

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Heiko Lige 193384IABB

Portfelli põhise aktsia soovitaja rakenduse kavandamine ja arendus

Bakalaureusetöö

Juhendaja: Karl-Erik Karu
MSc

Tallinn 2024

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: Heiko Lige

19.05.2024

Annotatsioon

Bakalaureuse töö keskendub probleemidele seoses investeerimisalaste teadmistega. Kuigi raha investeerimine on muutunud lihtsamaks kui kunagi varem, tehakse seda sageli kas liiga kergekäeliselt või välditakse seda täielikult, kartes rahast ilma jääda. Käesoleva töö eesmärgiks on disainida ja luua veebirakendus, mis proovib just neid probleeme lahendada andes võimaluse kasutajatel oma portfelli ja teadmisi laiendada uute aktsiatega. Veebirakendus soovitab ja annab infot aktsia kohta, mis sobib kasutaja riskitaseme ja ülejäänud portfelliga.

Töö analüüsi osas uuritakse juba turul olevate sarnaste rakenduste kohta, pannakse paika kasutuslood ja rakendusele kehtivad nõuded. Töö praktilises osas, rakenduse loomisel, arendatakse aktsiasoovituste algoritm ja luuakse veebirakendus, mis vastab nõuetele ja kasutuslugudele, aidates saavutada töö eesmärki. Töö tulemusena hinnatakse kas rakendus on äriiselt kasulik ning kas valminud toode täidab seatud eesmärgid ning aitab lahendada püstitatud probleemi.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 26 leheküljel, 6 peatükki, 12 joonist, 3 tabelit.

Abstract

Design and development of portfolio based stock recommender application

The bachelor's thesis focuses on issues related to investment knowledge. Although investing money has become easier than ever, it is often either done too lightly or avoided altogether for fear of losing money. The goal of this work is to design and create a web application that tries to solve these problems by giving users the opportunity to expand their portfolio and knowledge with new stocks. The web application recommends and provides information about a stock that fits the user's risk level and the rest of the portfolio.

In the analysis section of the work, existing similar applications on the market are examined, and user stories along with requirements for the application are defined. In the practical part of the project, during the application development phase, a stock recommendation algorithm and a web application is created that meet the requirements and usage scenarios, helping to achieve the objectives of the work. As a result of the work, the application's commercial viability will be assessed, along with whether the developed product fulfills the set objectives and helps to solve the identified problem.

The thesis is in estonian and contains 26 pages of text, 6 chapters, 12 figures, 3 tables.

Lühendite ja mõistete sõnastik

API	<i>Application Programming Interface</i> (programmeerimise liides)
<i>Backend</i>	Teenuse- või äripool
CSS	Keel veebilehtede stiili kujundamiseks
ETF	<i>Exchange traded fund</i> (Börsil kaubeldavad fondid)
<i>Feature</i>	Arendustükk
<i>Frontend</i>	Kasutajaliides
HTML	Keel veebilehtede ülesmärkimiseks
HTTP	Hüperteksti edastusprotokoll ehk <i>Hypertext transfer protocol</i>
JSON	<i>JavaScript Object Notation</i> ehk andmevahetusvorming, mis põhineb JavaScripti programmeerimiskeele alamhulgal
NASDAQ	<i>National Association of Securities Dealers Automatic Quotation System</i> ehk USA elektrooniline aktsia turg
.NET	Microsofti avatud lähtekoodiga arvutitarkvara raamistik
REST	<i>Representational State Transfer</i>
SPA	<i>Single Page Application</i>
SQL	Microsofti avatud lähtekoodiga arvutitarkvara raamistik Struktuuripäringukeel ehk <i>Structured Query Language</i>

Sisukord

1 Sissejuhatus	10
2 Metoodika ja taust.....	12
2.1 Aktsia valiku algoritm	12
2.2 Tööriistad	12
2.3 Protsess	13
3 Analüüs.....	15
3.1 Alternatiivsed lahendused.....	15
3.2 Uue rakenduse vajadus	17
3.3 Funktsionaalsed nõuded	18
3.4 Mittefunktsionaalsed nõuded.....	18
3.5 Kasutajalood	19
4 Realisatsioon.....	22
4.1 Üldine rakenduse arhitektuur.....	22
4.2 Rakenduse kasutajaliides	23
4.3 Andmebaas ja taustatööd	25
4.4 Kontrolleri klassid	26
4.5 Testid	28
4.5.1 Ühiktestid.....	29
4.5.2 Integratsioonitestid	29
4.6 Aktsia soovitamise algoritm	29
5 Tulemus ja järeldused	32
5.1 Valminud funktsionaalsus	32
5.2 Äriline kasum.....	32
5.2.1 Kulud	32
5.2.2 Tulud.....	33
5.3 Kordaminekud ja õppetunnid.....	33
5.4 Rakenduse edasiarendus	33
6 Kokkuvõte	35

Kasutatud kirjandus	36
Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks	38

Jooniste loetelu

Joonis 1. Agiilse arenduse töövoog [12].....	14
Joonis 2. AssetRebalanceri UI [13]	16
Joonis 3. Lõppkasutaja töövoog	20
Joonis 4. Rakenduse arhitektuur	22
Joonis 5. Erinevused Single page application ja Multiple page application vahel [22] .	23
Joonis 6. App.Vue.....	24
Joonis 7. Kasutajaliidese näide "Home" vaade	24
Joonis 8. Hangfire töövoog [24]	25
Joonis 9. Aktsia kontrolleri ja aktsia soovitamise meetod	27
Joonis 10. Nswag abil genereeritud kood SuggestStocks meetodist	28
Joonis 11. Rakenduse testide kaetus	28
Joonis 12. Aktsia soovitamise algoritm	30

Tabelite loetelu

Tabel 1. Alternatiivsete rakenduste võrdlus	17
Tabel 2. protsesside kirjeldused.....	20
Tabel 3. Aktsiate andmebaasi tabel	26

1 Sissejuhatus

Iga aastaga muudetakse investeerimist ühe lihtsamaks ning kättesaadavamaks huvilistele tänu uutele platvormidele, rakendustele ja tehnoloogiatele [1]. Tänapäeval on võimalik igapäev, kellel väheke tahtmist ja raha, osta endale murdosa aktsiast või veel väiksem osake krüptorahast ning kõike seda kodust lahkumata. Jälgides Eesti ja ülejäänud Euroopa viimaste aastate inflatsioonimäärasid, näib raiskamine hoida oma säästud seisvana pangakontol. Euroopas on viimase kahe aasta jooksul olnud keskmine inflatsioonimäär 7% (2022-2024 jaanuar) [2], mis on ilmselget tavalisest inflatsioonist kõrgem, kuid näitab, et iga modernne inimene peaks ühel või teisel viisil investeerima, et neile kuuluva raha väärtus ei kaoks inflatsiooni tagajärjel.

1.1 Taust ja probleem

Üha rohkem on tänapäeva meedias juttu krüptorahadest, uutest investeerimisvõimalustest ja kiiresti rikkaks saamise viisidest, kuid vähem räägitakse nendega kaasnevatest riskidest. Investeerimisplatvormidel on lihtne mõne liigutusega osta aktsiaid ja krüptot, kuid kui sellega kaasnevaid tulusid või riske vaja arvutada, peab kasutaja minema kolmanda osapoole lehekülgedele, millest keskmine investeerija midagi aru ei saa. Investor võiks saada ilma spetsiaalse hariduseta koostada endale balansseeritud portfoolio [3], kus investeeringud on jaotatud laiali erinevate tööstusvaldkondade vahel, kasutades matemaatilisi näitajaid, et minimeerida riski ning maksimeerida tulu. Suur osa inimestest jätavad investeerimata, kuna neil puuduvad vastavad teadmised või nad kardavad oma raha kaotada [4].

1.2 Eesmärgid

Antud töö eesmärk on luua veebirakendus, mille abil on võimalik luua tasakaalustatud investeerimisportfell vastavalt oodatavale tulule ja riskile, mis oleks lihtsasti kasutatav ning kättesaadav kõigile. Kasutades modernse portfelliteooria [5] teadmisi ja aktsiate beetat [6] suudab süsteem soovitada kasutajale aktsiaid, mis balansseeriks nende portfooliot samal ajal hoides tahetud riski ja tulu suhet. Eesmärgi saavutamiseks on töö

jagatud järgmisteks alametappideks: Eesmärgi saavutamiseks on töö jagatud järgmisteks alametappideks:

- Sarnaste rakenduste kaardistamine ja analüüs koos võrdlusega
- Portfelliteooria uurimine kirjandusest
- Rakenduses kasutatava vajamineva arhitektuuri analüüs
- Rakenduse arendamine ja testimine
- Kasutajate tagasiside põhjal tulemuste valideerimine

Eesmärk ei ole luua kasutajale võimalikult suurt tulu, vaid informeerida inimesi, kuidas võiks aktsiaid valida ning portfelli koostada. Peale rakenduse kasutamist on kasutaja loodetavasti saanud lisa informatsiooni ning investeerib aktsiaturul madalama riski ja kergema südamega.

1.3 Töö struktuur

Käesolev bakalaureusetöö koosneb kuuest põhiosast: sissejuhatus, metoodika, analüüs, realisatsioon, tulemus ja järeldused ning kokkuvõte. Sissejuhatuses tutvustatakse tööd, probleeme selle eesmärgi. Metoodikas kirjeldatakse arendus metoodikat, kasutatavaid tööriistu ning kuidas eesmärgini jõuda soovitakse. Analüüsi osas uuritakse alternatiivseid lahendusi nende tugevaid ja nõrku külgi, pannakse paika rakenduse nõuded ja kasutajalood ning uuritakse rakenduse vajalikkust ühiskonnas. Neljandas osas ehk realisatsioonis kirjeldatakse kuidas projekt arendati ning tulemuseni jõuti. Tulemus ja järeldus peatükis tuuakse välja ärilised tulemused ning potentsiaalsed edasiarenduse võimalused. Töö viimases osas võetakse kokku kogu projekt ning kas jõuti oodatud tulemuseni.

2 Metoodika ja taust

Käesolevas peatükis antakse ülevaade bakalaureuse töös kasutatavatest tööriistadest, metoodikast, teoreetilisest taustast, töö protsessidest ning miks neid kasutati.

2.1 Aktsia valiku algoritm

Optimaalse aktsiasoovituse pakkumiseks on vajalik välja töötada algoritm, mis arvestab kasutaja sisestatud andmetega. Algoritmi väljatöötamisel tuleb tähelepanu pöörata mitmetele olulistele aspektidele: lõppeesmärk, andmete kättesaadavus ja nende ajakohasus. Aktsiaturg on kiiresti muutuv keskkond, mistõttu on kriitiline tagada juurdepääs ajakohastele andmetele, et vältida aegunud teabe kasutamist. Antud lõputöös valiti peamiseks andmeallikaks aktsiate beeta, mis aitab arvesse võtta turu volatiilsust ja riske [7]. Aktsia beeta näitab kui volatiilne see on võrreldes turu keskmisega. Antud lõputöö puhul on turu keskmiseks võetud S&P 500 indeksi fond [8]. Beeta, mis on alla ühe, on võrreldes turuga vähem volatiilne ning seega ka madalama riskiga ja vastupidiselt kui beeta on üle ühe. Lõputöö käigus arendatud programm soovib kasutajale aktsiat vastavalt soovitud riskile (beetale) ning tema praegusele investeerimis portfelligile. Portfelli beeta [9] arvutatakse liites kokku kõikide portfellis olevate aktsiate kohandatud beetad (aktsia osa portfelist * aktsia beeta). Kui kasutaja soovitud riski tase on üsna kõrge, näiteks beeta on 1.3 ning tema portfelli praegune beeta on 1, siis talle soovitatakse aktsiat, mille ostmisel portfelli beeta oleks kõige lähemal 1.3-le. Algoritmi on võimalik veelgi täpsemaks teha tuues sisse lisa tegureid nagu näiteks P/E suhe, ennustatav aktsia hind jne, kuid käesoleva töö raames ei olnud selliseid andmed usaldusväärset kättesaadavad.

2.2 Tööriistad

Projekti arendamiseks jaotati projekt kaheks: kasutajaliides (*frontend*) ja äri loogika ning teenused (*backend*). Mõlema puhul valiti arenduskeskkonnaks (IDE'ks) *JetBrains Rider*, kuna töö autoril on sellega kõige rohkem kogemust ning see on kiire ja lihtsasti kasutatav.

Äriloogika ja kasutajaliidese omavaheliseks suhtluseks kasutatakse Rest (*Representational State Transfer*) API (*Application Programming Interface*), kuna need on paindlikud ning toetavad kliendi ja serveri eraldamist täpselt nagu antud projektis vaja läheb [10].

2.2.1 Kasutajaliides

Kasutajaliidese arendamiseks kasutatakse HTML-i (*HyperText markup language*), CSS (*Cascading style sheets*) ja Vue 3 kooslust, kuna neil kõigil on väga hea dokumentatsioon ning autoril on sellega varasemaid kogemusi. Lisaks kasutatakse veel TypeScripti, mille abil saame tekitada tugevalt tüübitud objekte, mis omakorda aitab leida Vue koodis vigu ning hoiab koodibaasi puhtamana ja loetavamana.

2.2.2 Äriloogika

Äriloogika poolel kasutatakse C# keelt, mis võimaldab kasutada .NET raamistikku. Spetsiifilisemalt kasutatakse .NET Core 7, sest see on töö kirjutamise ajal kõige kaasaegsem, sellel on suurepärase dokumentatsioon ning kuna see on (*open source*), ehk igal inimesel on õigus seda muuta kuidas iganes, on sellel väga palju kasulikke tööriistu, mis on arendajatele suureks abiks [11]. Lisaks kasutatakse Hangfire, et automaatselt käivitada ja hallata taustal vajaminevaid töid. Äriloogika testimiseks kasutatakse NUnit raamistikku, mis lubab mugavalt ja lihtsalt ühik teste kirjutada, et tulevikus rakenduse muutmisel kõik tööle jääks.

2.2.3 Andmebaas

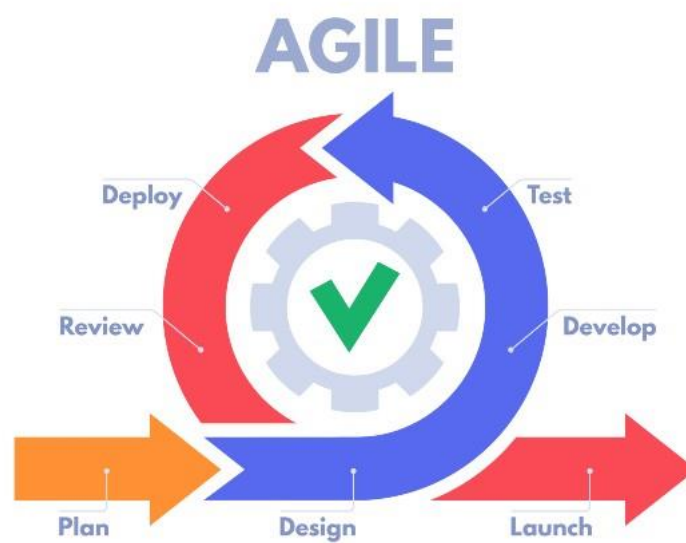
Andmebaasiks valiti Microsoft SQL (edaspidi MSQl) server, kuna tegu on relatsioonilise andmebaasiga, mis suudab JSON päringutega ümber käia ning seda on väga lihtne üles seada ja hallata tänu SQL *Server Management Studio*le. Suur põhjus miks just MSQl valiti on kuna see on Hangfire poolt toetatud.

2.3 Protsess

Kuna projektiga tegeleb ainult üks arendaja, siis kasutatakse agiilset tarkvara arendamise põhimõtteid ehk arendatakse pisikeste juppide (*feature*) kaupa, mis kokku moodustavad lõpuks ühe terviku. See võimaldab teha projektis jooksvalt muudatusi ning soodustab

paindlikkust. Lisaks võimaldab see ka kasutajatel üsna varakult testida ning vajadusel muudatusi lihtsasti sisse viia.

Agiilne arenduse töövoog on justkui ring (Joonis 1) kus liigutakse etapist etappi, kuid vajadusel minnakse tagasi algusesse. Selle projekti arendus käis järgmiselt: planeerimine + disain ehk kasutajaloo tegemine ja kavandamine, seejärel arendus, testimine ning lõpuks lükatakse tulemus Github'i üles. Kui näiteks peale testimist leitakse mingi viga on täiesti realistlik, et minnakse tagasi planeerimise etappi ning muudetakse kasutajalugu ümber.



Joonis 1. Agiilse arenduse töövoog [12]

Kui üks tükk rakendusest on valmis, siis alustatakse nõ „ringi“ uuesti uue funktsionaalsuse arendamisega.

3 Analüüs

Selles peatükis kirjeldatakse rakenduse peamisi nõudeid, kasutajalugusid ja rakenduse töövoogu. Lisaks analüüsitakse turul olevaid alternatiivseid lahendusi ning nende erinevusi selle projektiga.

3.1 Alternatiivsed lahendused

Nõuete ja kasutajalugude paika panemiseks tuleb esmalt kaardistada turul olemasolevaid rakendusi, neid analüüsida ning vaadata kas käesoleval projektil on üldse vajadust. Kaardistuse käigus tuli esile kolm rakendust.

AssetRebalancer

AssetRebalancer [13] on veebileht, mis võtab sisse kasutaja praeguse portfelli ja soovitud summa, mida investeerida. Kasutajale näidatakse tema varade praegust jaotust ning ta saab märkida enda soovitud jaotuse, pärast mida näitab rakendus milliseid tehinguid on vajalikud, et soovitud varade jaotus realiseerida. Erinevalt sellest projektist, saab AssetRebalanceris ringi käia ainult *ETF-idega* (Börsil kaubeldavad fondid) ning see ei soovita uut aktsiat või fondi vaid näitab kuidas hetkel hoitavate fondidega ümber

mängida, et saaks soovitud tulemuse. Veebilehe kasutajaliides (joonis 3) jääb esmasel kasutusel segaseks ning vajab süvenemist, et rakendusest tähendusrikast kasu saada.

Target

What are your asset allocation targets?

Asset Type	Percentage
US Stocks	<input type="range" value="34.7"/> 34.7 %
US Bonds	<input type="range" value="65.3"/> 65.3 %

Cash Changes

Are you depositing or withdrawing any money?

Depositing
 Withdrawing

\$ 2000

Factoring in \$2,000.00 of additional investments.

Trades

You can rebalance your portfolio with 2 trades:

USD → BSV
\$1,306.60
Buy \$1,306.60 of BSV using cash.

USD → IVOO
\$693.40
Buy \$693.40 of IVOO using cash.

Joonis 2. AssetRebalanceri UI [13]

Kuigi sellel töö on üsna palju sarnasusi AsserRebalanceriga, siis on veebilehede eesmärk ja rakendus vägagi erinev.

WealthFront

WealthFront [14] on automatiseeritud investeerimis portfelli haldur, mis vastavalt riski soovile haldab kasutaja portfelli ning jagab ressursse laiali või müüb/ostab. Erinevalt selle töö rakendusest, mis ainult soovitab aktsiaid, teeb WealthFront tehingud ise ära. Kahjuks on WealthFront saadaval ainult USA sotsiaalkindlustuse numbriga kodanikele ning see on tasuline.

Betterment

Betterment [15] on portfelli haldamise platvorm, kus sarnaselt WealthFront'ile on võimalus kasutada automatiseeritud investeerimist, müümist, tasakaalustamist ja palju

muud. Betterment teeb investori jaoks kõik liigutused ise ära, mistõttu ei saa kasutaja ise mitte mingit infot või õppimiskogemust, millega tulevikus paremaid finants otsuseid langetada. Ka see rakendus on saadaval ainult USA kodanikele ning on tasuline.

Nagu tabelist 1 näha, on igal rakendusel oma plussid ja miinused ning hetkel ei ole turul ühtegi rakendust, mis oleks täpselt sarnane.

Tabel 1. Alternatiivsete rakenduste võrdlus

	AssetRebalancer	WealthFront	Betterment	Uus rakendus
Tasuline	Ei	Jah	Jah	Ei
Kättesaadavus	Terve maailm	USA	USA	Terve maailm
Aksia soovitamise	Ei	Jah	Jah	Jah
Kasutaja saab kasulikke infot	Jah	Ei	Ei	Jah

3.2 Uue rakenduse vajadus

Nähtavasti on turul olemas sarnaseid lahendusi, mis lähevad käesoleva töö võimekusest sammu võrra edasi. Investeerimine ei ole lihtsalt suvaliselt kellegi kätte raha andmine, vaid informeeritud otsuste tegemine ja kalkuleeritud riskite vastu võtmine, et teenida võimalikult madala riskiga võimalikult suurt tulu. Finants haritus ei ole tähtis ainult investoritel turul, vaid inimeste igapäeva elus tähtsate valikute tegemisel nagu näiteks: pensionifondi, pangalaenu, tankla või poe sooduskaardi valimisel. [16]

Lõputöö rakenduse eesmärk on pigem informatiivne ehk kasutajad saavad otsida enda portfelliga sobivaid aktsiaid ning neid edasi uurida ja teha informeeritud ostu või müügi otsus. 2015. aastal läbiviidud uuringu [17] tulemustel selgus, et finantskirjaoskus on tähtis osa majandusliku stabiilsuse ja kasvu saavutamisel. Lisaks selgus ka, et vähem arenenud maades on finantskirjaoskus madalam kui arenenud maades, kus inimestel on rohkem raha, et maksta tasuliste veebilehtede nagu WealthFront ja Betterment. Uuringu tulemuste alusel tundub, et on vajadus tasuta rakendustele, mis on kergesti seeditavad ning algajasõbralikud, et inimesed saaksid teha paremaid finants otsuseid nii aktsiaturul kui ka päris elus. Sellest tulenevalt valmisid lõputöö rakenduse nõuded

3.3 Funktsionaalsed nõuded

Järgmisena on välja toodud minimaalsed funktsionaalsed nõuded, millele rakendus peab vastama.

1. Kasutaja saab valida NASDAQ [18] turul olevate aktsiate vahel.
2. Aktsiate info uueneb iga 24 tunni tagant.
3. Kasutaja saab valida endale sobiva riskitaseme.
4. Kasutajale soovitatakse vähemalt 3 aktsiat vastavalt portfelliile ning riskitasemele.
5. Kasutaja viidatakse lehekülgedele, kus saab soovitatava aktsia kohta lisa infot.

3.4 Mittefunktsionaalsed nõuded

Järgmisena on välja toodud minimaalsed mittefunktsionaalsed nõuded, millele rakendus peab vastama.

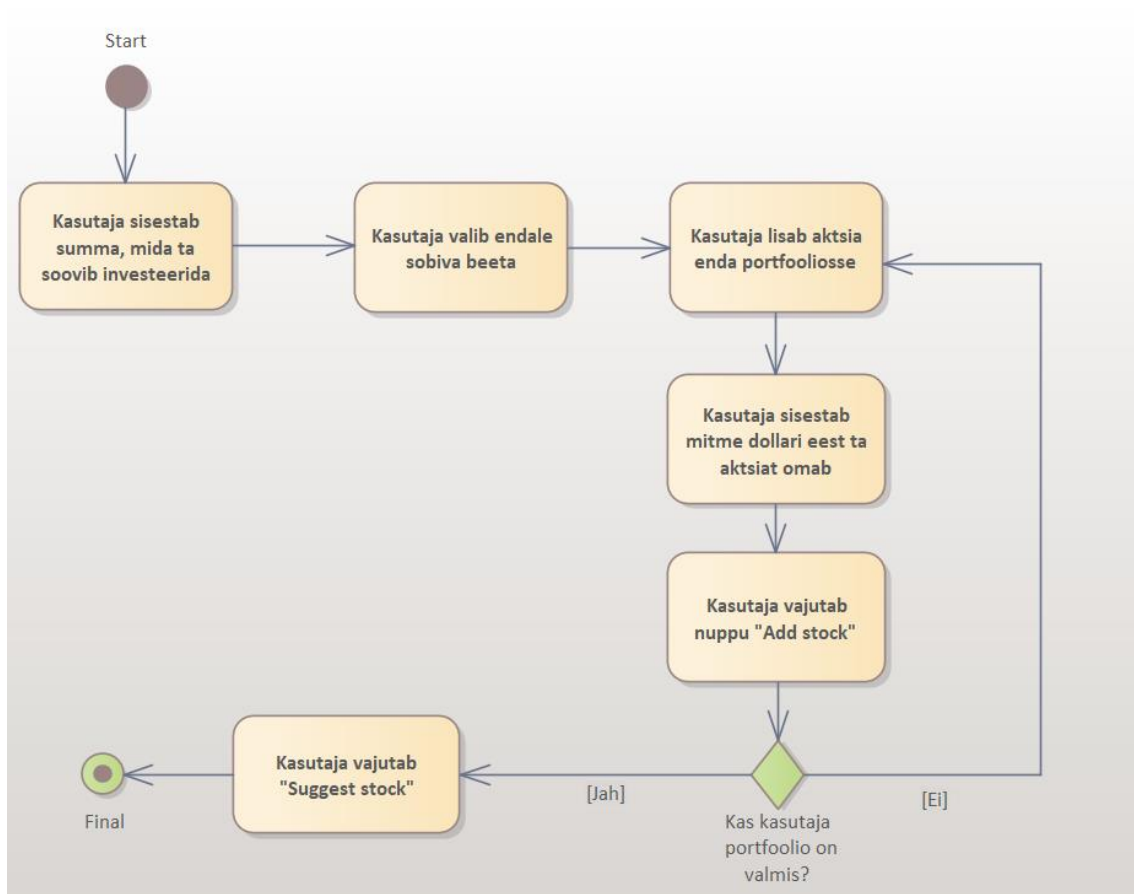
1. Rakendust peaks olema lihtne ja loogiline kasutada. Kasutaja peaks ilma lisa abita suutma kõiki rakenduse funktsionaalsusi kasutada. Seda peavad toetama rakenduse kasutajaliides ja kasutajakogemus.
2. Rakendus peab suutma algoritmiga abiga leidma korrektsed aktsiad vähem kui 5 sekundiga.
3. Aktsiate uuendamisel ei tohiks rakendus lõpetada töötamist. Rakendus peab töötama kui andmebaasi uuendatakse uute aktsiatega.
4. Rakenduse kood peaks olema loetav, hallatav ning testitud. Testide kattuvus peaks olema vähemalt 80% vastavalt headele tavadele [19]
5. Kasutajaliidese keeleks on selle töö raames inglise keel, kuna see on rahvusvaheline keel.
6. Kõik kasutatavad hinnad ja valuutad on USA dollarites, kuna NASDAQ turul käib enamasti kauplemine dollarites.

3.5 Kasutajalood

Vastavalt agiilse arendamise põhimõtetele jupitatakse rakenduse arendus väikesteks kasutajalugudeks. Järgnevalt on kirjas peamised kasutajalood, mille alusel rakendust arendatakse. Kasutatud on Atlassiani agiilse projektijuhtimise kasutajalugude malli [20], mis defineerib lood kui: kes mida tahab mille jaoks.

1. Mina kasutajana tahan valida aktsiaid kaasaegse infoga, et enda portfelli luua.
2. Mina kasutajana soovin aktsiaid otsida otsingu riba abil, et neid kergemini üles leida.
3. Mina kasutajana soovin oma riskitaluvust märkida, kas olemasolevate nuppudega nagu nt „High risk” või siis ise sisestades beeta 0-2.
4. Mina kasutajana soovin, et nupule „Suggest Stocks” vajutades soovitab rakendus kolme erinevat aktsiat
5. Mina kasutajana soovin, et soovitatavatel aktsiatel oleks lisaks link, mis viitab vastava aktsia Yahoo Finance lehele.
6. Mina kasutajana soovin, et kõikidel sisenditel oleks valideerimine.

Kasutajalugudest tulenevalt tekib rakenduse lõppkasutaja töövoog (Joonis 3).



Joonis 3. Lõppkasutaja töövoog

Vastavalt töövoole saame protsessid ning nende kirjeldused (Tabel 2).

Tabel 2. protsesside kirjeldused

Kirjeldus	Eeltingimus	Järeltingimus
Kasutaja sisestab kulutatava raha koguse	Kasutaja on „Money to spend“ sisestusriba juures. Sisestatud arv peab olema suurem kui 0	Kulutatav raha kogus on sisestatud
Kasutaja valib soovitud beeta	Kasutaja on „Desired beta“ sisestusriba juures. Sisestatud arv peab olema suurem või võrdne nulliga	Soovitud beeta on sisestatud
Kasutaja lisab enda aktsia portfooliosse	Kasutaja on „Search for a stock“ otsinguriba juures. Kasutaja sisestab vähemalt 1 tähe otsinguks.	Kasutaja saab valida 1 aktsia valikust, mis koosneb aktsiatest, mille nimes on otsinguribale sisestatud tähed

Kirjeldus	Eeltingimus	Järeltingimus
Kasutaja sisestab mitme dollari eest ta eelnevalt valitud aktsiat omab	Kasutaja on „Amount owned“ sisestusriba juures. Sisestatud arv peab olema suurem kui 0	Omatud aktsia kogus on siestatud
Kasutaja vajutab nuppu „Suggest Stock“	Kõik eelnevad väljad peavad olema korrektselt täidetud. Vähemalt 1 aktsia on praegusesse portfooliosse lisatud	Kuvatakse soovitatav aktsia koos lingiga, mis annab rohkem infot aktsia kohta

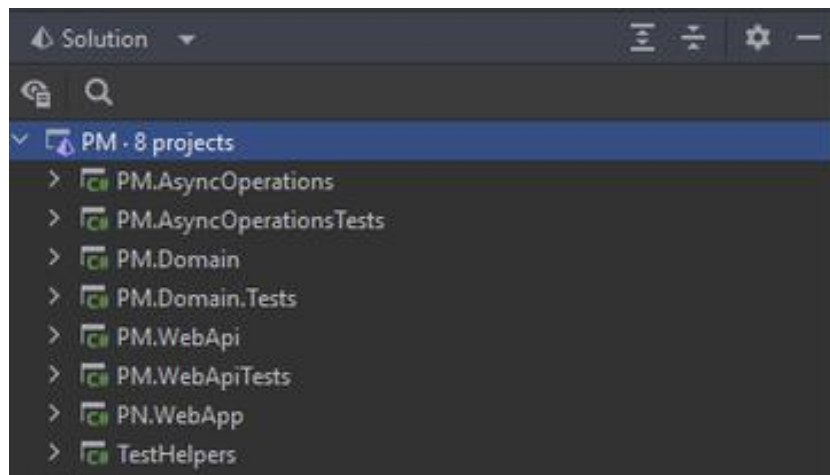
Kasutajal ei lubata sisestada ebakorrektsed sisendeid ning ka „Suggest stock“ nuppu ei saa enne vajutada kui kõik väljad on täidetud. Protsesside lihtsus ja vähesus toetavad varasemalt kirjeldatud nõudeid, et hoida rakendus minimalistlikuna ja kasutajasõbralikuna.

4 Realisatsioon

Selles peatükis antakse ülevaade koodibaasist ja rakendusest endast. Tutvustatakse aktsia valmise algoritmi, kuidas saavutati lõppeesmärk ning viimaseks vaadatakse kuidas on rakendus testitud.

4.1 Üldine rakenduse arhitektuur

Rakenduse kood on jagatud laiali nelja erinevasse klasside kogumikku (Joonis 4), kus igal kihil on oma ülesanne. Kõige tähtsam on neist domeeni kiht, kus asub andmebaasi kontekst ning selle mudelid. Lisaks on seal ka välis API-de teenused ja mudelid. *WebApi* kihis on nõ taustaprogramm, mis koosneb kontrollieritest ja vaate mudelitest. See kiht suhtleb andmebaasiga, et luua vajalikud mudelid mida kasutajaliidese kihis lõppkasutajale näidata. Eraldi kiht on veel *AsyncOperations*, kus toimuvad automaatselt käivituvad taustatööd. Viimaseks on rakenduse kasutajaliidese kiht, kus asuvad kõik vaated ja nende kuvamiseks vajalikud komponendid.

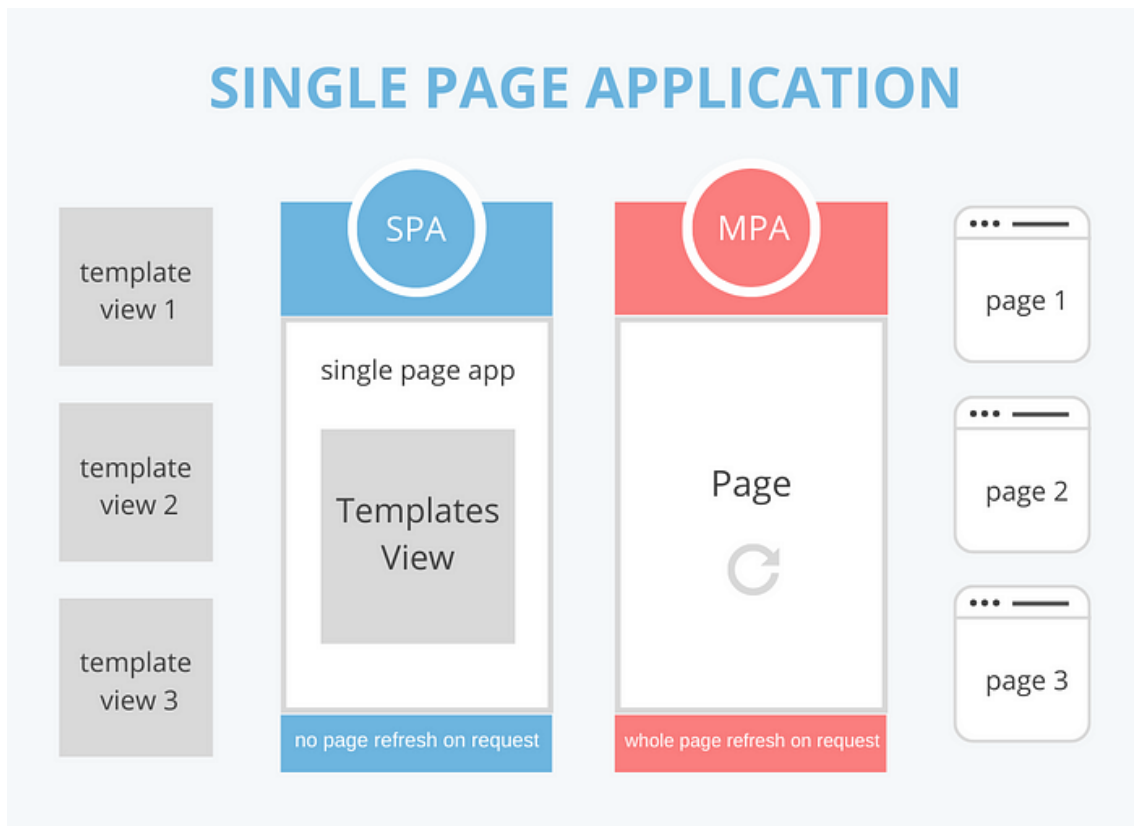


Joonis 4. Rakenduse arhitektuur

Lisaks on igal kihil ka omad testide kogumid ja testimiseks vajaminevad abistajad *TestHelpers* näol.

4.2 Rakenduse kasutajaliides

Rakendus on ehitatud kui üheleheline rakendus ehk SPA (*Single page application*) [21], mis tähendab, et kasutajale *renderdatakse* esialgu üks ainus leht ning kuvab vastavalt vajadusele esialgse *javascriptiga* uusi lehtesid. Võrreldes näiteks mitme lehe rakendustega (Joonis 5), pakuvad ühekordselehe rakendused kiiremat lehtede laadimist, vähendatud päringute arvu ning minimaalset vajadust lehti värskendada.



Joonis 5. Erinevused Single page application ja Multiple page application vahel [22]

Selles projektis on üks peamine leht App.Vue (Joonis 6), millel on kaasas lehtede navigatsioon ning *RouterView*, mis renderdab vajaminevaid vaateid.

```

<template>
  <header>
    <div class="wrapper">
      <div class="component-container">
        <Welcome />
      </div>

      <nav>
        <RouterLink to="/">Home</RouterLink>
        <RouterLink to="/about">About</RouterLink>
      </nav>
    </div>
  </header>

  <RouterView />
  
</template>

```

Joonis 6. App.Vue

Welcome to Portfolio Manager!

Here you can learn about managing your stock portfolio with stocks on the **NASDAQ** market.

[Home](#) | [About](#)



My portfolio

\$ Money to spend

Desired beta

Suggest Stocks

Add stocks to your portfolio

Search for a stock

Type to search...

Amount Owned(\$):

Enter amount owned...

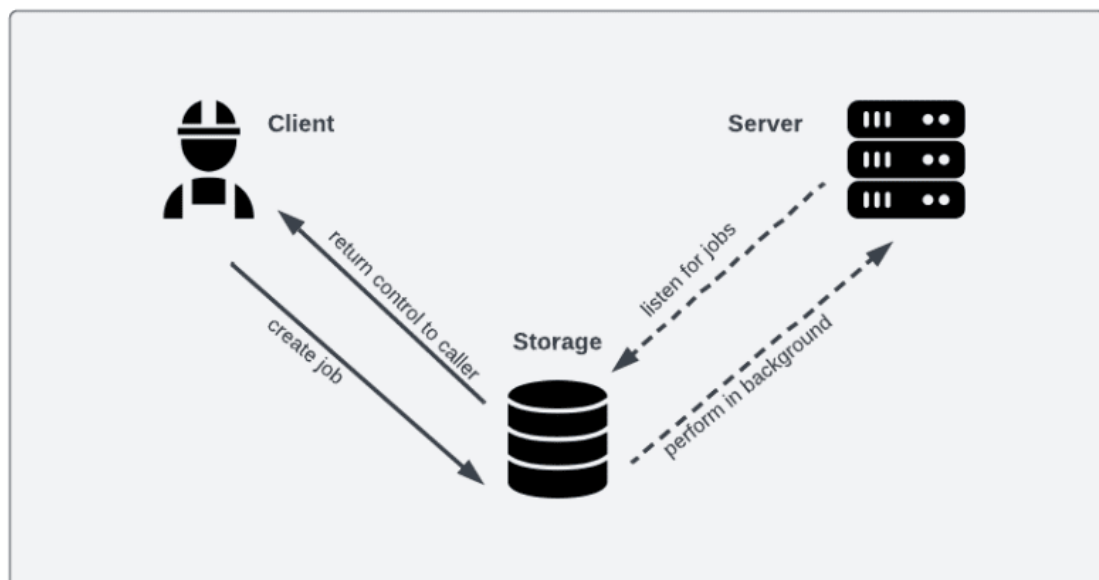
Add stock

Joonis 7. Kasutajaliidese näide "Home" vaade

Kasutajaliidese disain (Joonis 7) on minimalistlik ning praktiline. Kasutaja ei saa sisestada valesid andmeid tänu validaatoritele ning sisend väljade juures on tööriistavihjed, mis abistavad kasutajat.

4.3 Andmebaas ja taustatööd

Kuna rakendusel ei ole kasutajaid või sisselogimist, ei salvestata ka kasutaja andmeid ehk rakendusel pole vajadust suure või keerulise andmebaasi järele. Andmebaasi ennast on siiski mingil määral vaja, kuna aksia info päritakse läbi väliste API-de nimega *Twelve Data API* [23] ja *Yahoo Finance API*. Isegi kõige kallimat paketti kasutades ning selle kaudu 3300 päringu (umbes nii palju erinevaid aktsiaid on kirjutamise hetkel NASDAQ turul) pärimine võtab aega umbes 2 minutit, mis ei ole aktsepteeritav kiirus. Lisaks, kui peaks mingil põhjusel olema väliste API-de teenused maas, oleks ka selle projekti funktsionaalsus kadunud. Nende probleemide lahendamiseks kasutatakse Hangfire, mis võimaldab automaatselt rakenduse taustal hallata ja käivitada töid. Hangfire loob andmebaasi vajalikud tabelid, et seal tasuta tööde kohta infot hoida ning hiljem serveri abil neid ka käivitada (Joonis 8).



Joonis 8. Hangfire töövoog [24]

Hangfire kasutades saame iga 24 tunni tagant küsida vajaminevaid andmeid aktsiate kohta *Twelve data ja Yahoo finance API*'lt ning need omakorda salvestada rakenduse enda andmebaasi, kust saame hiljem murdosa sekunditega vajalikud andmed kätte.

Kasutame kahte erinevat API't kuna vajaliku info saamiseks on kõigepealt Twelve data'lt küsida kõiki NASDAQ turul olevaid aktsia sümboleid ning seejärel saame pärida Yahoo finance kaudu sümboli kaudu aktsia infot. Selline lahendus ei ole küll ideaalne, kuid selliseid probleeme tuleb lahendada kui kasutada teiste süsteemide API-sid. Üks variant oleks teha enda API mis pärib kõik vajalikud andmed otse, kuid see jääb selle projekti skoobist välja.

Välistest API-dest võetakse ainult vajaminevad andmed: aktsia sümbol, hind ja beeta ning salvestatakse need rakenduse enda andmebaasi. Kui peaks juhtuma, et API-de teenused on maas, siis jääb rakendus siiski töökorda ning kasutaja saab kasutada vananenud andmeid.

Stocks	
Symbol	Varchar(255) (unikaalne võti)
Price	Double
Beta	Double

Tabel 3. Aktsiate andmebaasi tabel

Kuigi andmebaasis on ainult üks tabel nimega Stocks, on selle populeerimine andmetega üsnagi tülikas ja keeruline.

4.4 Kontrolleri klassid

Kontroller klassid (Joonis 9) võtavad sisse andmebaasi konteksti ja objektide mudelite validaatorid ning sisaldavad endas ainult otspunktide meetodeid, mis pärivad andmebaasist infot või muudavad väljastavad andmeid vastavalt kasutaja sisestatud andmetele. Meetodid saavad vajalikud andmed kasutaja sisendist ning teenustest läbi Provide() meetodi tagastavad andme objekti nagu näiteks algoritmi poolt soovitatavad aktsiad. Nswag [25] abil saab automaatselt genereerida (Joonis 10) API kliendi kasutajaliidese kihis, et kontrolleri meetodeid otse Vue lehtedel välja kutsuda. Lisaks genereeritakse ka kõik vajaminevad andme mudelid ja luuakse tugevalt tüübitud objektid *TypeScript* abiga.

```

[ApiController]
[Route("stocks")]
public class StockController : ControllerBase
{
    [HttpPost]
    [ProducesResponseType(typeof(List<StockInfoViewModel>), 200)]
    [ProducesResponseType(typeof(int), 400)]
    public async Task<IActionResult> SuggestStocks([FromServices]
    StockListViewModel.IProvider provider, [FromBody] StockInfoInputModel
    input)
    {
        try
        {
            var validation = await
            _stockInfoInputValidator.ValidateAsync(input);
            if (!validation.IsValid)
            {
                validation.AddToModelState(ModelState);
                return BadRequest(ModelState);
            }
            var suggestedStock = await provider.Provide(input);
            if (suggestedStock.StockInfos != null &&
            !suggestedStock.StockInfos.Any())
            {
                return BadRequest("Something went wrong when
            suggesting a stock");
            }
            return Ok(suggestedStock.StockInfos);
        }
        catch (Exception e)
        {
            Console.WriteLine(e);
            return BadRequest("Error getting stock info, perhaps there
            is a problem with your input");
        }
    }
}

```

Joonis 9. Aktsia kontrolleri ja aktsia soovitamise meetod

```

suggestStocks(input: StockInfoInputModel):
Promise<StockInfoViewModel[]> {
    let url_ = this.baseUrl + "/stocks";
    url_ = url_.replace(/[?&]$/, "");

    const content_ = JSON.stringify(input);

    let options_: RequestInit = {
        body: content_,
        method: "POST",
        headers: {
            "Content-Type": "application/json",
            "Accept": "application/json"
        }
    };

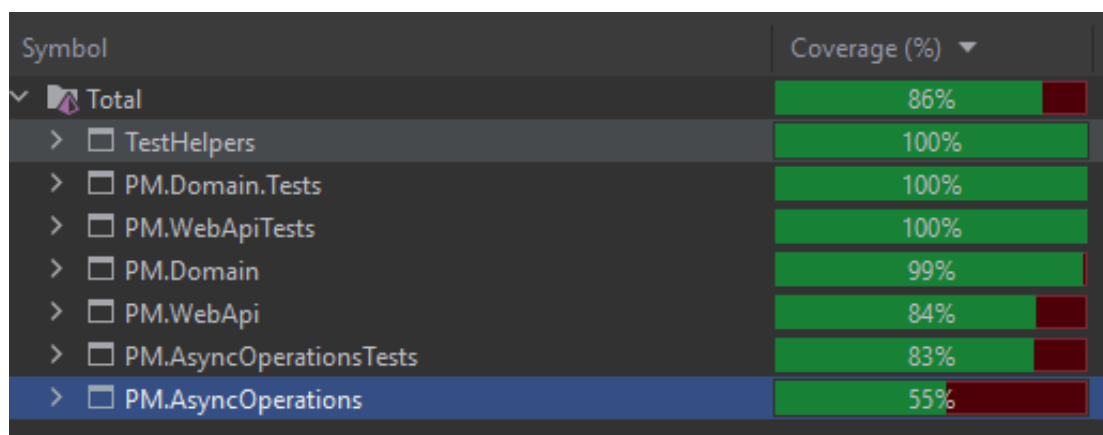
    return this.http.fetch(url_, options_).then((_response:
Response) => {
        return this.processSuggestStocks(_response);
    });
}

```

Joonis 10. Nswag abil genereeritud kood SuggestStocks meetodist

4.5 Testid

Rakendust testitakse kahte sorti testidega - ühiktestid, mis kontrollivad funktsionaalsuse või loogika väikest osa ning integratsiooni testid, mis kontrollivad, et funktsionaalsus tervikuna töötaks vastavalt ootustele. Kood peab olema kaetud võimalikult paljude testidega, et saaksime kindlad olla, et rakendus töötab ning, et saaksime tulevikus kergemini muudatusi sisse tuua ilma ettenägematute vigadeta. Terve rakenduse testide kaetus (Joonis 11) on 86%, mis vastab eelnevalt välja toodud nõuetele.



Joonis 11. Rakenduse testide kaetus

4.5.1 Ühiktestid

Ühiktestid kontrollivad koodi komponentide korrektsust isoleeritult, st ilma välisteenuste -või andmete kasutamiseta. Ühiktestidega katame ära funktsioonide või meetodite kõik võimalikud juhtumid, ehk kontrollitakse kõik valiidsed ja vigased võimalused üle. Mõlemal juhul kontrollitakse, et meetod tagastaks soovitud tulemuse, milleks on näiteks soovitud mudel, *HTTP* päringu vastuse koodid või veateated.

4.5.2 Integratsioonitestid

Integratsioonitestidega kontrollime, et rakenduse väikesed osad omavahel kõik koos töötaksid. Siin kohal olid kõige tähtsamad testid väliste API-de testimine ehk kontrollime, et saame kätte õiged andmed ja muudame need korrektseteks mudeliteks ning salvestame kohalikku andmebaasi.

4.6 Aktsia soovitamise algoritm

Algoritm (Joonis 12) soovib aktsia kasutajale tema enda sisendite kaudu ehk kasutaja peab sisestama enda portfoolios olevad aktsiad ning nende väärtus dollarites, lisaks sisestatakse ka raha mida ollakse nõus kulutama ja soovitud beta ehk riskitase. Algoritm võtab info sisse ning hakkab üks haaval kombineerima andmebaasis olevate aktsiatega, et saada portfelli riskitase võimalikult lähedale soovitud tulemusele. Hetkel on algoritmi keerukus O notatsioonis [26] $O(n)$, kus n on andmebaasis olevate aktsiate arv. See on üsna *kallis* tegevus, kuid kuna on teada, et aktsiad on selle töö kirjutamise ajal üle 3300 ning nende kogus ei suurene eksponentsiaalselt, siis ei võta see tegevus üle 5 sekundi aega, mis on eelnevalt mainitud nõue rakenduse jõudlusele.

```

public async Task<StockListViewModel> Provide(StockInfoInputModel input)
{
    StockListViewModel currentBestStock = null;
    var currentPortfolioTotalAdjustedBeta =
    ToViewModel(input).Result.TotalAdjustedBeta;

    if (currentPortfolioTotalAdjustedBeta > input.DesiredBeta)
    {
        double currentBestBeta = 1000;
        foreach (var stock in _context.Stocks)
        {
            var currentStock = new OwnedStockInfo {
AmountOwnedInDollars = input.MoneyToSpend, Symbol = stock.Symbol };
            input.StockInfos!.Add(currentStock);
            var stockInfoWithCurrentStock = await ToViewModel(input);
            if (stockInfoWithCurrentStock.TotalAdjustedBeta <
currentPortfolioTotalAdjustedBeta &&
                Math.Abs(stockInfoWithCurrentStock.TotalAdjustedBeta
- input.DesiredBeta) < Math.Abs(currentBestBeta - input.DesiredBeta))
            {
                currentBestBeta =
stockInfoWithCurrentStock.TotalAdjustedBeta;
                currentBestStock = stockInfoWithCurrentStock;
            }
            input.StockInfos.Remove(currentStock);
        }
    }
    else {
        double currentBestBeta = -1000;
        foreach (var stock in _context.Stocks)
        {
            var currentStock = new OwnedStockInfo {
AmountOwnedInDollars = input.MoneyToSpend, Symbol = stock.Symbol };
            input.StockInfos!.Add(currentStock);
            var stockInfoWithCurrentStock = await ToViewModel(input);
            if (stockInfoWithCurrentStock.TotalAdjustedBeta >
currentPortfolioTotalAdjustedBeta &&
                Math.Abs(stockInfoWithCurrentStock.TotalAdjustedBeta
- input.DesiredBeta) < Math.Abs(currentBestBeta - input.DesiredBeta))
            {
                currentBestBeta =
stockInfoWithCurrentStock.TotalAdjustedBeta;
                currentBestStock = stockInfoWithCurrentStock;
            }
            input.StockInfos.Remove(currentStock);
        }
    }
    return currentBestStock!;
}

```

Joonis 12. Aktsia soovitamise algoritm

Algoritmi saaks veel parandada tuues sisse lisa parameetreid peale beeta, mille abil saab olla kindlam, et soovitatud aktsia on parim. Lisaks saaks tuua sisse ka valikuid nagu dividendidega või dividendideta aktsiad või hoopis välistada halvasti toiminud aktsiad.

5 Tulemus ja järeldused

Selles peatükis käsitletakse valminud töö ärilisi kasumeid, tuleviku arendamisvõimalusi ning analüüsitakse töö käigus esinenud õnnestumisi ja ebaõnnestumisi.

5.1 Valminud funktsionaalsus

Lõpuks valminud rakenduses on peaaegu kõik funktsionaalsused mis eelnevalt välja toodud nõuetes kirjas ehk kasutaja saab luua enda portfoolio erinevatest NASDAQ aktsiatest ning seejärel soovitatakse vastavalt riskile kasutajale aktsia. Aktsiatega seostuv info uueneb iga 24h tagant ning algoritmi annab vastuse vähem kui sekundi jooksul. Ainukesed puudused nõuetest on, et kolme aktsia asemel soovitatakse vaid ühte.

5.2 Äriline kasum

5.2.1 Kulud

Rakenduse ülalhoidmise kulud jaotuvad järgnevalt:

- Domeen + hostimine on 21 eurot aastas namecheap.com vahendusel
- Andmebaasi ülalhoidmise kulu 21.12 eurot aastas hetzner.com vahendusel
- Yahoo Finance API 336 eurot aastas kui tahta iga päev värskaid aktsiate andmeid
- Twelve Data API 0 eurot aastas

Kokku kulub aastas ligikaudselt 378 eurot rakendus üleval hoidmisele.

5.2.2 Tulud

Kuna töö eesmärk on pakkuda võimalust tasuta inimestel investeerimisteadmisi ja -oskusi harida ei ole rakendusel ühtegi otsesest tuluallikat. Võimalus on kasutada otseseid annetusi kasutajatelt sarnaselt paljudele veebilehtedele nagu näiteks Wikipedia, LibreOffice jne. Kuna tegemist on haridusteamalise projektiga, on võimalik saada toetusi Haridus- ja Teadusministeeriumilt või teha koostööd koolidega. Lisatulu allikana on ka võimalik müüa lehekülge reklaami pinnana.

Praeguses seisus ei ole rakendus mõeldud ärikasumiks, vaid selle eesmärk on pigem panustada ühiskondliku teadlikkuse tõstmisesse ja harimisse.

5.3 Kordaminekud ja õppetunnid

Projekti käigus omandasin palju uusi teadmisi sarnaste tööde kirjutamise ja arendamise kohta ning kinnistasin varasemalt õpitut.

Suurimaks kordaminekuks oli erinevate API-dega töötamine ja nende koostöö rakendamine vajaliku info saamiseks. Sain teada miks üritatakse tavaliselt võimalikult palju väliseid API-sid vältida, kuna neist endale just sobiva info saamine ei ole alati lihtne, lisaks on need üsna kallid ja vahel aeglased.

Omandasin väärtuslikku kogemust luues iseseisvalt *full stack* rakenduse täiesti nullist. Sain rakendada kõiki varasemalt õpitut nii arenduse kui ka majanduse poolelt, ühes suures projektis.

Kõige suuremaks õppetunniks oli kasutajaliidese tööle saamine ja disainimine, kuna eelnevalt on sellega üsna vähe kogemusi. Veel ühe väärtusliku õppetunni sain, kui ei tegelenud projektiga mitu kuud. Tagasi tulles kogesin loetava koodi ja testide tähtsuse olulisust. Kokkuvõtvalt loen projekti suureks kordaminekuks.

5.4 Rakenduse edasiarendus

Kuna terve rakenduse põhimõte on soovitada kasutajale aktsiaid, siis üks põhilistest potentsiaalsetest edasiarendustest oleks kindlasti aktsia algoritmi täpsemaks tegemine. Kasutades rohkem sisendeid nagu näiteks P/E suhtearvu, võlgade taset, tulude kasvu jne. saame palju täpsemalt soovitada kasutajale aktsiaid.

Kulude kärpimiseks ja tuleviku tööde lihtsustamiseks oleks vaja asendada välised API-d enda tehtud API-ga, mis päriks Yahooolt ainult vajaliku info. See eemaldaks meie seose väliste ressurssidega ning kiirendaks andmete pärimist märkimisväärselt. Suureks kasuks tuleks funktsionaalsete kasutajaliidese testide kasutuselevõtt. Hetkel on kasutajaliidese poolne rakendus täiesti testimata, kuna seda ei ole lihtne teha ning ei ole tavapraktikas kasutusel. Kasutades raamistiku nagu Selenium saame kirjutada testid, mis käivad läbi kasutaja töövood vajutades läbi vajaminevad nupud ja sisestades korrektsed sisendid.

Lisaks nendele tehnilistele töödele tuleks läbi viia kasutajate tagasiside küsitlus, mille põhjal saaksime parandada kasutaja kogemust, kasutaja liidest ning tuua sisse uusi vajalikke funktsionaalsusi, mis teeks kõik rakenduse üldiselt paremini kasutatavaks.

Kõik need arendused aitaksid rakendusel olla paremini kasutatavam, silmajäävam ning ka äriiselt kasulikum.

6 Kokkuvõte

Tänapäeval räägitakse järjest rohkem kui suur roll on investeerimisel finantsvabaduse saavutamisel ning see on tehtud lihtsamaks kui kunagi varem. Selle tulemusel on üha rohkem inimesi, kellel tekib hirm, et nad jäävad ilma ning kes investeerivad oma raha lihtsalt selleks, et seda teha, ilma selget eesmärki või arusaama omamata.

Antud töö eesmärk on luua kõigile kättesaadav veebirakendus, mis suurendaks teadmisi ja huvi juba investeerinud inimeste seas. Töö arendati kasutades kaasaegseid arenduse tööriistu ja meetodikaid. Koodi kirjutamisel kasutati puhta koodi printsiipe ning arendusprotsessi aluseks võeti agiilne arendus meetodika, mis võimaldas pidevalt väikseid muudatusi teha, et lõpp produkt oleks võimalikult hea ja kasutajasõbralik. Enne töö tegemist analüüsiti juba olemasolevaid alternatiivseid lahendusi, mille põhjal pandi paika rakenduse nõuded ja kasutajalood.

Bakalaureuse töö tulemusena valmis töötav veebirakendus, mis suudab kasutajale vastavalt tema sisenditele ja portfelli soovitada sobiva aktsia, mis laiendaks kasutaja portfelli hoides soovitud riskitaset. Kuid, et töö oleks ka äriiselt kasumlik oleks vaja teha mõningad lisaarendused, et saavutada suurimat edu võimalust. Sellele vaatamata vastab töö ettenähtud nõuetele ning täidab oma eesmärki harida ühiskonda

Kasutatud kirjandus

- [1] C. Bond, „Yahoo finance,“ 23 January 2022. [Võrgumaterjal]. Available: https://finance.yahoo.com/news/investing-changing-experts-think-better-130014384.html?guccounter=1&guce_referrer=aHR0cHM6Ly93d3cuZ29vZ2xlLmNvbS8&guce_referrer_sig=AQAAABS w6qvViDo4fdAeBzS-FdkTn9x1wpXm1qYlFnllA5s_OiwfseuNKQSfd73P9rs6TqScKnIQIX3m_O53r5_rPlkb. [Kasutatud 11 November 2023].
- [2] [Võrgumaterjal]. Available: <https://tradingeconomics.com/euro-area/inflation-cpi>. [Kasutatud 16 April 2024].
- [3] J. Fernando, „Balanced Investment Strategy: Definition and Examples,“ 6 April 2022. [Võrgumaterjal]. Available: <https://www.investopedia.com/terms/b/balancedinvestmentstrategy.asp>. [Kasutatud 11 November 2023].
- [4] K. Green, „LinkedIn,“ 18 January 2021. [Võrgumaterjal]. Available: <https://www.linkedin.com/pulse/four-common-excuses-why-people-dont-invest-how-avoid-ken>. [Kasutatud 11 December 2023].
- [5] E. J. Elton ja M. J. Gruber, „Modern portfolio theory, 1950 to date,“ *Journal of Banking & Finance*, kd. Volume 21, nr Issues 11–12, pp. Pages 1743-1759, 1997.
- [6] F. Black, Beta and Return, Institutional Investor, Inc, 1993.
- [7] W. KENTON, „Investopedia,“ 30 June 2022. [Võrgumaterjal]. Available: <https://www.investopedia.com/terms/b/beta.asp>. [Kasutatud 13 12 2023].
- [8] B. BEERS, „Investopedia,“ 30 August 2023. [Võrgumaterjal]. Available: <https://www.investopedia.com/ask/answers/041315/what-are-pros-and-cons-using-sp-500-benchmark.asp>. [Kasutatud 13 12 2023].
- [9] „Wallstreetprep,“ 24 September 2023. [Võrgumaterjal]. Available: <https://www.wallstreetprep.com/knowledge/portfolio-beta/>. [Kasutatud 16 May 2024].
- [10] A. W. Services, „Amazon,“ [Võrgumaterjal]. Available: <https://aws.amazon.com/what-is/restful-api/>. [Kasutatud 11 12 2023].
- [11] A. W. Services, „Amazon,“ [Võrgumaterjal]. Available: <https://aws.amazon.com/what-is/net/>. [Kasutatud 11 12 2023].
- [12] „A Go-to Guide for Agile Workflow Process,“ cflow, 2024.
- [13] contact@assetrebalancer.com, „Portfolio Rebalancer,“ 2020. [Võrgumaterjal]. Available: <https://assetrebalancer.com/tool>. [Kasutatud 15 12 2023].
- [14] „WealthFront,“ [Võrgumaterjal]. Available: <https://www.wealthfront.com/>. [Kasutatud 15 12 2023].
- [15] „Betterment,“ [Võrgumaterjal]. Available: <https://www.betterment.com/investing>. [Kasutatud 1 April 2024].

- [16] a. trunk, v. dermol ja n. š. trunk, „Financial Literacy among the Young,“ ToKnowPress, 2018.
- [17] Ani Caroline Grigion Potrich, Kelmara Mendes Vieira ja Guilherme Kirch, „Determinants of Financial Literacy: Analysis of the Influence of Socioeconomic and Demographic Variables,“ 2015.
- [18] T. Taulli, „What Is the Nasdaq?,“ Kiplinger, 2023.
- [19] S. Pittet, „Atlassian,“ [Võrgumaterjal]. Available: <https://www.atlassian.com/continuous-delivery/software-testing/code-coverage>. [Kasutatud 19 April 2024].
- [20] M. Rehkopf, „Atlassian,“ [Võrgumaterjal]. Available: <https://www.atlassian.com/agile/project-management/user-stories>. [Kasutatud 1 April 2024].
- [21] J. Emmitt A. Scott, SPA Design and Architecture, Burke Holland, 2015.
- [22] N. Ferguson, „Medium,“ 1 January 2023. [Võrgumaterjal]. Available: <https://medium.com/@teamtechsis/single-page-applications-spa-48b1b845b446>. [Kasutatud 19 May 2024].
- [23] „Twelvedata,“ [Võrgumaterjal]. Available: <https://twelvedata.com/stocks>. [Kasutatud 3 April 2024].
- [24] C. Ikewelugo, „dev.to,“ 31 May 2023. [Võrgumaterjal]. Available: <https://dev.to/chinonsoike/background-job-scheduling-in-net-using-hangfire-3ehm>. [Kasutatud 17 May 2024].
- [25] R. Suter, „Nswag“.
- [26] S. Huang, „FreeCodeCamp,“ 16 January 2020. [Võrgumaterjal]. Available: <https://www.freecodecamp.org/news/big-o-notation-why-it-matters-and-why-it-doesnt-1674cfa8a23c/>. [Kasutatud 3 April 2024].

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks¹

Mina, Heiko Lige

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose “Portfelli põhise aktsia soovitaja rakenduse kavandamine ja arendamine“ mille juhendaja on Karl-Erik Karu
 - 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.

19.05.2024

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