

TALLINNA TEHNIKAÜLIKOOL  
Infotehnoloogia teaduskond

Artjom Sarapov 193650IADB

# **Automaatse andmepõhise toodete jääkväärtuse arvutamise prototüüp ettevõtte Fairown näitel**

Bakalaureusetöö

Juhendaja: Brita Moorus  
B. Sc.

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: Artjom Sarapov

24.04.2023

## **Annotatsioon**

Käesoleva lõputöö eesmärgiks on luua prototüüp, et automatiseerida ettevõtte Fairown toodete analüüsimisega seotud protsesse. Autor kavatseb luua prototüübi, mis võimaldab koondada toodetega seotud andmed ühte kohta, agregeerida neid ja töödelda, andes võimaluse andmeid analüüsida ja nende põhjal arvutada toodete võimalikku jääkväärtust mingi ajaperioodi jooksul.

Eesmärgi saavutamiseks analüüsib autor olemasolevaid Fairowni toodete jääkväärtuse arvutamise seotud protsesse ning arendab prototüüpi antud protsesside automatiseerimiseks.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 29 leheküljel, 4 peatükki, 11 joonist, 11 tabelit.

## **Abstract**

### **Design and Development of the Prototype for an Automatic Data-based Product Residual Value Calculation Platform on the Example of Fairown**

The Fairown company is leading circular economy provider in Baltics and Scandinavia. Due to popularity of their product, the company have expanded very quickly and started selling new products. With expansion of products variety, it became hard to analyze new items which have resulted in some problems in one of their core processes – product’s residual value analyzing and forecasting.

Current thesis work theoretical part analyses the Fairown’s company business model and provides an overview of problems it poses. Through this process the author tries to identify the company’s needs and determine the inevitable functionality of the solution.

In practical part of current paper, the author tries to create a solution prototype which will be able to solve the identified problems and create added value to the company. The solution would be created based previously made business analyses, using development best practices. The aim is to create the solution which will be easily maintainable and improvable in future. In practical part of thesis work the author will create dataset model based on business rules, as will develop sustainable front- and back-end parts of the solution.

In conclusion of thesis work the author will examine the theoretical and practical parts of current paper.

The thesis is in Estonian and contains 29 pages of text, 4 chapters, 11 figures, 11 tables.

## Lühendite ja mõistete sõnastik

API	<i>Application Programming Interface</i> , rakendusliides
AR	<i>Autoregression</i> , autoregressioon
ARIMA	<i>Autoregressive Integrated Moving Average</i> , autoregressiivne integreeritud liikuv keskmine
ARMA	<i>Autoregressive Moving Average</i> , autoregressiivne liikuv keskmine
<i>Controller</i>	Arhitektuuri kontekstis klass, mis kirjeldab võimalike toiminguid ressursiga
DDD	<i>Domain Driven Design</i> , domeenipõhine disain
DTO	<i>Data Transfer Object</i> , objekt, mis kannab üle andmeid
Kasjutajalugu	<i>User-story</i> , lugu, mis kirjeldab ärivajadust
LSTM	<i>Long Short-Term Memory networks</i> , pikk lühiajaline mälu
MA	<i>Moving Average</i> , liikuv keskmine
REST	<i>Representation State Transfer</i> , arhitektuurne stiil, mis kirjeldab veebirakenduse arendamist
SARIMA	<i>Seasonal Autoregressive Integrated Moving Average</i> , hooajaline autoregressiivne integreeritud liikuv keskmine
<i>Service</i>	Arhitektuuri kontekstis klass, mis hoiustab endas ärioloogikat
SPA	<i>Single Page Application</i> , veebirakenduse liik, mis laeb esmaselt ühe dokumendi ja muudatuste puhul värskendab ainult muutuva osa
Traatraam	<i>Wireframe</i> , kasutajaliidese üldistatud visuaalne skeem
URI	<i>Unified Resource Identifier</i> , interneti ressursside adresseerimise tehnoloogia

## Sisukord

1 Sissejuhatus .....	10
1.1 Probleemi püstitus .....	10
1.2 Lõputöö eesmärk .....	11
2 Analüüs ja kliendi vajaduste kaardistamine .....	12
2.1 Projekti taust .....	12
2.1.1 Olemasoleva äriprotsessi kirjeldus .....	13
2.1.2 Kliendi vajadused .....	13
2.1.3 Minimaalne vajalik toode .....	15
2.2 Teoreetiline taust .....	15
2.2.1 Andmete päritolu .....	16
2.2.2 Andmete analüüs ja käsitlemise strateegia .....	16
2.2.3 Võimalikud mudelid .....	19
2.2.4 Klassikalised aegridade mudelid .....	19
2.2.5 Juhendamisega õppemudelid .....	19
2.2.6 Sügavõppe mudelid .....	19
2.2.7 Mudeli valik .....	20
2.3 Tehnoloogia valik .....	20
2.3.1 Üldine arhitektuur .....	20
3 Praktiline osa .....	22
3.1 Domeenipõhine disain .....	22
3.2 Andmebaas .....	23
3.2.1 Tabel “category” .....	24
3.2.2 Tabel “manufacturer” .....	24
3.2.3 Tabel “product” .....	25
3.2.4 Tabel “data_point” .....	25
3.2.5 Tabel “country” .....	26
3.2.6 Tabel “auth_user” .....	26
3.2.7 Lokaalne arendus .....	27
3.3 Kasutajaliides .....	27

3.3.1 Prototüüp - traatraamide loomine .....	28
3.3.2 Projekti seadistamine .....	29
3.3.3 Arhitektuur.....	29
3.3.4 Peavaate mooduli struktuur .....	31
3.4 Rakendusliides.....	33
3.4.1 Projekti seadistamine ja tehnoloogiad .....	33
3.4.2 Arhitektuur.....	33
3.4.3 Kasutusjuhtumid .....	36
4 Tulemused ja analüüs .....	37
Kokkuvõte .....	39
Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks .....	42
Lisa 2 – Jääkväärtuse analüüsimise ja prognoosimise olemi-suhte diagramm .....	43
Lisa 3 – Kliendi rakenduse tee ja komponendi struktuuri seose näide.....	44

## Jooniste loetelu

Joonis 1. iPhone Pro 256 GB mudelite hinnakäitumine ajas [5]. .....	17
Joonis 2. Apple iPhone 12 128 GB Black hinnamuutused aja vältel [6].....	18
Joonis 3. Apple iPhone 13 128 GB Midnight hinnamuutused aja vältel [7]. .....	18
Joonis 4. Rakenduse arhitektuur API tasandil. ....	21
Joonis 5. Docker Compose konfiguratsioon andmebaasi käivitamiseks. ....	27
Joonis 6. Kasutajaliidese traatraam prototüübi ülevaatliku lehekülje kohta. ....	28
Joonis 7. Kasutajaliidese projektikeskne arhitektuur. ....	31
Joonis 8. Peavaates asuv ülevaatlik statistika vaade.....	33
Joonis 9. Rakendusliidese entiteedid. ....	34
Joonis 10. Isoleeritud rakendusliidese projekti arhitektuur. ....	35
Joonis 11. Arhitektuuri osade jaotus kihtidena.....	35



## Tabelite loetelu

Tabel 1. Esimese kasutajaloo kriteeriumid.....	13
Tabel 2. Teise kasutajaloo kriteeriumid .....	14
Tabel 3. Kolmanda kasutajaloo kriteeriumid .....	14
Tabel 4. Andmebaasis olevate tabelite semantika .....	23
Tabel 5. Andmebaasi tabeli “category” struktuur. ....	24
Tabel 6. Andmebaasi tabeli “manufacturer” struktuur.....	24
Tabel 7. Andmebaasi tabeli “product” struktuur. ....	25
Tabel 8. Andmebaasi tabeli “data_point” struktuur. ....	25
Tabel 9. Andmebaasi tabeli “country” struktuur. ....	26
Tabel 10. Andmebaasi tabeli “auth_user” struktuur.....	26
Tabel 11. Andmekirje domeeni URI koos meetodite ja nende kirjeldustega. ....	36

# 1 Sissejuhatus

Suurandmete tulekuga ja tänapäeval kiirelt muutuvate turgude tõttu on andmete analüüs hakanud mängima üha olulisemat rolli keskkonnaalastes sektorites, mõjutades vastuvõetavaid otsuseid ja luues uusi võimalusi. Andmete uurimine on andnud erasektorile innovatiivseid tööriistu ja meetmeid äriprotsesside täiendamiseks ning konkurentsieelise saavutamiseks. Ringmajanduse põhimõttel seisneva ärimudeli olulisemad protsessid on toodete hinna tendentside analüüsimine ja jääkväärtuse ennustamine. Antud protsessid on otseselt seotud ettevõtte toodete hinnakujundamisega, mõjutades tekitatavat kasumit ning määrates ettevõtte positsioneerimist turu peal. Fairown on juhtiv ringmajanduse pakkuja Baltimaades ja Skandinaavias, kes on võimeline saama kasumit automatiseeritud toodete hinnastatistika kujundamisega ning täpsema toodete jääkväärtuse arvutamisega.

Käesoleva töö analüüsi raames uurib autor ettevõtte Fairown ärimudelit ja püstitatud probleemi. Selle käigus proovib autor teha kindlaks ettevõtte vajadused ja määrata lahenduses vajalikku funktsionaalsust.

Töö praktilises osas kavatses autor põhineda eelnevalt tehtud analüüsile ning luua prototüüp, mis lahendab eelnevalt püstitatud probleemi ning loob ettevõttele lisaväärtust. Arendatav prototüüp peab looma tugeva põhja selle edasi arendamiseks. Selle jaoks kavatses autor luua andmebaasi mudeli ning arendada nii kliendipoolse liidese, kui ka rakendusliidese, kasutades selleks parimaid arenduse meetodeid.

## 1.1 Probleemi püstitus

Ettevõtte Fairown on Eesti start-up, mis võimaldab partneritele pakkuda ringmajanduse põhimõttel e-kommerts toodete kasutamist. Nende üheks oluliseimaks äriprotsessiks on toodete hinna analüüsimine ja nende potentsiaalse jääkväärtuse arvutamine. Toodete ja nende hindade analüüs mängib olulist rolli tarbija laenu kuumakse määramises ja mõjutab ettevõtte kasumit. Olemasolevad analüüsi- ja hindade arvutamise protsessid on

manuaalsed ja ajamahukad. Kuna ettevõtte on jõudsalt kasvanud ning spetsiifilisi tooteid on turule järjest lisandumas, on oluline välja töötada automatiseeritud ning jätkusuutlik lahendus, mis võimaldab automatiseerida analüüsi protsessi ja anda hea aluse selle lahenduse edasi arendamiseks.

## **1.2 Lõputöö eesmärk**

Töö käigus hindab autor erinevaid andmete töötlemise ja prognoosimise protsesse, leiab sobivama ettevõtte Fairown jaoks ning põhjendab enda valikut sõltuvalt äri spetsiifikast ja püstitatud erivajadustest. Tuginedes analüüsile koostab autor standardiseeritud ja jätkusuutliku toodete jääkväärtuste hindamise protsessi ning loob kasutajaliidese prototüübi. Arendatud liides peab saama agregeerida toodete andmeid ning võimaldama lõppkasutajal analüüsida toodetega seotud väärtusi ja prognoosida hindade võimalikku kujunemist teatud ajaperioodi möödudes. Antud prototüüp peab looma tugeva aluse selle edasiarenduseks.

## 2 Analüüs ja kliendi vajaduste kaardistamine

Käesolevas peatükis on esitatud töö analüütiline ja teoreetiline osa. Selles peatükis analüüsitakse ettevõtte tausta ja tegevusala ning olemasolevat lahendust. Antud peatükis kirjeldatakse tehtud analüüsi, mille käigus vaadeldi erinevad matemaatilisi prognoosimise meetodeid. Uuringute käigus oli leitud sobivaim lahendus püstitatud eesmärkide täitmiseks.

### 2.1 Projekti taust

Fairown on Eesti start-up, mille fookus on arendada innovatiivset lahendust, et suurendada toodete olelusringi ja muuta toodete kasutamist jätkusuutlikumaks. Ettevõtte on asutatud aastal 2018 kahe eksperdi poolt, kes nägid suurt potentsiaali muuta paremaks e-kaubanduse toodete elutsüklit. Lahenduseks sai toode, mis võimaldab implementeerida e-kaubanduse partneritele ringmajanduse põhimõttel toodete kasutamist. Fairowni ärivoog pakub võimalust lõppkasutajal kasutada tooteid ning maksta nende eest igakuiselt nende kasutamise eest. Perioodi lõpus on kasutajal võimalus osta toode jääkväärtuse eest välja või uuendada seda jätkusuutlikult, jätkates igakuiste maksetega. Toote uuendamise aeg on hoolikalt välja arvatud, arvestades kindla toote eripärasid nii, et vahetuse hetkel omaks toode piisavalt turuväärtust, aktuaalsust ning ilmemas on esimesed amortisatsioonimärgid. Arvestades kõiki tegureid, on toote järgmiseks võimalikuks eluetapiks üks järgnevast:

- toode saab teise elu järelturul ning vajadusel tehakse sellele pisiparandusi või täiendusi;
- tootest saab varuosade doonor teiste seadmete remondiks või uute toodete valmistamiseks;
- toode töödeldakse ümber ja materjale taaskasutatakse.

Ringmajanduse põhimõttel lähenemine on võimaldanud tugevasti pikendada toodete elutsüklit. Toodete taasringlus ning elutsükli pikendamine aitab vähendada jäätmeid ning kahju keskkonnale, sest tootmine on toote elutsükli kõige energiakulukam faas. Antud lähenemine on näidanud ka häid majanduslikke tulemusi, võimaldades Fairownil väga kiiresti kasvada ja laieneda ka välisriikidele. Tänapäevaks tegutseb Fairown nii Baltikumis,

Poolas, Saksamaal ning ka Skandinaavia riikides, pakkudes enda lahendust suurematele ettevõtetele ja partneritele, kes soovivad vähendada enda ökoloogilist jalajälge ja mõju keskkonnale [1, 2].

### 2.1.1 Olemasoleva äriprotsessi kirjeldus

Olemasolevad toodetega seotud analüütilised protsessid on manuaalsed, ajamahukad ning ei oma head struktuuri. Analüüsiks tarvilikud andmed on otsitud erinevatest ressursidest ja on valideeritud Fairowni ekspertide poolt, tuginedes nende aastatega omandatud teadmistele ja kogemustele. Andmete korjamiseks, töötlemiseks ja käsitlemiseks on seni kasutatud erinevaid manuaalseid praktikaid, mis on paraku ajamahukad ning veaohklikud.

### 2.1.2 Kliendi vajadused

Analüüsi- ning arendusprotsessi teostamiseks on valitud agiilne lähenemine. Antud lähenemise peamiseks eesmärgiks on tihe koostöö tellijaga, et pidevalt valideerida ja määrata uusi vajadusi ning vaadelda arendustiimi poolt tehtud tööd. Antud lähenemine on valitud Fairowni keerulise äriprotsessi tõttu, et luua täpsem prototüüp ja juba algfaasis tuua ettevõttele võimalikult suurt lisaväärtust [3].

Vajaduste määramine on samuti tehtud agiilses vormis – pidevate koosolekute ja kohtumiste põhjal analüüsiti äriprotsesse, vaadeldi kasutajalugusid, määrati prototüübi kriteeriumid, koostati kasutajaliidese ja –teekonna disainid ning valideeriti saavutatud tulemusi tellijaga [3, 4].

Analüüsi käigus kirjeldati kolm peamist kasutajalugu, millele järgnevalt määrati esimesed vajadused.

Esimene kasutajalugu – kasutaja soovib vaadata konkreetse toote hinnaseisu turul, et analüüsida toote hinna käitumist. Tabelis 1 on toodud välja esimese kasutajaloo kriteeriumid ning tulemused.

Tabel 1. Esimese kasutajaloo kriteeriumid

Olemas	Tegevus	Tulemus
Olemas on kasutatud/järelturu toodete hinna andmed.	Kasutaja valib analüüsitava toote kategoorias „kasutatud“.	Kasutaja näeb toodete hinnastatistikat graafikuna.
Olemas on uute toodete hinna andmed.	Kasutaja valib analüüsitava toote kategoorias „uued“.	Kasutaja näeb uute toodete hinnastatistikat graafikuna.

Olemas on nii järelturu kui ka uute toodete hinna andmed.	Kasutaja valib analüüsitava toote.	Kasutaja näeb valitud toodete olemasolevate andmete numbrilist statistikat ja graafikut.
---	------------------------------------	--

Teine kasutajalugu – kasutaja soovib näha toodete hinnaprognosi, et kontrollida või määrata laenusummad, toodete jääkväärtuse ning kuumakse. Tabelis 2 on toodud välja teise kasutajaloo kriteeriumid ning tulemused.

Tabel 2. Teise kasutajaloo kriteeriumid

Olemas	Tegevus	Tulemus
Olemas on nii järelturu kui ka uute toodete hinna andmed.	Kasutaja valib analüüsitava toote.	Kasutaja näeb valitud toote graafikul hinnastatistika prognoosi.
Olemas on nii järelturu kui ka uute toodete hinna andmed.	Kasutaja valib analüüsitava toote ja sisestab vajalikud andmed.	Kasutaja näeb matemaatilist prognoosi arvulises väärtuses.
Olemas on nii järelturu kui ka uute toodete hinna andmed.	Kasutaja valib analüüsitava toote.	Kasutaja näeb valitud toodete olemasolevate andmete numbrilist statistikat.

Kolmas kasutajalugu – kasutaja soovib hallata olemasolevaid andmeid nii uute kui ka järelturu toodete kohta, et neid muuta ja lisada uusi andmeid. Tabelis 3 on toodud välja kolmanda kasutajaloo kriteeriumid.

Tabel 3. Kolmanda kasutajaloo kriteeriumid

Olemas	Tegevus	Tulemus
Olemas on toodete andmed.	Kasutaja soovib andmeid lisada.	Kasutaja lisab uued andmed andmebaasi kindlaksmääratud kujul.
Olemas on toodete andmed andmebaasis.	Kasutaja valib andmete muutmise.	Kasutaja muudab olemasolevaid andmeid andmebaasis.
Olemas on toodete andmed andmebaasis.	Kasutaja valib andmete kustutamise.	Kasutaja kustutab olemasolevad andmed andmebaasis.

### **2.1.3 Minimaalne vajalik toode**

Analüüsi käigus oli määratud rakenduse algne funktsionaalsus läbi kasutajalugude. Järgnevalt toodud ülevaade skoobist on järjestatud tähtsuse järgi, et algses faasis oleks võimalik valideerida püstitatud eesmärgid ja luua minimaalne vajalik toode (ingl. k *Minimum Viable Product*).

Esimene skoop: selle ja varasemate hooegade toodete hinnaajalugu on rakenduses nähtav visuaalsel graafikul. Graafik peab andma hea ülevaate toote hinna kujunemisest ning muutustest mingi kindla ajaperioodi möödumisel. Samuti peab olema ülevaade toodete hinna prognoosist 1–3 aasta peale vähemalt 6–kuuliste etappidena.

Teine skoop: rakendus suudab kuvada toote hinnaprognoozi, kasutades selleks erinevaid matemaatilisi mudeleid ja kasutaja poolt sisestatud andmeid. Hinnaprognoozi arvutus peab arvestama kasutaja poolt antud sisendiga ning väljundina saavutama arvulisi väärtusi. Saavutatud väärtused on kasutatud jääkväärtuste ja kuumaksete arvutamiseks või nende kontrollimiseks.

Kolmas skoop: rakendus peab võimaldama hallata andmeid, neid lisada, muuta ja kustutada.

Käesoleva töö raames otsustati alustada esimese skoobi realiseerimisega, kuna selle skoobi kirjeldatud funktsionaalsus annab võimaluse valideerida püstitatud eesmärgid ning ühtlasi luua minimaalne vajalik toode.

## **2.2 Teoreetiline taust**

Tuginedes eelnevale analüüsile oli kindlaks tehtud, et tellijal on vajadus ennustada erinevate toodete hinna kujunemist mingi kindla ajaperioodi möödudes. Fairown pakub mitmetasandilist ringmajanduse põhimõttega äriteenust, mille raames toode läbib mitmeid etappe ja see võimaldab ettevõttel korjata ja analüüsida erinevaid andmeid hinna kujunemise kohta. Kõige suuremat mõju toodete jääkväärtuse kujundamisel omavad etapid, mille andmed illustreerivad toote algset hinnaseisu, näitavad selle kasutamise perioodi ja lõplikku väärtust, millega toode ostetakse tagasi. Andmete valiidsuse tagamiseks pikemas perspektiivis otsustati koostada jääkväärtuste arvutamise mudel, mis kasutaks olemasolevaid andmeid põhilise infoallikana. Antud lähenemine illustreerib õiget hinna kujunemist ja andmete käsitlemist, arvestades ettevõtte spetsiifikat.

### **2.2.1 Andmete päritolu**

Ärilise analüüsi tulemusena saab eristada toodetega seotud etappe, mis on vajalikud andmete käsitlemiseks:

1. toote hind, millega tarbija soovib osta valitud toodet;
2. kasutamise periood;
3. toode lõplik jääkväärtus.

Antud etapid võimaldavad fikseerida tootega seotud andmeid:

1. uue toote hind;
2. uue toote ostukuupäev ja –aeg;
3. laenuperiood;
4. kasutatud toote hind;
5. kasutatud toote edasimüügi kuupäev ja aeg;
6. riik;
7. valuuta.

Riigiga arvestamine mängib olulist rolli jääkväärtuse prognoosimise protsessis, kuna turu käitumine on suuresti riigi- ja regioonipõhine. Ühe riigi turuhinnad ja toodete hinnadünaamika võib tugevalt erineda teistest turgudest.

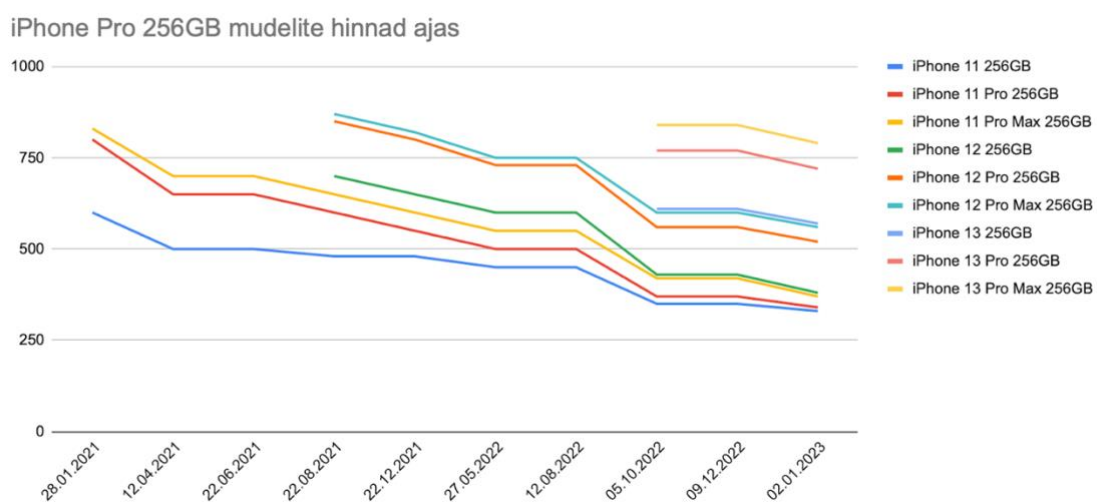
Andmete käsitlemises olev andmemudel peab võimaldama käsitleda andmeid ka kolmandatest allikatest. Nimetatud andmeid saab kasutada mudeli täiendamiseks ja valideerimiseks. Kuna Fairowni platvormi poolt väljastatud tooted ei ole veel jõudnud mõnes riigis uuendamise faasi, siis esmast jääkväärtuse arvutamist peab tegema kasutades selleks kolmandate allikate andmeid. Hilisemas faasis on võimalik mudelit täiendada reaalse ja valideeritud andmetega.

### **2.2.2 Andmete analüüs ja käsitlemise strateegia**

Äriprotsessi analüüsi tagajärjel oli välja töötatud potentsiaalselt sobivaim andmete käsitlemise voog. Toodete andmeid on võimalik agregeerida toote kategooria, tootja ja



konkreetsel mudelil järgi. Iga toote juurde kogutakse ajahetke andmed, mis on välja toodud peatükis 2.2.1. Antud andmekogum illustreerib kindla mudeli hinnakäitumist aja vältel. Prognosi strateegia määramiseks on vajalik uurida erinevate toodete vahelisi seoseid. Joonisel 1 on näidatud järelturul leiduvate toodete hinnakäitumine pikema perioodi vältel. Näidisandmetena on kasutatud Apple iPhone Pro mudelid. Graafik illustreerib järelturul leiduvate sarnaste seadmete hinna käitumist ajas. Selle põhjal on võimalik järeldada, et konkreetsel mudelil esineb hinna languse trend ning toodete hinnalangus toimub ühtlaselt.

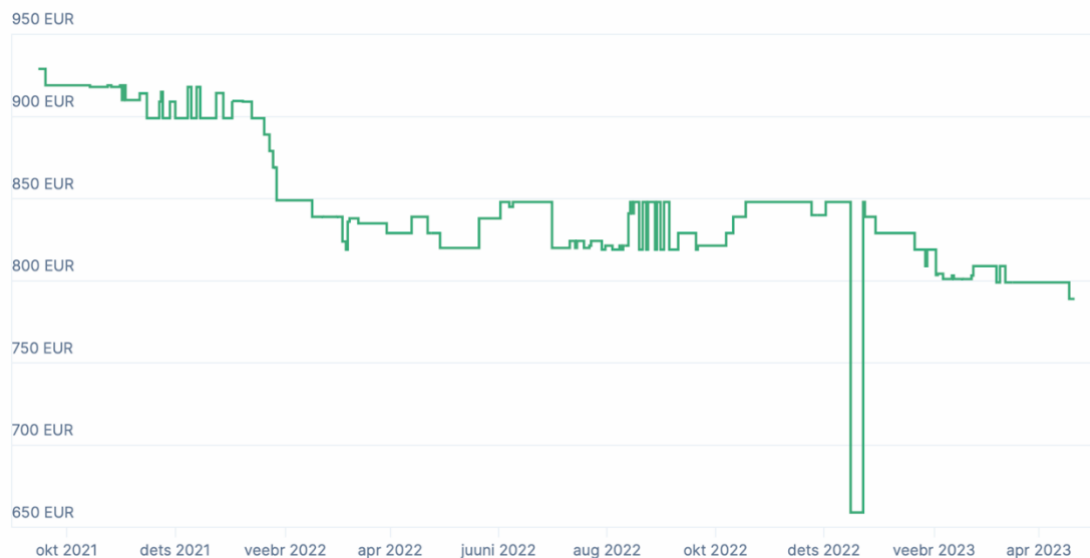


Joonis 1. iPhone Pro 256 GB mudelite hinnakäitumine ajas [5].

Joonisel 2 ja Joonisel 3 on toodud uute iPhone 12 ja iPhone 13 seadmete hinna kujunemine peale nende turule sisenemist. Analüüsidest toodete esimest aastat turul on võimalik järeldada, et toodetel esineb sarnane hinnakäitumine.



Joonis 2. Apple iPhone 12 128 GB Black hinnamuutused aja vältel [6].



Joonis 3. Apple iPhone 13 128 GB Midnight hinnamuutused aja vältel [7].

Analüüsi põhjal on võimalik püstitada hüpotees, et uue toote hinna muutmine toimub sarnaselt sellele eelneva mudeliga. Antud hüpotees võimaldab lähtuda jääkväärtuse prognoosimisele, arvestades eelnevalt sarnaste toodete hinna käitumist. Andmete üldistamiseks saab vaadata toodete protsentuaalse väärtuse muutumist selle müügipäevast, mis võimaldab joondada sarnaseid tooteid ühele teljele. Mudeli täiendamiseks saab selles määrata ka tähtsuse koefitsient, mis tuleneb andmete vanusest, selleks et uued andmed oleksid suurema mõjuga kui seda on vanemad andmed.

### 2.2.3 Võimalikud mudelid

Toote jääkväärtuse prognoosimine on võimalik kasutades selleks erinevaid matemaatilisi mudeleid. Mudeli valikuks on oluline klassifitseerida sellega analüüsitavaid andmed. Lähtuvalt peatükis 2.2.1 kirjeldatud struktuurile, on tegemist aeGRIDade andmetega, kuna üheks muutujaks on aeg ning andmed on järjestatud kronoloogilises järjekorras [8].

AeGRIDade prognoosimise mudeli valik sõltub andmete spetsiifikast, kvaliteedist ja mahust. Mudeleid saab jaotada kolmeks peamiseks kategooriaks:

- klassikalised aeGRIDade mudelid (ingl. k. *classical time series models*);
- juhendamiseiga õpe (ingl. k. *supervised models*);
- süvaõppe närvivõrgud (ingl. k. *deep learning-based models*) [8].

### 2.2.4 Klassikalised aeGRIDade mudelid

Klassikaliste aeGRIDade mudelite grupp on kasutatud paljudes ennustamise ülesannetes. Antud grupp põhineb tugevalt ajutistes varieeruvustes aeGRIDade andmete sees ja on võimeline jälgima suuresti ainult üht muutujat. Näited klassikalistest aeGRIDadega mudelistest: AR (ingl. k. *autoregression*), MA (ingl. k. *moving average*), ARMA (ingl. k. *autoregressive moving average*), ARIMA (ingl. k. *autoregressive integrated moving average*), SARIMA (ingl. k. *seasonal autoregressive integrated moving average*) [8].

### 2.2.5 Juhendamiseiga õppemudelid

Juhendamiseiga õppemudelid on sobilikud masinõppe ülesannete lahendamiseks. Mudelid on võimaldavad aeGRIDade prognoosimist, kuid vajavad eelnevat andmete valideerimist ja seadistamist. Selliste mudelite puhul on oluline andmete dekompositsioon hooegade muutujasse eraldamiseks. Antud muutuja on järgnevalt kasutatud  $x$ -telje väärtusena ja  $y$ -telje muutujaks on võetud teine, aeGRIDadest jälgitav muutuja [8]. Näited juhendamiseiga õppemudelistest: Lineaarne regressioon, juhuslik mets, XGBoost [8].

### 2.2.6 Sügavõppe mudelid

Viimaste aastate sügavõppe mudelite areng on võimaldanud kasutada antud tehnoloogiat aeGRIDade prognoosimiseks. Antud mudelid on võimelised töötama keeruliste andmestruktuuridega ja tuvastama mitmetes eraldiseisvates aeGRIDades ühtivaid ja

sarnaseid osi. Näited süvaõppe mudelitest: Facebook Prophet, LSTM (ingl. k. *long short-term memory networks*), DeepAR [8].

### **2.2.7 Mudeli valik**

Autor seab hüpoteesi, et uute toodete hindade käitumine saab olla sarnane eelnevalt turul esinenud vanade toodetega. Peatükkides 2.2.2 ja 2.2.3 tehtud järelduste alusel soovib autor kasutada toodete hindade ja jääkväärtuste prognoosimiseks mudeleid, mis on võimelised korraga töötleva mitmeid aegridu. Näiteks Apple iPhone 14 hinna käitumise prognoosi tegemiseks kasutatakse eelnevate mudelite Apple iPhone 12 ja Apple iPhone 13 aegridade hinnaandmed.

Eelneva analüüsi alusel on võimalik lähtuda prognoosimise probleemist, kasutades selleks sügavõppe mudelit – LSTM (ingl. k. *long short-term memory networks*).

## **2.3 Tehnoloogia valik**

Autori eesmärgiks on töö raames luua lahendus või prototüüp, mis annab piisava põhja tulevikus selle täiendamiseks, arendamiseks ja mugavaks haldamiseks. Seetõttu sai tehnoloogia valik tehtud põhinedes spetsiifilisele probleemile kui ka Fairownis kasutusel olevatele tehnoloogiatele ja tehnilisele arhitektuurile. Lisaks peab tehnoloogiate valikul võtma arvesse nende tänapäevast aktuaalsust ning turul kasutatavust.

### **2.3.1 Üldine arhitektuur**

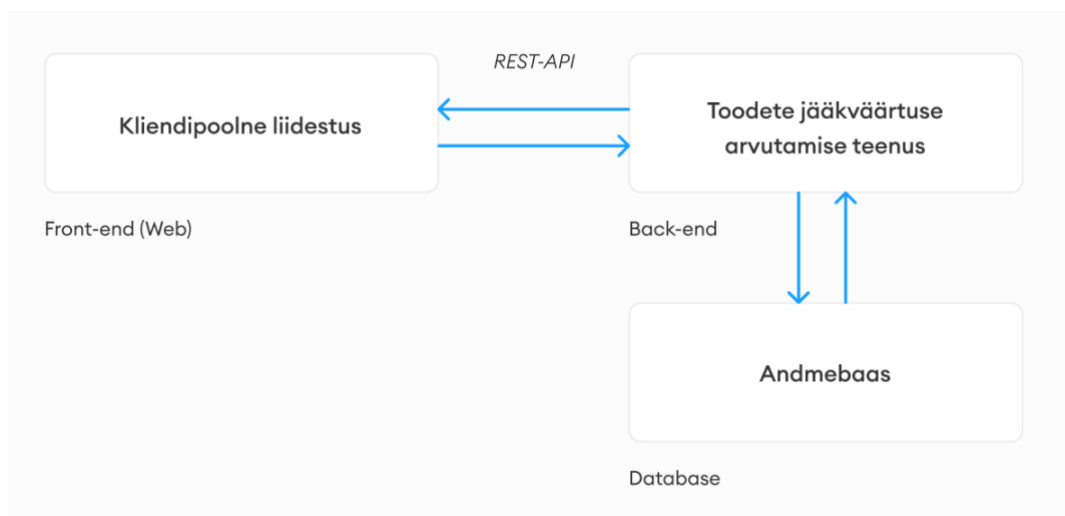
Fairowni üldine arhitektuurne lähenemine on tugevalt seotud mikroteenuse arhitektuuri põhimõttega. Mikroteenuse arhitektuur on arhitektuurne stiil või lähenemine, mille raames kõik rakendused või teenused moodustavad grupi, mille:

- eraldiseisvad lahendused on juurutatud isoleeritud keskkondades;
- iga lahendus täidab enda vastutusalas olevat äriprotsessi;
- lahendused on hallatud väiksema meeskonnaga;
- lahendused on väheses sõltuvuses teineteisega [9].

Võrreldes mikroteenustel põhinevat arhitektuuri monoliitse arhitektuuriga, mille peamine mõte on kokku koondada paljud teenused ühes keskkonnas ja hallata need ühisena, siis mikroteenuse lähenemine võimaldab lahti lüüa koodibaasi ning hallata koodibaasi ning ressursi jätkusuutlikumalt, suurendades arvutusvõimet ainult seal, kus see on vajalik [9]. Käesoleva töö probleemi lahendus peab olema arendatud eraldiseisva teenusena, et see saaks toetada mitut erinevat olemasolevat süsteemi. Analüüsi käigus oli kindlaks tehtud, et algses lahenduses ei ole vajadust suhtlemiseks teiste Fairowni teenustega.

Arvestades Fairowni arhitektuurilist eelistust ja mikroteenuste arhitektuurilisi põhimõtteid, jagatakse tehniline lahendus kolmeks eraldiseisvaks teenuseks: kliendipoolne vaade (ingl. k. *front-end*), äriloogika teenus (ingl. k. *back-end*) ja andmebaas. Kliendipoolne vaade suhtlemine äriloogika teenusega toimub REST API põhimõttel.

REST API on protokollide ja kokkulepete kogum, mis võimaldab mitmel programmiliidesel suhelda ja teineteisega vahetada andmeid [10]. Joonisel 4 on toodud välja rakenduse arhitektuur API tasandil ning erinevate osade omavaheline suhtlus.



Joonis 4. Rakenduse arhitektuur API tasandil.

## 3 Praktiline osa

Käesolevas peatükis on esitatud töö praktiline osa, mis põhineb eelnevalt kirjeldatud analüüsil. Praktilise osa raames tehti andmebaasi modelleerimine, arendati serveri loogika ning koostati kliendipoolne visuaalne vaade.

### 3.1 Domeenipõhine disain

Arendustöö on tehtud järgides tihedalt domeenipõhist disaini (ingl. k. *domain driven design*, DDD) lähenemist. Domeenipõhine disain on lähenemisviis tarkvaraarendusele, mille praktikad aitavad lihtsustada keeruliste süsteemide loomist. Põhiliseks fookuseks arenduses on äridomeeni areng ja tihe koodibaasi sõltuvus koos äriterminitega, kaasates domeeniteadust tarkvara arendusprotsessi [11].

DDD lähenemise keskpunktiks on äriiline domeen, mis täidab tarkvarasüsteemi tuuma rolli. Kõik äriteadus on illustreeritud tarkvara mudelis, mis kajastab ühist äridomeeni keelt ja ärikontseptsioone. Sellisel juhul muutub äriprotsesside ja nõuete arusaamine lihtsamaks, tehes nõuete interpreteerimist tarkvara protsessidesse sujuvamaks [11]. Eristatud on ka tähtsus projekti sidusrühmade vahelises suhtluses. On oluline, et kõik arendustöö osalised – domeenieksperdid ja arendajad, oleksid pidevas koostöös ja suhtluses teineteisega, jagades domeeniteadust ja projekti ülevaadet. Tihe koostöö peab säilitama projekti nõuete jälgimist ja kooskõlastada arendatavat tarkvarasüsteemi koos äri vajadustega, täites kõikide osapoolte eesmärgid [11].

On võimalik eristada olulisemaid DDD lähenemisi ja kontseptsioone:

- Ühine domeenipõhine ärikeel – kõik arendustiimi osapooled kasutavad ühist ärispetsiifilist keelt ja terminoloogiat. See aitab tuua projekti ühist sõnavara protsesside kirjeldamiseks ja tagab parema protsesside mõistmise kõikidele osapooltele.
- Domeeni piiritletud kontekstid – domeeni sees esinevad kontekstid on isoleeritud ja selgelt piiritletud. Antud kontekstid illustreerivad erinevaid osi nii äridomeenist kui ka arendatavast tarkvarast. Iga kontekst omab enda ärikeelt, mudeleid, reegleid ja suhtleb teiste kontekstidega, jälgides täpselt määratletud reegleid.

- Olemid ja väärtusobjektid – vaadeldava domeeni objektid, mille eesmärk on peegeldada äridomeeni atribuute ja kontsepte. Olemid omavad unikaalset identiteeti ja on võimelised aja vältel muutuma. Väärtusobjektid teiselt poolt on muutumatud ja illustreerivad kontseptsioone, mis ei oma unikaalset identiteeti [11].

DDD implementeerimine antud lahenduses omab olulist rolli, kuna Fairowni ärispetsiifika ja olemasolev domeen on unikaalne ja omab keerukust. DDD toetab tugevalt püstitatud eesmärke – arendada süsteem, mis on lihtsasti mõistetav, kooskõlastatud ärispetsiifikaga ja on seda on võimeline hõlpsasti edasi arendada [11].

### 3.2 Andmebaas

Andmebaasiks sai valitud relatsiooniline andmebaas PostgreSQL, mis koosneb tabelite seostest. Nimetatud andmebaas on kasutatud kõikides olemasolevates Fairowni projektides ja on sobilik käesoleva projekti raames, kuna võimaldab mugavalt rakendada ärispetsiifilist domeeni ja luua skaleeritav süsteem.

Andmebaasi modelleerimine on tehtud järgides domeenipõhist lähenemist. Oluline on hoida andmebaasi mudeleid seotud üldise ärikeelega. Fairown omab siseselt enda domeeni äriteadust, mis peab olema peegeldatud projekti andmebaasis.

Andmebaasi mudelid ja domeen illustreerivad ning toetavad äriprotsesse ja püstitatud eesmärke. Mudelid on koostatud peatüki 2.2.1 analüüsi alusel. Tabel 4 kirjeldab andmebaasi tabelite semantikat.

Tabel 4. Andmebaasis olevate tabelite semantika

<b>Tabeli nimetus</b>	<b>Semantika</b>
category	Sisaldab andmeid toodete kategooria kohta.
manufacturer	Sisaldab andmeid toodete tootjate kohta.
country	Sisaldab andmeid andmepunkti riikide kohta.
product	Sisaldab andmeid toodete kohta.
data_point	Sisaldab andmeid toode ringmajanduse elutsükli kohta.
auth_user	Sisaldab andmeid kasutajatest.

Lisa 2 sees on kirjeldatud andmebaasi olemi–suhte diagramm. Visuaalne diagramm näitab seoseid andmebaasis, tabelite sisemist struktuuri ja illustreerib äridomeeni teadust andmebaasi kontekstis.

### 3.2.1 Tabel “category”

Tabel „category“ kirjeldab erinevate toodete kategooriaid. Antud tabelit on domeeni kontekstis kasutatud selleks, et illustreerida toodete jaotust erinevate gruppide alusel (näiteks telefonid, televiisorid, arvutid jne). Tabelis 5 on toodud välja antud tabeli struktuur.

Tabel 5. Andmebaasi tabeli “category” struktuur.

Veeru nimetus	Andmetüüp	Tähendus
id	INT	Unikaalne identifikaator (primaarvõti).
name	VARCHAR (255)	Kategooria nimetus.
created_datetime	TIMESTAMP	Kirje lisamise ajatempel.
modified_datetime	TIMESTAMP	Kirje muutmise ajatempel.

### 3.2.2 Tabel “manufacturer”

Tabel „manufacturer“ kirjeldab konkreetse toote tootjat. Antud tabelit on domeeni kontekstis kasutatud selleks, et illustreerida toodete jaotust tootja alusel (näiteks Apple, Stihl, LG jne). Tabelis 6 on toodud välja antud tabeli struktuur.

Tabel 6. Andmebaasi tabeli “manufacturer” struktuur.

Veeru nimetus	Andmetüüp	Tähendus
id	INT	Unikaalne identifikaator (primaarvõti).
name	VARCHAR (255)	Tootja nimetus.
created_datetime	TIMESTAMP	Kirje lisamise ajatempel.
modified_datetime	TIMESTAMP	Kirje muutmise ajatempel.



### 3.2.3 Tabel “product”

Tabel „product“ tabel kirjeldab üht toodet. Domeeni kontekstis on antud tabelit kasutatud selleks, et kirjeldada toodet, mis kuulub kindla tootja- ja tootekategooria alla (näiteks Apple iPhone 13 Pro Max 256GB, Apple MacBook Pro 14, jne). Tabelis 7 on toodud välja antud tabeli struktuur.

Tabel 7. Andmebaasi tabeli “product” struktuur.

Veeru nimetus	Andmetüüp	Tähendus
id	INT	Unikaalne identifikaator (primaarvõti).
name	VARCHAR (255)	Toote nimetus.
manufacturer_id	INT	Tootja ID (võõrvõti), kelle juurde kuulub antud toode.
category_id	INT	Kategooria ID, millesse kuulub antud toode.
created_datetime	TIMESTAMP	Kirje lisamise ajatempel.
modified_datetime	TIMESTAMP	Kirje muutmise ajatempel.

### 3.2.4 Tabel “data\_point”

Tabel „data\_point“ kirjeldab ühe toote elutsüklit ringmajanduse ahelas. Domeeni kontekstis on antud tabelit kasutatud selleks, et illustreerida üht toodet selle kasutuselevõtu ajast, kuni selle uuenduseni. Tabelisse on lisatud ärilised parameetrid, mis võimaldavad teha lisaanalüüsi, säilitades samas andmete seose. Tabelis 8 on toodud välja andmebaasi tabeli „data\_point“ struktuur.

Tabel 8. Andmebaasi tabeli “data\_point” struktuur.

Veeru nimetus	Andmetüüp	Tähendus
id	INT	Unikaalne identifikaator (primaarvõti).
product_id	INT	Toote ID, millele kuulub antud kirje (võõrvõti).
country_id	INT	Riigi ID, mille turu peal oli fikseeritud antud kirje.
usage_period	INT	Kasutamise periood. Mõõdu tüüp saab olla kokkulepitud.

retail_price	NUMERIC (9,2)	Toote lõplik hind, millega oli müüdnud kasutajale.
aftermarket_price	NUMERIC (9,2)	Toote hind peale kasutusperioodi, millega on müüdnud edasi partnerile.
retail_datetime	TIMESTAMP	Ajatempel, millal uus toode oli müüdnud tarbijale.
aftermarket_datetime	TIMESTAMP	Ajatempel, millal kasutatud toode oli müüdnud partnerile
created_datetime	TIMESTAMP	Kirje lisamise ajatempel.
modified_datetime	TIMESTAMP	Kirje muutmise ajatempel.

### 3.2.5 Tabel “country”

Tabel „country“ kirjeldab üht riiki. Domeeni kontekstis on seda kasutatud selleks, et illustreerida andmepunkti kindla turu peale. Tabelis 9 on toodud välja antud tabeli struktuur.

Tabel 9. Andmebaasi tabeli “country” struktuur.

Veeru nimetus	Andmetüüp	Tähendus
id	INT	Unikaalne identifikaator (primaarvõti).
name	VARCHAR (255)	Riigi nimetus.
code	CHAR(2)	ISO Alpha-2 kood.
created_datetime	TIMESTAMP	Kirje lisamise ajatempel.
modified_datetime	TIMESTAMP	Kirje muutmise ajatempel.

### 3.2.6 Tabel “auth\_user”

Tabel „auth\_user“ kirjeldab registreeritud kasutajat. Domeeni kontekstis on antud tabelit kasutatud, et illustreerida kasutajat, kellel on õigus kasutada antud ressursi. Tabelis 10 on toodud välja antud tabeli struktuur.

Tabel 10. Andmebaasi tabeli “auth\_user” struktuur.

Veeru nimetus	Andmetüüp	Tähendus
id	INT	Unikaalne identifikaator (primaarvõti).

username	VARCHAR (255)	Kasutaja nimi.
password	VARCHAR (100)	Kasutaja parooli räsi.
roles	VARCHAR (50)	Kasutaja rollid.
enabled	BOOLEAN	Kasutaja ressursi kasutamise õiguse omamise staatus.
created_datetime	TIMESTAMP	Kirje lisamise ajatempel.
modified_datetime	TIMESTAMP	Kirje muutmise ajatempel.

### 3.2.7 Lokaalne arendus

Lokaalseks arenduseks oli andmebaasi käivitamiseks kasutatud Docker'i konteinereid. Antud tehnoloogia võimaldab käivitada rakendusi isoleeritud konteinerites. Konteinerid sisaldavad kõiki vajalikke keskkonnasõltuvusi, et tarkvara saaks edukalt töötada sõltumata sellest, kus oli see käivitatud [12].

Andmebaasi käivitamiseks on moodustatud Docker Compose'i konfiguratsioon, mis võimaldab luua ja käivitada Dockeri võrku, kus eksisteerivad mitmed Docker konteinerid. Joonisel 5 on toodud välja Docker Compose'i konfiguratsioon.

```

version: "3.9"
services:
  postgres_db:
    image: "postgres:15.0-alpine"
    container_name: residual-value-forecast-db
    restart: always
    ports:
      - "5432:5432"
    environment:
      POSTGRES_DB: residual_value_forecast
      POSTGRES_USER: admin
      POSTGRES_PASSWORD: somepassword

```

Joonis 5. Docker Compose konfiguratsioon andmebaasi käivitamiseks.

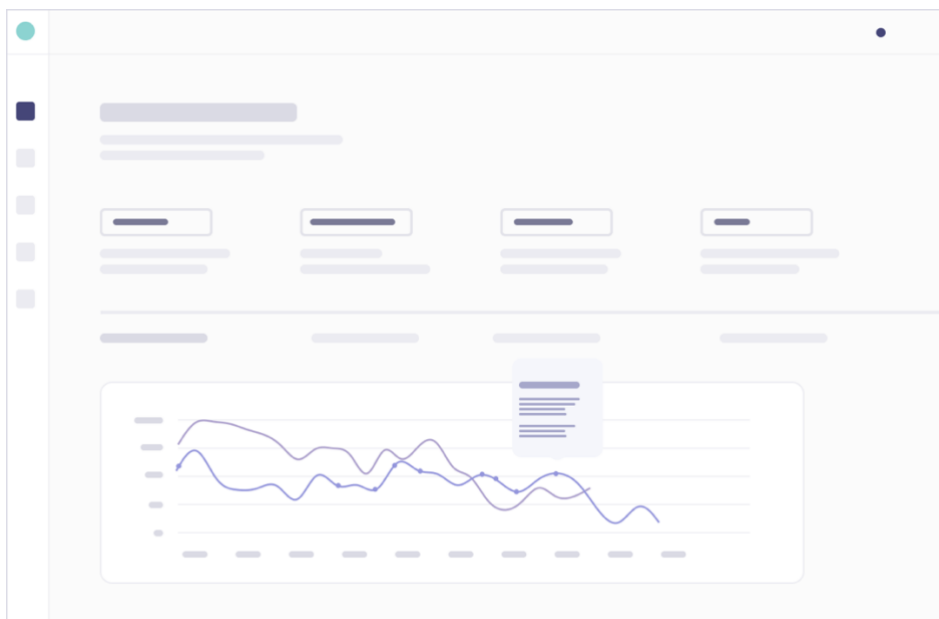
### 3.3 Kasutajaliides

Antud peatükk kirjeldab kasutajaliidese arendamise protsessi. Töö käigus sai loodud prototüüp, mille peale arendati kasutajaliides.

### 3.3.1 Prototüüp - traatraamide loomine

Kasutajaliidese arendamise esmaseks etapiks oli valitud traatraamide (ingl. k. *wireframing*) disainiprotsessi lähenemine. Antud tehnika võimaldab enne funktsionaalsuse implementeerimist illustreerida ja kirjeldada arendatava rakenduse ärivood ning valideerida seda tellijaga [13]. Visualiseerimise protsessi käigus luuakse visuaalne skeem, kirjeldatakse rakenduse struktuuri ja selle funktsionaalsust. Peamiseks eesmärgiks on luua abstraktne ja üldine vaade, mis võimaldab üle vaadata analüüsi käigus kirja pandud kriteeriumite täitmine [13].

Peatükis 2.1.2. kirjeldatud kasutajalugude alusel oli koostatud skoobi esimene vaade ja kirjeldatud ärivoog. Joonis 6 illustreerib vaadet, mis võimaldab kasutajal analüüsida ja vaadelda kindla tootega seotud andmeid. Traatraamide loomisel on kasutatud ettevõtte stiiliraamatu värve.



Joonis 6. Kasutajaliidese traatraam prototüübi ülevaatliku lehekülje kohta.

Esimese vaate ärivoog:

1. Kasutaja valib toodete kategooriad.
2. Kasutaja valib tootja ning toote mudeli.
3. Kasutaja valib riigi, mille turul ta soovib analüüsida toote andmeid.
4. Kasutaja näeb andmeid valitud toote kohta.

### 3.3.2 Projekti seadistamine

Arendusprotsess toimub jälgides eelnevalt tehtud analüüsi ja koostatud kasutajaliidese vaadet. Kasutajaliidese raamistiku valik on seotud lähtuvalt tänapäevaste parimate praktikatega arvestades Fairowni tehnilist domeeni. Fairown kasutab kliendipoolsete rakenduste ehitamiseks Vue.js raamistikku. Antud raamistik võimaldab arendada ühelehelisi rakendusi (ingl. k. *single page application*). Nimetatud rakenduse tüübid laetakse keskkonnas ainult ühe korra ning muudatused toimuvad dünaamiliselt vastavalt kasutaja käitumisele.

Vue.js 3 projekt loodi lokaalselt tüüpprojekti põhjal. Vue pakub palju funktsionaalsust projekti alustusel, kuid spetsiifilisem funktsionaalsus tuleb lisada eraldi teekidega. Projekti olid lisatud järgnevad teegid:

- @fairown/subscription-frontend-common (sisemiselt ehitatud komponentide raamistik),
- @tanstack/vue-query,
- apexcharts,
- axios,
- vue-router.

### 3.3.3 Arhitektuur

Kliendipoolse rakenduse arhitektuuriks on valitud kombineeritud lähenemine, mis hõlmab endas nii kihilist jaotamist kui ka funktsionaalsete osade grupeerimist. Kihiline jaotus struktureerib projekti koostisosi vastavalt nende rollile projekti vaates. Iga kiht vastutab konkreetse ülesande täitmise eest (näiteks andmebaasi kiht, äriline kiht, visuaalne kiht) ja suhtleb teiste kihtidega läbi kindlaksmääratud reeglite [14].

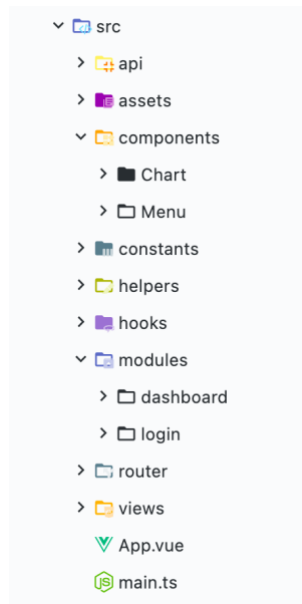
Funktsionaalne arhitektuur jaotab projekti vastavalt äridomeenile ja kasutajafunktsionaalsusele. Domeenid on teineteisest eraldatud ja illustreerivad kindlat kasutajalugu [14].

Joonis 7 illustreerib arhitektuurselt kombineeritud lähenemist. Juurkaustas on võimalik eristada projektiosade kihilist jaotumist ning kihtide sees asuvad domeeni- ja funktsionaalse jaotusega komponendid. Antud jaotus võimaldab võtta mõlema

lähenemise parimad omadused mis muudab projekti hästi skaleeritavaks nii horisontaalselt, kui ka vertikaalselt.

Joonis 7 näitab, et projekt on jaotatud järgnevateks kaustadeks:

- **api:** sisaldab funktsionaalset *axios* klienti, mis võimaldab äriteenuse taustsüsteemiga suhtlemist. Kaustas on hoitud ka API päringutega seotud TypeScript tüübid;
- **assets:** sisaldab staatilisi varasid, näiteks globaalseid CSS faile ja ikoone;
- **components:** sisaldab projekti baaskomponente – projekti raames on komponendid stiliseeritud kasutajaliidese baaselementitena, mis on kasutatud erinevates kohtades (menüü komponent, diagrammi element jne);
- **constants:** sisaldab globaalseid konstantseid väärtusi;
- **helpers:** sisaldab globaalseid abifunktsioone;
- **hooks:** sisaldab spetsiifilisi kohandatud konkse (ingl. k. *hooks*);
- **modules:** sisaldab projekti mooduleid – projekti raames on komponendid suuremad funktsionaalsed tükid, mis omavad spetsiifilist kasutust;
- **router:** kirjeldab projekti sees kasutatavaid marsruute (*vue router* seadistus);
- **views:** sisaldab rakenduse vaateid – vaated on marsruudile määratud baaskomponendid.



Joonis 7. Kasutajaliidese projektikeskne arhitektuur.

### 3.3.4 Peavaate mooduli struktuur

Arenduse protsess on kirjeldatud peavaate loomise alusel. Peavaate komponendid on loogiliselt grupeeritud antud kujul, et moodustada täiuslik ärivoog:

- **Peavaade**

**Tee:** /dashboard

**Komponent:** DashboardWrapper.vue

**Kirjeldus:** Pealtvaate sisepunkt, mis laeb menüüs olevaid vaateid. Antud komponent täidab ka ümbrise rolli, lisades pealtvaatele globaalseid vajalikke komponente (menüü element, ettevõtte logo jne).

- **Kodu komponent (statistika vaatamine)**

**Tee:** /dashboard/home

**Komponent:** HomeView.vue

**Kirjeldus:** Kodu komponendi peamine eesmärk on võimaldada kasutajal vaadata toodetega seotud statistikat. Antud vaade sisaldab toodete kategooriaid, tootjaid ja mudelite valimist. Kõik andmed on küsitud serveri poolt komponendi seadistamise ajal. Peale toote valikut on kasutajale laetud sisemised komponendid, millele on edastatud valitud toote unikaalne identifikaator, võimaldades küsida valitud toote kohta andmed.

- **Statistika komponent**

*Tee:* /dashboard/home/price-history/{id}

*Komponent:* PriceHistory.vue

*Kirjeldus:* Statistika komponendi eesmärk on näidata kasutajale valitud tootega seotud hinnaajalugu. Komponent saab sisendiks toote identifikaatori, mille alusel pärib vajaliku informatsiooni serveri poolt.

- **Andmete jaotuse komponent**

*Tee:* /dashboard/home/data-distribution/{id}

*Komponent:* DataDistribution.vue

*Kirjeldus:* Andmete jaotuse komponendi eesmärk on näidata kasutajale andmete olemasolu kindlal ajaperioodil. Komponent saab valitud toote identifikaatori, mille alusel pärib vajaliku informatsiooni serveri poolt.

- **Prognoosimise komponent**

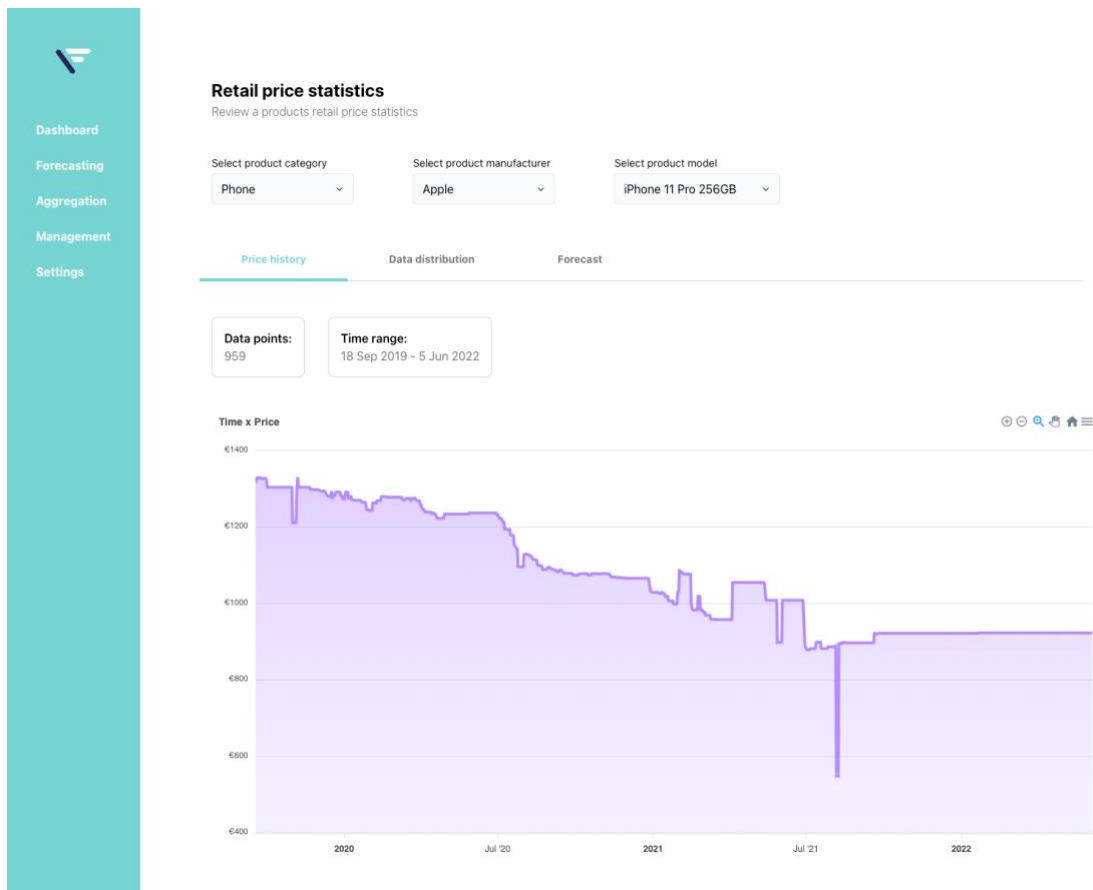
*Tee:* /dashboard/home/forecasting/{id}

*Komponent:* Forecasting.vue

*Kirjeldus:* Prognoosimise komponendi eesmärk on näidata kasutajale hinna prognoosi.

Joonis 8 illustreerib arenduse käigus saavutatud vaadet. Selles vaates on võimalik valida tooteid, nende mudeleid ja analüüsida nende hinnakõikumisi kindla ajaperioodi vältel.





Joonis 8. Peavaates asuv ülevaatlik statistika vaade.

### 3.4 Rakendusliides

Käesolevas peatükis on kirjeldatud rakendusliidese arendamise protsess.

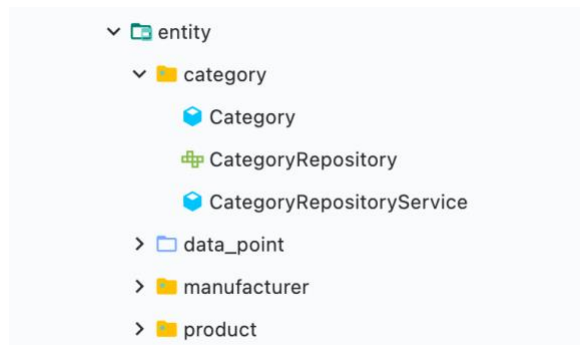
#### 3.4.1 Projekti seadistamine ja tehnoloogiad

Analüüsi käigus sai tehtud kindlaks, et Fairowni rakendusliideste peamiseks kasutatavaks keeleks on Java ning liidese raamistikuks on Spring Boot. Antud tehnoloogia on sobilik püstitatud eesmärgi täitmiseks.

#### 3.4.2 Arhitektuur

Rakendusliidese arhitektuur on tihedalt seotud domeenipõhise disainiga. Käesoleva lahenduse raames olulisemaks arhitektuuri komponendiks peetakse entiteete (ingl. k. *entity*), mis sisaldab endas domeenipõhist teadmist äriprotsessidest. Entiteet põhineb eelnevalt tehtud andmebaasi mudelite peal [11]. Joonis 9 illustreerib kategooria entiteeti, mis koosneb kolmest peamisest elemendist:

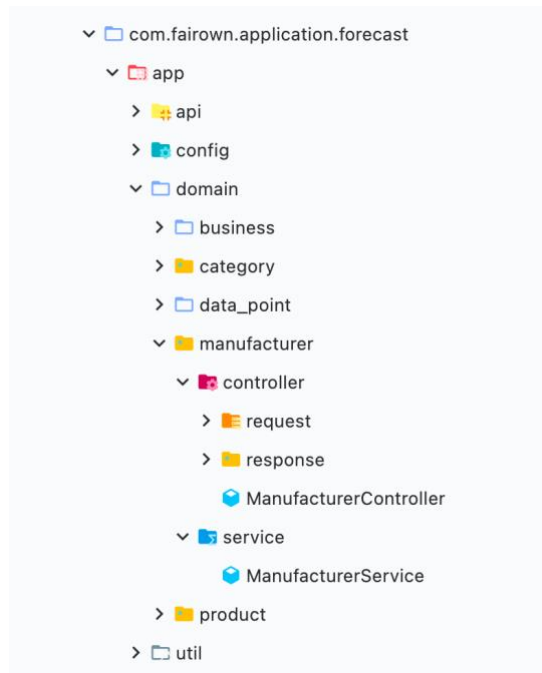
1. Entiteedi mudel (*Category*) - mudel, mis kirjeldab andmebaasi kategooria objekti.
2. Repositoorium (*CategoryRepository*) - kategooria liides, mis tekitab keskse andmete pärimise koha.
3. Repositooriumi teenus (*CategoryRepositoryService*) - teenus, mis abstraherib ja isoleerib repositooriumi kasutamise välisressursidele.



Joonis 9. Rakendusliidese entiteetidid.

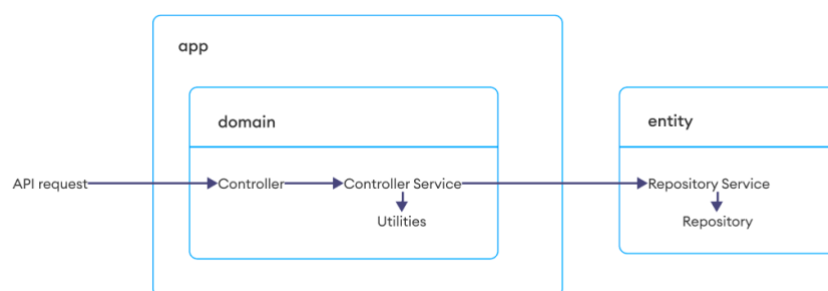
Rakenduse põhiline osa on inspireeritud puhta arhitektuuri (ingl. k. *clean architecture*) filosoofiast. Robert C. Martin kirjeldab mitmeid lähenemisviise, mis muudavad arhitektuuri modulaarsemaks ja skaleeritavamaks, sealhulgas:

1. Vastutuse eraldamine (ingl. k. *Separation of Concerns*) - arhitektuuri komponendid on jaotatud ja eraldatud vastutuse kihtidesse või komponentidesse. Joonis 10 illustreerib rakenduse domeeni (ingl. k. *domain*) isoleeritust. Iga domeeni mudel implementeerib kindlat ärikonteksti osa, mis koosneb eraldatud kihtidest.



Joonis 10. Isoleeritud rakendusliidese projekti arhitektuur.

2. Sõltuvuse pöördus (ingl. k. *Dependency inversion*) – arhitektuur kasutab liidestusi ja abstraksioone. Kõrgema taseme moodulid ja komponendid kasutavad ainult alamkomponente [15]. Arendatud rakendus implementeerib antud printsiipi, jaotades arhitektuursed osad erinevateks kihtideks. Joonis 11 illustreerib komponentide taseme osade jaotust.



Joonis 11. Arhitektuuri osade jaotus kihtidena.

3. Fookus ärioloogikale (ingl. k. *focus on business logic*) – arhitektuur peab prioritseerima ärioloogikat ja äri vajaduse mitte-tehnilist poolt. Arendatud

rakendus on väga tihedalt seotud DDD filosoofiaga, mille peamine fookus on rakenduse ehitamine kooskõlastades see ärikontekstiga [11, 16].

### 3.4.3 Kasutusjuhtumid

Selleks, et süsteemi oleks võimalik kasutada väliselt ning süsteem täidaks ärilisi vajadusi, on vajalik seadistada rakenduspõhiselt URI (ingl. *k uniform resource identifier*). Iga unikaalne identifikaator on seatud rakenduse ressursiga ja täidab üht konkreetset kasutusjuhtu (ingl. k. *use-case*). Rakenduse URI pöördumisel on päring töödeldud kontrolleri (ingl. k. *controller*) tasandil, mis omakorda kasutab kontrolleri-poolset teenust (ingl. k. *service*) ja täidab määratletud ärivajadust. Tabelis 11 on kirjeldatud andmekirje (ingl. k. *data\_point*) domeeni funktsionaalsus koos URI kirjeldusega.

Tabel 11. Andmekirje domeeni URI koos meetodite ja nende kirjeldustega.

URI	Meetod	Kirjeldus
/data-point?countryId={id}&productId={id}	GET	Tagastab kõik andmekirjde põhinedes riigi ja toote identifikaatoritele.
/data-point?countryId={id}&productId={id}	POST	Lisab andmekirje kindla toote juurde, määratles riigi.
/data-point?id={id}	GET	Tagastab kindla andmekirje.
/data-point	PUT	Muudab kindlat andmekirjet.
/data-point?id={id}	DELETE	Kustutab kindla andmekirje.

## 4 Tulemused ja analüüs

Käesoleva töö eesmärgid on sõnastatud peatükis 1.2. Töö peamiseks eesmärgiks oli luua jätkusuutlik ja automatiseeritud andmete analüüsi prototüüp Fairowni jaoks, mis võimaldaks lõppkasutajale kohaldada andmetel analüüsi ja prognoosida toodete hinna muutumist ajas ning nende jääkväärtuseid.

Antud töö raames vaatles autor olemasoleva protsessi toimimist ja töötas välja metoodika selle automatiseerimiseks. Koostöö raames on autor teinud mahuka analüüsi, mille tagajärjel sai kirjeldatud olemasolev protsess ning määratud kliendipoolsed vajadused ja nõuded. Analüüsi tulemusena kirjeldas autor süsteemi vajalikku funktsionaalsust, struktureeris seda ja jaotas arendustsükli skoopideks, mille peal oli töötas välja arengustrateegia jätkusuutlikumaks platvormi edasiarenduseks.

Kliendipoolseks oluliseks vajaduseks oli toodete jääkväärtuse prognoosimine, siis seetõttu said analüüsi käigus vaadeldud võimalikud meetmed antud funktsionaalsuse rakendamiseks. Töö käigus sai kindlaks määratud andmete domeen, mille fookus oli suunatud prognoosimise eesmärgi teostamiseks. Samuti olid vaadeldud erinevad prognoosimise algoritmid ning välja töötatud võimalik voog antud eesmärgi saavutamiseks.

Praktiline osa põhines tugevalt analüüsil, ning järgis skoobis seatud eesmärkide saavutamist. Antud töö raames olid loodud peavaate traatraamid ning nende peal tehtud arendustöö, arvestades eelnevalt määratud vajadusi. Domeen ja ärifunktsionaalsus said implementeeritud rakendusliidese poolel. Töö tulemusena saavutati tugev projekti põhi koos funktsionaalsusega, mis võimaldab illustreerida ja teha toodetega seotud andmete analüüsi.

Arendustöö käigus sai tehtud kindlaks, et jääkväärtuse prognoosimise funktsionaalsuse arendamine on eraldiseisvalt väga mahukas protsess ning vajab suuremat arendustööd ja testimist. Autor ei ole varem kokku puutunud andmete analüüsiga ja ennustamise mudelitega, mis tekitas raskuse õige mudeli valikus ja selle arenduses. Tunduv raskus oli ka seotud keerulise Fairowni domeeniga, mille analüüs ja kaardistamine oli väga mahukas töö. Lõputöö kirjutamise hetkel jätkab autor prognoosimise funktsionaalsuse elluviimist, kasutades eelnevalt tehtud analüüsi.

Lõputöö raames loodud arendustöö vajab täiendavat testimist koos reaalsete ning tõetruude andmetega. Arendatud põhi on koostatud vastavalt tellija kasutajalugude kirjeldamisele, kuid Fairowni äriprotsessid ei ole jõudnud faasi, milles oleks võimalik lahendust täismahus testida. Reaalsete andmete testimine võib esile tuua uusi võimalusi, kitsaskohti ja lähenemisi. Autor soovib jätkata projekti arendamist peale antud töö esitamist ning viia ellu ja testida antud töös kirjeldatud funktsionaalsust.

Käesoleva lõputöö püstitatud eesmärgid võib suuremas osas arvestada täidetuna. Antud töö raames autor on teinud mitmekesise äriprotsessi analüüsi, kirjeldanud lahenduse võimalikke lähenemisviisi, pannud paika arendusplaani ning teinud prototüübi, mille põhjal on võimalik jätkata analüüsi ja arendusprotsessi.

## Kokkuvõte

Käesoleva lõputöö raames on autor teinud ettevõtte Fairowni äriprotsessi automatiseerimist ja täiendamist. Fairowni üheks olulisemaks äriprotsessiks on toodete jääkväärtuse arvutamine ning sellega seotud andmete analüüs. Ettevõtte kasvuga tekkisid spetsiifilised protsessid, mille manuaalne analüüs oli raskendatud. Vastavalt probleemile oli vajadus teha analüüs ning pakkuda võimalikku, automatiseeritud lahendust.

Antud töö analüüsi osas on autor kasutanud erinevaid lähenemise tehnikaid informatsiooni kogumiseks. Toetudes püstitatud eesmärkidele, on autor teinud tihedat koostööd Fairowni arendusmeeskonnaga. Tulemusena olid määratud ärivajadused ja koostatud kasutajalood. Põhinedes vajadustele on autor teinud mitmeid uuringuid ja välja töötanud lahenduse funktsionaalsust, mis olid valideeritud Fairowni poolt. Analüüsi käigus autor on uurinud võimalikke jääkväärtuse prognoosimise mudeleid ja lähenemisi. Antud töö praktilises osas on töö autor teinud arendustööd, toetudes tugevalt eelnevalt tehtud analüüsile. Autor lõi andmebaasi mudelid, mis olid kooskõlastatud ärivajadustega. Põhinedes eelnevalt määratud funktsionaalsusele, said töö raames koostatud traatraamid, mis andsid kliendile ülevaate loodavast prototüübist. Peale nende valideerimist oli välja töötatud kliendipoolne kasutaja- ja rakendusliides. Autor lähenes arendusprotsessile kaaludes erinevaid meetodikaid ja kasutades selleks parimaid võimalikke praktikaid.

Töö koostamisel omandas autor tugevaid praktilisi teadmisi kliendiga koos töötamisest. Autor sai teada, kuidas läbi viia arutusprotsesse, koguda nõudeid, valideerida neid ja määrata skoopt, mis tooks kliendile võimalikult palju lisaväärtust töö varajases etapis. Lisaks õppis autor prognoosimise mudelite spetsiifikat ja erinevaid tehnilise lähenemisi arendusprotsessile.

Käesoleva töö tulemusena sai arendatud prototüüp, mis võimaldab teha toodete hinna analüüsi mudeli- ja riigipõhiselt. Kirjeldatud olid võimalikud lähenemisviisid ja meetodid, mis kooskõlastati praktilise rakendusega ja annavad tugeva põhja edasi arendamisele ning platvormi täiendamiseks.

## Kasutatud kirjandus

- [1] [Võrgumaterjal]. Available: <https://www.fairown.com>. [Kasutatud 23 03 2023].
- [2] R. Liive, „Estonian FinTech Fairown allows tech giants like Apple to sell products as a service,“ Juuli 2022. [Võrgumaterjal]. Available: <https://investinestonia.com/estonian-fintech-fairown-allows-tech-giants-like-apple-to-sell-products-as-a-service>. [Kasutatud 22 03 2023].
- [3] „What is Agile?,“ [Võrgumaterjal]. Available: <https://investinestonia.com/estonian-fintech-fairown-allows-tech-giants-like-apple-to-sell-products-as-a-service>. [Kasutatud 24 03 2023].
- [4] M. Rehkopf, „User stories with examples and a template,“ [Võrgumaterjal]. Available: <https://www.atlassian.com/agile/project-management/user-stories>. [Kasutatud 24 03 2023].
- [5] „100% töötava seadme tagasiostu hinnakiri,“ [Võrgumaterjal]. Available: <https://www.mobipunkt.ee/vana-seade-ueeks>. [Kasutatud 27 03 2023].
- [6] „Mobiiltelefon Apple iPhone 12 128GB Black,“ [Võrgumaterjal]. Available: <https://www.hinnavaatlus.ee/1930807/apple-iphone-12-128gb-black>. [Kasutatud 28 03 2023].
- [7] „Mobiiltelefon Apple iPhone 13 128GB Midnight,“ [Võrgumaterjal]. Available: <https://www.hinnavaatlus.ee/2197459/apple-iphone-13-128gb-midnight>. [Kasutatud 28 03 2023].
- [8] A. Bajaj, „How to Select a Model For Your Time Series Prediction Task [Guide],“ 19 April 2023. [Võrgumaterjal]. [Kasutatud 30 03 2023].
- [9] „What are microservices?,“ [Võrgumaterjal]. Available: <https://microservices.io>. [Kasutatud 02 04 2023].
- [10] „What Is A RESTful API?,“ [Võrgumaterjal]. Available: <https://aws.amazon.com/what-is/restful-api>. [Kasutatud 04 04 2034].
- [11] V. Vernon, Implementing Domain-Driven Design, 2013.



- [12] „Use containers to Build, Share and Run your applications,“ [Võrgumaterjal].  
Available: <https://www.docker.com/resources/what-container>. [Kasutatud 2023 04 2023].
- [13] J. Hannah, „What Exactly Is Wireframing? A Comprehensive Guide,“ 19 April 2023. [Võrgumaterjal]. Available: <https://www.oreilly.com/library/view/software-architecture-patterns/9781491971437/ch01.html>. [Kasutatud 14 04 2023].
- [14] M. Richards, „Software Architecture Patterns,“ Veebruar 2015. [Võrgumaterjal].  
Available: <https://www.oreilly.com/library/view/software-architecture-patterns/9781491971437/ch01.html>. [Kasutatud 16 04 2023].
- [15] R. C. Martin, Clean Code: A Handbook of Agile Software Craftsmanship, 2008.
- [16] T. Karagiannis, „Structure by Type vs Feature,“ 20 November 2021.  
[Võrgumaterjal]. Available: <https://maestros.io/structure-by-type-vs-feature>.  
[Kasutatud 16 04 2023].

## **Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks**

Mina, Artjom Sarapov

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Automaatse andmepõhise toodete jääkväärtuse arvutamise prototüüp ettevõtte Fairown näitel“, mille juhendaja on Brita Moorus
  - 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.

24.04.2023

## Lisa 2 – Jääkväärtuse analüüsimise ja prognoosimise olemissuhte diagramm



# Lisa 3 – Kliendi rakenduse tee ja komponendi struktuuri seose näide

