeks.org/functional-vs-non-functional-requirements> [Kasutatud 26 veebruar 2023].
- [28] “What is Non-Functional Requirement in Software Engineering?”, [Võrgumaterjal]. Available: <https://www.guru99.com/non-functional-requirement-type-example.html> [Kasutatud 26 veebruar 2023].
- [29] „Best Captcha Solving Services/APIs For Web Scraping and Automation“ [Võrgumaterjal]. Available: <https://geekflare.com/captcha-solving-services-api> [Kasutatud 13 märts 2023].

- [30] Coursera „What Is Python Used For? A Beginner’s Guide“ [Võrgumaterjal]. Available: <https://www.coursera.org/articles/what-is-python-used-for-a-beginners-guide-to-using-python> [Kasutatud 13 märts 2023].
- [31] „TensorFlow case studies“ [Võrgumaterjal]. Available: <https://www.tensorflow.org/about/case-studies> [Kasutatud 13 märts 2023].
- [32] Selenium WebDriver dokumentatsioon [Võrgumaterjal]. Available: <https://www.selenium.dev/documentation/webdriver> [Kasutatud 20 märts 2023].
- [33] Sri Manikanta Palakollu „Scrapy Vs Selenium Vs BeautifulSoup for Web Scraping“ [Võrgumaterjal]. Available: <https://medium.com/analytics-vidhya/scrapy-vs-selenium-vs-beautiful-soup-for-web-scraping-24008b6c87b8> [Kasutatud 20 märts 2023].
- [34] Apache Tika [Võrgumaterjal]. Available: <https://tika.apache.org> [Kasutatud 20 märts 2023].
- [35] Tika-Python [Võrgumaterjal]. Available: <https://github.com/chrisattmann/tika-python> [Kasutatud 20 märts 2023].
- [36] „WebAssembly Concepts“ [Võrgumaterjal]. Available: <https://developer.mozilla.org/en-US/docs/WebAssembly/Concepts> [Kasutatud 20 märts 2023].
- [37] Gary Brooks „What is Blazor, and why should you use it?“ [Võrgumaterjal]. Available: <https://fabrity.com/blog/software-engineering/what-is-blazor-and-why-should-you-use-it> [Kasutatud 20 märts 2023].
- [38] David Grace „Why You Should Use Blazor over JavaScript Frameworks to Build Your Single-Page Application“ [Võrgumaterjal]. Available: <https://www.telerik.com/blogs/why-you-should-use-blazor-over-javascript-frameworks-to-build-your-single-page-application> [Kasutatud 20 märts 2023].
- [39] Entity Framework Core [Võrgumaterjal]. Available: <https://learn.microsoft.com/en-us/ef/core> [Kasutatud 12 märts 2023].
- [40] „Learn Entity Framework Core“ [Võrgumaterjal]. Available: <https://www.learnentityframeworkcore.com> [Kasutatud 20 märts 2023].
- [41] GeeksforGeeks “Types of Databases”, [Võrgumaterjal]. Available: <https://www.geeksforgeeks.org/types-of-databases> [Kasutatud 12 märts 2023].
- [42] Sagar Bhatia „PostgreSQL vs MySQL: Everything You Need to Know in 2023“ [Võrgumaterjal]. Available: <https://hackr.io/blog/postgresql-vs-mysql> [Kasutatud 20 märts 2023].
- [43] „SQLite vs MySQL vs PostgreSQL: A Comparison Of Relational Database Management Systems“ [Võrgumaterjal]. Available: <https://www.digitalocean.com/community/tutorials/sqlite-vs-mysql-vs-postgresql-a-comparison-of-relational-database-management-systems> [Kasutatud 20 märts 2023].
- [44] MariaDB „MariaDB vs. MySQL“ [Võrgumaterjal]. Available: <https://mariadb.com/database-topics/mariadb-vs-mysql> [Kasutatud 20 märts 2023].
- [45] Joydip Kanjilal „Visual Studio vs JetBrains Rider: A Detailed Comparison“ [Võrgumaterjal]. Available: <https://codingsight.com/visual-studio-vs-jetbrains-rider-a-detailed-comparison> [Kasutatud 20 märts 2023].

- [46] TalTech Pythoni õppematerjalid „PyCharm IDE” [Võrgumaterjal]. Available: <https://pydoc.pages.taltech.ee/pycharm/pycharm.html> [Kasutatud 20 märts 2023].
- [47] Hugo Escafit “GitHub vs. GitLab vs. BitBucket” [Võrgumaterjal]. Available: <https://blog.mergify.com/github-vs-gitlab-vs-bitbucket> [Kasutatud 20 märts 2023].
- [48] SQLAlchemy “SQLAlchemy 2.0 - Major Migration Guide” [Võrgumaterjal]. Available: [https://docs.sqlalchemy.org/en/20/changelog/migration\\_20.html](https://docs.sqlalchemy.org/en/20/changelog/migration_20.html) [Kasutatud 20 märts 2023].
- [49] Lucidchart tööriist [Võrgumaterjal]. Available: <https://www.lucidchart.com/pages/> [Kasutatud 20 märts 2023].
- [50] Adam Hayes „What a Stock Split Is and How It Works, With an Example“ [Võrgumaterjal]. Available: <https://www.investopedia.com/terms/s/stocksplit.asp> [Kasutatud 20 märts 2023].
- [51] Cloudflare „How CAPTCHAs work | What does CAPTCHA mean?“ [Võrgumaterjal]. Available: <https://www.cloudflare.com/learning/bots/how-captchas-work/> [Kasutatud 26 märts 2023].
- [52] Manvi Goel, Nandini Sidana, Siddharthsamber „CAPTCHA Recognition using Convolutional Neural Network“ [Võrgumaterjal]. Available: <https://medium.com/@manvi./captcha-recognition-using-convolutional-neural-network-d191ef91330e> [Kasutatud 26 märts 2023].
- [53] Reema Oamkumar “MVC vs. Web API: Which ASP.NET technology should you use?” [Võrgumaterjal]. Available: <https://www.software-developer-india.com/mvc-vs-web-api-which-asp-net-technology-should-you-use> [Kasutatud 20 märts 2023].
- [54] Manish Kumar „Blazor server vs blazor webassembly“ [Võrgumaterjal]. Available: <https://www.zenesys.com/blazor-server-vs-blazor-web-assembly> [Kasutatud 20 aprill 2023].
- [55] IBM „What is three-tier architecture?“ [Võrgumaterjal]. Available: <https://www.ibm.com/topics/three-tier-architecture> [Kasutatud 22 aprill 2023].
- [56] Microsoft „Design the infrastructure persistence layer“ [Võrgumaterjal]. Available: <https://learn.microsoft.com/en-us/dotnet/architecture/microservices/microservice-ddd-cqrs-patterns/infrastructure-persistence-layer-design> [Kasutatud 22 aprill 2023].
- [57] Boodhoo, Jean-Paul S „Layered Architecture, Dependency Injection, and Dependency Inversion“ [Võrgumaterjal]. Available: <https://www.codemag.com/article/0705071/Layered-Architecture-Dependency-Injection-and-Dependency-Inversion> [Kasutatud 22 aprill 2023].
- [58] Blazor QuickGrid „A simple but effective data grid for Blazor“ [Võrgumaterjal]. Available: <https://aspnet.github.io/quickgridsamples> [Kasutatud 22 aprill 2023].
- [59] Jonny Larsson „MudBlazor is here to stay“ [Võrgumaterjal]. Available: <https://mudblazor.com/mud/announcements/mudblazor-here-to-stay#a-company-behind-the-mud> [Kasutatud 22 aprill 2023].



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

Mina, Holger Roosioja

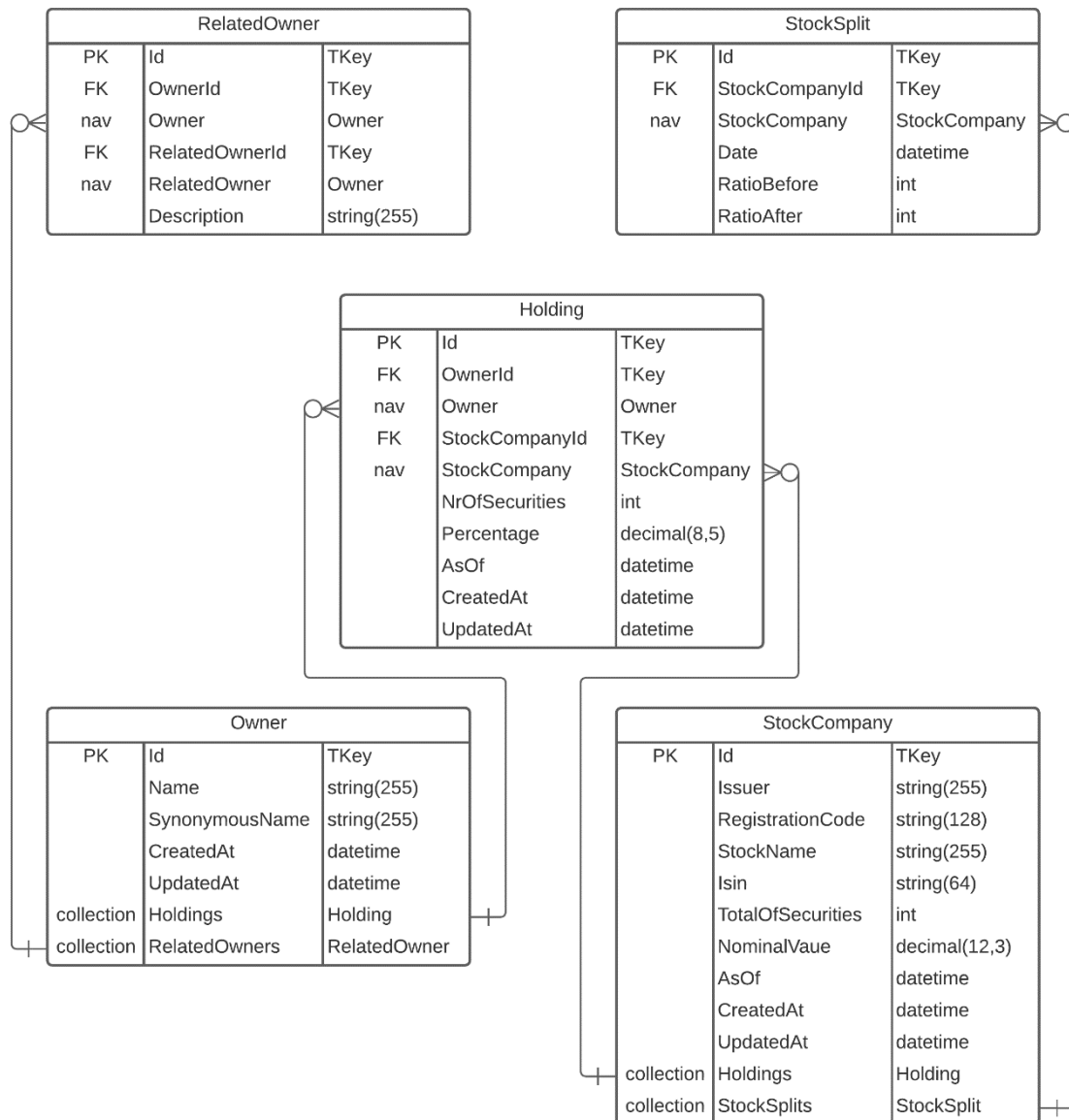
1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Nasdaq Tallinna börsil noteeritud ettevõtete aktsiaraamatute väljavõtteks vajaliku rakenduse arendamine“, mille juhendaja on Lembit Viilup
  - 1.1. reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil,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

15.05.2023

---

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

## Lisa 2 – Täielik olemi-suhte diagramm



## Lisa 3 – Kuvatõmmis registripidaja lehelt aktsiaraamatu väljavõtteks



Emitent	Enefit Green AS
Registrikood	11184032
Väärtpaberi nimi	ENEFIT GREEN AKTSIA
ISIN	EE3100137985
Väärtpaberite arv kokku	264 276 232
Nimiväärtus	1 EUR
Päeva lõpu seisuga	24.03.2023

Turvakood



Laadi alla aktsionäride nimekiri (PDF)

## Lisa 4 – Kuvatõmmis registripidaja aktsiaraamatu PDF failist



Emitent	Enefit Green AS
Registrikood	11184032
Väärtpaberite nimi	SHS ENEFIT GREEN ORD
ISIN	EE3100137985
Väärtpaberite arv kokku	264 276 232
Nimiväärtus	1 EUR
Päeva lõpu seisuga	24.03.2023

	Omaniku nimi	Väärtpaberite arv	Osalus
1.	EESTI ENERGIA AS	203 931 405	77,166 %
2.	EUROPEAN BANK FOR RECONSTRUCTION AND DEVELOPMENT	2 773 277	1,0494 %
3.	SEB AB/SAASTOPANKKI KORKO PLUS - SIJOITUSRAHASTO	1 255 056	0,4749 %
4.	SEB AB LUX BRANCH - UCITS CLIENTS	1 043 235	0,3948 %
5.	Clearstream Banking AG	1 037 945	0,3928 %
6.	CITIBANK ( NEW YORK ) / GOVERNMENT OF NORWAY	920 471	0,3483 %
7.	SEB PROGRESSIIVNE PF	828 521	0,3135 %
8.	SWEDBANKI PENSIONIFOND 1970-79 SÜNDINUTELE	766 142	0,2899 %
9.	SWEDBANK AB/ SWEDBANK INVESTICIJU VALDYMAS, UAB/ SWEDBANK PENSIIJA 1975-1981	683 034	0,2585 %
10.	SEB banka AS	674 063	0,2551 %