

TALLINNA TEHNIKAÜLIKOOL

Infotehnoloogia teaduskond

Anton Meenov 222917IAIB

# **Veebipõhise inventuurirakenduse arendamine**

Bakalaureusetöö

Juhendaja: Siim Rebane

BSc

Tallinn 2025

## **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: Anton Meenov

04.06.2025

## Annotatsioon

Käsitsi tehtavad inventuuriprotsessid on sageli ebatõhusad, alid inimlikele vigadele ning neil puudub reaajas andmete nähtavus, mis takistab tõhusat varude haldamist ja täpset aruandlust. Käesolev lõputöö järgib disaineaduse uurimismeetodit. Töö ajendiks olid Eesti tootmisettevõttes ilmnenud varude haldamise probleemid ning selle eesmärk oli kujundada ja arendada veebipõhine rakendus, et muuta inventuuriprotsess sujuvamaks ja digitaalsemaks. Arendatud IT-artefakti tehnoloogiavirn koosneb .NET-i tagarakenduse rakendusliidest, TypeScriptis loodud Vue.js esirakendusest ja PostgreSQL-i andmebaasi haldussüsteemist.

Rakenduse põhifunktsioonide hulka kuuluvad rollipõhine autentimine ja juurdepääsu kontroll, kaubaartiklite põhiandmete importimine standardsest ERP (ettevõtte ressursiplaneerimise) süsteemist Exceli eksportimisega, inventuurilehtede loomine ja haldamine ning inventuurikirjete esitamise lubamine. Rakendus toetab nii käsitsi andmete sisestamist kui ka kaamerapõhist koodilugemist kaubaartiklite tuvastamiseks. Oluline ärilooika on jõustatud serveripoolselt ja lõplikke inventuuriandmeid saab eksportida struktureeritud Exceli vormingusse. Arendusprotsess hõlmas automatiseeritud testimist ja konveiereid GitHub Actionsi kaudu Dockeri konteinerdamisega.

Peamine tulemus on toimiv veebirakendus, mis rakendab edukalt põhilist inventuuri töövoogu. Läbiv testimine valideeris kõik peamised kasutajateed, näidates rakenduse elujõulisust. Väljatöötatud IT-artefakt pakub praktilist lahendust, et leevendada käsitsi inventuuriga seotud ebatõhusust ja vigu. See on tsentraliseeritud rollipõhine rakendus mobiilse koostoime võimalusega, mis parandab täpsust ja tegevuse tõhusust. Rakendus täidab edukalt oma esialgsed eesmärgid ja loob aluse edasistele täiustustele, näiteks otsesele ERP-süsteemi lõimimisele.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 35 leheküljel, 5 peatükki, 10 joonist, 1 tabel.

## **Abstract**

### **Development of a Web Application for Stocktaking**

Manual stocktaking processes are frequently inefficient, prone to human error, and lack real-time data visibility, thereby hindering effective stock management and accurate reporting. This thesis follows a design science research methodology. Motivated by inventory management challenges identified within an Estonian manufacturing company, its aim was to design and develop a web-based application to streamline and digitize the stocktaking process. The developed IT artifact utilizes a technology stack comprising a .NET backend application programming interface, a Vue.js frontend developed in TypeScript, and a PostgreSQL database management system.

Key functionalities of the application include role-based authentication and access control, importing item master data from standard ERP (Enterprise Resource Planning) system Excel exports, creating and managing stocktaking sessions, and enabling count entry submission. The application supports both manual data input and camera-based code scanning for item identification. Essential business logic is enforced server-side, and finalized stocktaking data can be exported to a structured Excel format. The development process incorporated automated testing, and pipelines via GitHub Actions with Docker containerization.

The primary result is a functional web application that successfully implements the core stocktaking workflow. End-to-end testing validated all primary user paths, demonstrating the application's viability. The developed IT artifact offers a practical solution to mitigate the inefficiencies and errors associated with manual stocktaking. It is a centralized, role-based application with mobile interaction capabilities, thereby improving accuracy and operational efficiency. The application successfully meets its initial objectives and establishes a foundation for future enhancements, such as direct ERP system integration.

The thesis is in Estonian and contains 35 pages of text, 5 chapters, 10 figures, 1 table.

## Lühendite ja mõistete sõnastik

|                       |                                                                                                                                           |
|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------|
| .NET                  | Microsofti toetatav tasuta ja avatud lähtekoodiga rakenduste platvorm [1].                                                                |
| API                   | Rakendusliides ( <i>Application Programming Interface</i> ); kogum funktsioone, mida saab käivitada programmiga [2].                      |
| Arhitektuur           | Süsteemi põhikontseptsioonid või -omadused ta keskkonnas, kehastatuna ta elementides, seostes ning kavanduse ja arengu põhimõtetes [3].   |
| ASP.NET Core          | Platvormideülene raamistik veebirakenduste loomiseks [4].                                                                                 |
| ASP.NET Core Identity | Liikmelisuse süsteem ASP.NET Core-i veebirakenduste loomiseks, mis hõlmab liikmelisust, sisselogimist ja kasutajaandmeid [5].             |
| Axios                 | Axios on lubaduspõhine HTTP klient Node.js-i ja brauseri jaoks [6].                                                                       |
| CI/CD                 | Pidevintegratsioon ja pidevvalmidus ( <i>Continuous Integration and Continuous Delivery</i> ) [7].                                        |
| C#                    | Üldotstarbeline objektorienteeritud programmeerimiskeel [8].                                                                              |
| Docker                | Avatud platvorm rakenduste arendamiseks, saatmiseks ja käitamiseks, mis annab võimaluse pakendada ja käivitada rakendust konteineris [9]. |
| Eesrakendus           | Rakenduse või veebisaidi kasutajapoolne osa, millega kasutaja otse suhtleb [10].                                                          |
| EF Core               | Entity Framework Core; avatud lähtekoodiga ja platvormideülene ORM [11].                                                                  |
| ERP                   | Ettevõtte ressursside plaanimine [12].                                                                                                    |
| Excel                 | Arvutustabeli rakendus (Microsoft Excel) [13].                                                                                            |
| Git                   | Tasuta ja avatud lähtekoodiga hajutatud versioonikontrolli tööriist [14].                                                                 |
| GitHub                | Veebipõhine platvorm Git repositooriumite majutamiseks ja koostööks.                                                                      |
| GitHub Actions        | Tööriist tarkvaraarenduse töövoogude automatiseerimiseks GitHubis.                                                                        |
| Haldur                | Selles töös määratletud kasutajaroll inventuurisüsteemis, kes võib luua ja hallata inventuure ning eksportida tulemusi.                   |
| HTTP                  | Hüperteksti edastusprotokoll ( <i>Hypertext Transfer Protocol</i> ).                                                                      |

|                  |                                                                                                                                                                        |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Inventuur        | Tegevus, mille käigus loetakse ettevõtte omanduses olevad või poes müügil olevad kaubad või materjalid teatud ajahetkel [15].                                          |
| Inventuurileht   | Selles töös kasutatav termin, mis tähistab dokumenti või süsteemi-osa, mis koondab ühe konkreetse inventuuri loendatavad artiklid ja nende kogused.                    |
| Iteratiivarendus | Süsteemiarenduse meetod, mis tükeldab suure süsteemi osadeks, igalühel neist oma täielik elutsükkel [16].                                                              |
| JS               | JavaScript; objektorienteeritud programmeerimiskeel dünaamilise veebis-<br>sisu skriptide loomiseks [17].                                                              |
| Konteiner        | Lõdvalt isoleeritud keskkonnad [9].                                                                                                                                    |
| LTS              | Kestustugi ( <i>Long-Term Support</i> ); tootearenduse poliitika, mis taot-<br>leb kasutamise stabiilsust [18].                                                        |
| Lugeja           | Selles töös määratletud kasutajaroll inventuurisüsteemis, kelle<br>peamine ülesanne on inventuurikirjete sisestamine.                                                  |
| Läbiv testimine  | Süsteemi testimine koos ta välisliidestega, kasutaja vaatepunktist<br>(end-to-end testing) [19].                                                                       |
| nginx            | HTTP veebiserver ja pöördproksi [20].                                                                                                                                  |
| ORM              | Objekt-relatsioonvastendus ( <i>Object-Relational Mapping</i> ) [21].                                                                                                  |
| Pinia            | Pinia on Vue.js salvestusteeke, mis võimaldab olekut komponentide<br>ja lehtede vahel jagada [22].                                                                     |
| PostgreSQL       | Avatud lähtekoodiga objekti-relatsioonandmebaaside süs-<br>teem [23].                                                                                                  |
| Pöördproksi      | Liik vaheservereid, vahendab välisklientide päringuid siseserveri-<br>tele, nii et kliendile tunduvad vastused tulevat pöördproksilt en-<br>dalt [24].                 |
| REST             | Hajustöötamise paradigma, taotleb veebi nüüdisnõuetele sobivatarhi-<br>tektuuristiili ja ühtset liidest <i>Representational State Transfer</i> [25].                   |
| SaaS             | Tarkvara teenusena ( <i>Software as a Service</i> ); tarkvara tarnimise<br>meetod, mis võimaldab kliendil saada kaugpääsuga kasutada tee-<br>nuseandja rakendusi [26]. |
| Scrum            | Välearenduse meetod [27].                                                                                                                                              |
| Tagarakendus     | Veebisaidi või rakenduse osa, mida kasutajad ei näe [28].                                                                                                              |
| Tehnoloogiavirn  | Rakenduse arendamiseks kasutatavate tehnoloogiate kogum [29].                                                                                                          |
| TS               | TypeScript; tugevalt tüübitud programmeerimiskeel, mis põhineb<br>JavaScriptil [30].                                                                                   |
| Tsoon            | Inventuuri läbiviimiseks määratletud füüsiline või loogiline ala.                                                                                                      |
| UI               | Kasutajaliides ( <i>User Interface</i> ); tarkvara ja riistvara ühend, mis<br>võimaldab kasutaja interaktsiooni arvutiga [31].                                         |
| Vite             | Kiire eesrakenduse ehitustööriist [32].                                                                                                                                |

Vue.js

Veebikasutajaliideste loomise toimiv raamistik [33].

## Sisukord

|       |                                                   |    |
|-------|---------------------------------------------------|----|
| 1     | Sissejuhatus.....                                 | 11 |
| 1.1   | Probleemipüstitus .....                           | 11 |
| 1.2   | Töö eesmärk.....                                  | 12 |
| 1.3   | Töö ulatus .....                                  | 13 |
| 2     | Metoodika.....                                    | 15 |
| 2.1   | Uurimisparadigma .....                            | 15 |
| 2.2   | Nõuete analüüs .....                              | 15 |
| 2.2.1 | Funktsionaalsed nõuded.....                       | 15 |
| 2.2.2 | Mittefunktsionaalsed nõuded .....                 | 16 |
| 2.3   | Rakenduse arhitektuur ja tehnoloogiate valik..... | 17 |
| 2.3.1 | Tagarakendus.....                                 | 18 |
| 2.3.2 | Eesrakendus .....                                 | 18 |
| 2.3.3 | Andmebaas.....                                    | 19 |
| 2.3.4 | Pidevintegratsioon ja pidevvalmidus.....          | 19 |
| 2.3.5 | Konteineriseerimine ja juurutamine.....           | 19 |
| 2.4   | Alternatiivsete lahenduste analüüs.....           | 20 |
| 2.4.1 | Arendusmetoodika .....                            | 21 |
| 2.5   | Valideerimine .....                               | 22 |
| 3     | Tulemused.....                                    | 23 |
| 3.1   | Loodud rakenduse ülevaade.....                    | 23 |
| 3.2   | Kasutajaliides ja funktsionaalsus.....            | 23 |
| 3.2.1 | Autentimine.....                                  | 23 |
| 3.2.2 | Artikliandmete import .....                       | 24 |
| 3.2.3 | Inventuurilehe loomine ja haldamine .....         | 25 |
| 3.2.4 | Inventuuriandmete eksport .....                   | 27 |
| 3.2.5 | Inventuurilehe valimine.....                      | 27 |
| 3.2.6 | Käsitsi koguse sisestamine .....                  | 28 |



|       |                                                                                                       |    |
|-------|-------------------------------------------------------------------------------------------------------|----|
| 3.2.7 | Skaneerimisega koguse sisestamine .....                                                               | 29 |
| 3.3   | Testimine ja valideerimine.....                                                                       | 31 |
| 3.3.1 | Funktsionaalse testimise tulemused .....                                                              | 31 |
| 3.3.2 | Kasutatavuse ja kasutajaliidese vaatlused .....                                                       | 32 |
| 3.3.3 | Koodilugemise jõudluse tulemused .....                                                                | 32 |
| 3.3.4 | Andmete valideerimise tulemused .....                                                                 | 32 |
| 4     | Analüüs ja arutelu .....                                                                              | 33 |
| 4.1   | Tulemuste analüüs .....                                                                               | 33 |
| 4.2   | Lahenduse tugevused.....                                                                              | 36 |
| 4.3   | Lahenduse piirangud .....                                                                             | 38 |
| 4.4   | Mõtiskelu arendusprotsessist .....                                                                    | 40 |
| 4.4.1 | Ilmnenud väljakutsed.....                                                                             | 40 |
| 4.4.2 | Tehnoloogiarahva hindamine.....                                                                       | 41 |
| 4.4.3 | Õppetunnid .....                                                                                      | 41 |
| 4.5   | Tulevased tööd.....                                                                                   | 42 |
| 4.5.1 | Funktsionaalsed täiustused .....                                                                      | 42 |
| 4.5.2 | Tehnilised täiustused.....                                                                            | 43 |
| 4.5.3 | Juurutus- ja integratsiooniparandused .....                                                           | 43 |
| 5     | Kokkuvõte.....                                                                                        | 45 |
|       | Kasutatud kirjandus .....                                                                             | 46 |
|       | Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks..... | 50 |

## Jooniste loetelu

|            |                                                                                                |    |
|------------|------------------------------------------------------------------------------------------------|----|
| Joonis 1.  | Rakenduse arhitektuur ja peamised komponendid Dockeri keskkonnas. ...                          | 17 |
| Joonis 2.  | Sisselogimisliides.....                                                                        | 24 |
| Joonis 3.  | Artikliandmete importimise UI. ....                                                            | 25 |
| Joonis 4.  | Artiklite nimekiri. ....                                                                       | 25 |
| Joonis 5.  | Inventuurilehte haldamise vaade.....                                                           | 26 |
| Joonis 6.  | Uue inventuurilehe loomise vorm.....                                                           | 26 |
| Joonis 7.  | Inventuurilehe andmete vaade.....                                                              | 27 |
| Joonis 8.  | Inventuurilehe valiku vaade lugejale. ....                                                     | 28 |
| Joonis 9.  | Inventuurikirje käsitsi sisestamise liides, näidates artikli valikut ja koguse sisestust. .... | 29 |
| Joonis 10. | Loendusliides, kus on valitud artikli skaneerimise nupp.....                                   | 30 |

## **Tabelite loetelu**

|          |                                        |    |
|----------|----------------------------------------|----|
| Tabel 1. | Alternatiivsete lahenduste võrdus..... | 20 |
|----------|----------------------------------------|----|

# 1 Sissejuhatus

Inventuur on ülioluline protsess igas suuruses ettevõtetele, mõjutades otseselt nende kasumlikkust ja tegevustõhusust [34]. Täpsed varude andmed on olulised tõhusa varude kontrolli jaoks, tagades piisava varude taseme nõudluse rahuldamiseks ilma liigsete varudeta. mis aitab vältida liigset kulusid [35]. Lisaks toetab täpne varude arvestus finantsarvestust varade õige hindamise abil, võimaldab optimeerida tellimisprotsesse ning tuvastada lahknevusi, mis võivad viidata kauba kahjustumisele, kadumisele või vargusele [36].

Traditsioonilised inventuurimeetodid, mis sageli tuginevad käsitsi loendamisele paberilehtede või lihtsate arvutustabelite abil, on märkimisväärselt keerukad [37]. Need meetodid on tavaliselt töömahukad ja aeganõudvad [38], sidudes ettevõtte ressursse ja takistades töötajatel keskendumast väärtust lisavamatele tegevustele. Inimlike vigade tekkimise oht on suur mitmes etapis: alates ebatäpsustest füüsilise loendamise käigus kuni vigadeni andmete ümberkirjutamisel paberilt digitaalsele vormingule või nende konsolideerimisel arvutustabelites [39]. Isegi väike laokadu võib tähendada ettevõttele märkimisväärt rahalist kahju aastas [38]. Lisaks põhjustavad need traditsioonilised lähenemisviisid sageli viivitusi füüsilise loendamise tulemuste ja tegelike, tegutsemiskõlblike andmete kättesaadavuse vahel, mis takistab õigeaegset otsuste langetamist [37] ja võib viia olukordadeni, kus müüakse kaupa, mida tegelikult laos ei ole [38]. Struktureeritud, tsentraliseeritud süsteemi puudumine muudab sageli ka auditeerimise keeruliseks ja pakub vaid piiratud ülevaadet kogu inventuuriprotsessist [40].

## 1.1 Probleemipüstitus

Käesoleva lõputöö kontekst käsitleb Eesti tootmisettevõtte (edaspidi ettevõtte) praegustes varude haldamise protseduurides täheldatud ebatõhusust. Olemasolevad protsessid, olgu need siis täielikult käsitsi teostatavad või sõltuvad omavahel mitteseotud arvutustabelisüsteemidest nagu Microsoft Excel (Excel), kannatavad sageli eelpoolmainitud puuduste all. Nendeks on märkimisväärne ajakulu ja suur andmesisestusvigade oht, nagu valesti loenda-

mine [39], ja viivitused lõpparuannete koostamisel. Need arvutustabelipõhised meetodid nõuavad sageli tülikat käsitsi andmetöötlust, näiteks veergude ümberjärjestamist, mis suurendab veelgi vigade ohtu ja raiskab väärtuslikku aega. Ebapiisav laohaldus viib ebatäpsete laojääkideni [35]. Lisaks puuduvad sageli selged kasutajate rollid. See tähendab, et tundlike toiminguid, nagu andmemuudatused, ei pruugita piisavalt kontrollida. Kaasaegsete tehnoloogiliste võimaluste, näiteks tootekoodide skannimise mobiilseadmete kaamerate abil, sidumine vanematesse töövoogudesse on sageli puudulik või puudub üldse. See jätab kasutamata olulise võimaluse suurendada nii lugemise kiirust kui ka täpsust. Järelikult on ilmne vajadus erilise, struktureeritud ja tõhusa digitaalse platvormi järele, mis võimaldaks hallata kogu inventuuri protsessi alates kaubaartiklite defineerimisest kuni tulemuste analüüsi ja aruandluseni.

## 1.2 Töö eesmärk

Käesoleva lõputöö peamine eesmärk on arendada veebipõhine rakendus, mis on kohandatud inventuuride läbiviimiseks, püüdes lahendada eelpool kirjeldatud traditsiooniliste meetoditega seotud piiranguid. Arenduses kasutatakse kaasaegset tehnoloogiavirna, mis koosneb Microsofti tarkvaraarendusplatvormil .NET 8 (edaspidi .NET) põhinevast tagaarakenduse API-st, Vue.js 3 (edaspidi Vue.js) põhinevast eesrakendusest ning PostgreSQL 17 (edaspidi PostgreSQL) andmebaasist andmete püsivaks salvestamiseks.

Arendatava rakenduse peamised funktsionaalsed eesmärgid on:

- Kasutajate haldamise rakendamine erinevate rollidega, jõustades rollipõhist juurdepääsukontrolli rakenduse erinevatele funktsioonidele.
- Põhiartiklite andmete importimise mehhanismi loomine Exceli failist, võimaldades loendatavate artiklite esialgset ja efektiivset seadistamist.
- Volitatud kasutajatel võimaluse loomine eraldi inventuurilehtede loomiseks ja haldamiseks, sidudes need tsoonidega.
- Määratud kasutajatel võimaluse loomine loenduskannete tõhusaks esitamiseks aktiivse loenduslehe artiklite kohta, toetades nii käsitsi andmete sisestamist kui ka integreeritud kaamerapõhise koodi lugemise abil lihtsustatud artiklite tuvastamist.
- Serveripoolsete põhiliste ärireeglite jõustamine, näiteks positiivsete loendatavate koguste nõudmine ja kasutajate tegutsemine vastavalt nende määratletud õigustele.

- Lõpetatud inventuuriandmete eksportimise hõlbustamine struktureeritud Exceli faili, mis sobib edasiseks analüüsiks või importimiseks teistesse ettevõtte ressurside planeerimise (ERP) süsteemidesse.

### 1.3 Töö ulatus

Käesolevas projektis töötatakse välja funktsionaalne rakendus, mis demonstreerib inventuuri põhilist töövoogu. See hõlmab järgmisi komponente ja tegevusi:

- Tagarakenduse arendus, mis sisaldab kasutajate autentimist, põhirollide haldamist, ning API lõpp-punkte artiklite, loenduslehtede ja loenduskannete haldamiseks.
- Eesrakenduse arendus, mis pakub kasutajaliideseid (UI) sisselogimiseks, artiklite nimekirja vaatamiseks, loenduslehtede valimiseks, käsitsi loenduskannete sisestamiseks, koodi lugemise algatamiseks ja artiklite otsimiseks ning esitatud kannete vaatamiseks.
- Loenduse esitamise ja autoriseerimise põhilise äriloojika rakendamine.
- Funktsionaalsuse loomine loendustulemuste eksportimiseks eelnevalt määratletud Exceli vormingusse.
- Rakenduse funktsionaalne testimine, et tagada põhifunktsioonide ettenähtud toimimine.
- Rakenduse koodibaasi ja peamise kasutusjuhendi dokumenteerimine.
- Konteineriseeritud rakenduse juurutamine pilveplatvormil, tagades sellele ligipääsu veebibrauseri kaudu.

Järgmised aspektid on käesoleva lõputöö raamidest teadlikult välja arvatud:

- Otsene andmebaasi tasemel integratsioon ERP-süsteemidega; suhtlus piirdub määratletud Exceli impordi- ja ekspordivormingutega.
- Keerukad inventuuri analüütika või reaajas laoseisu jälgimise funktsioonid rakenduses endas.
- Autonoomsed töövõimalused; eeldatakse võrguühenduse olemasolu rakenduse kasutamisel.
- Kasutajakontode integreerimine olemasolevate ettevõtte identiteedihaldussüsteemidega.

- Täiustatud turvameetmed nagu kaheastmeline autentimine või põhjalikud turvaauditid.
- Toote täielik lokaliseerimine mitmesse keelde peale inglise keele.

## **2 Metoodika**

Käesolev peatükk kirjeldab veebipõhise inventuurirakenduse arendamisel kasutatud metoodikat. Peatükk hõlmab projekti uurimisparadigmat, nõuete analüüsi, valitud tehnoloogiavirna, rakenduse arhitektuurilist disaini, arendusprotsessi ning juurutamise üksikasju.

### **2.1 Uurimisparadigma**

Käesolev lõputöö järgib disainiteaduse uurimismetoodika põhimõtteid, nagu [41] kirjeldab. Disainiteadus keskendub innovaatiliste infotehnoloogiliste artefaktide (IT-artefaktide), antud juhul inventuurirakenduse, loomisele, et lahendada praktilisi organisatsioonilisi probleeme ning laiendada valdkonna teadmistebaasi. Lõputöö panus seisneb nii loodud artefaktis endas, selle praktilises kasulikkuses, kui ka selle artefakti disaini, ehitamise ja esmase valideerimise käigus saadud teadmistes. Käesolev töö dokumenteerib sellise artefakti loomise protsessi ja hindab selle vastavust püstitatud eesmärkidele ning esialgset võimalikku kasulikkust. Loodud artefakti hindamisel arvestatakse ka [42] raamistiku aspekte, keskendudes artefakti funktsionaalsuse ja kasutatavuse valideerimisele prototüübi staadiumis.

### **2.2 Nõuete analüüs**

Projekti aluseks oli põhjalik analüüs, mis hõlmas konsultatiivset koostööd antud ettevõttega, selgitamaks välja nende spetsiifilistest vajadustest tulenevad tõhusa inventuurirakenduse funktsionaalsed ja kvaliteedinäitajad. Eesmärgiks oli ületada käsitsi teostatavate protsesside piirangud ning rakendada selleks kaasaegseid veebitehnoloogiaid.

#### **2.2.1 Funktsionaalsed nõuded**

Projekti eesmärkidest ja inventuuri tüüpilisest töövoost lähtuvalt tuvastati järgmised põhilised funktsionaalsed nõuded rakendusele:



- Rakendus peab tagama kasutajate turvalise sisselogimise ning juurdepääs rakenduse erinevatele funktsioonidele peab olema piiratud vastavalt kasutajale määratud rollile.
- Administraatoritel peab olema võimalus hallata rakenduse kasutajaid, luua uusi kontosid, ning omada ligipääsu rakenduse tundlikele seadetele. Administraatori roll hõlmab kõiki halduri õigusi.
- Halduritel peab olema võimalus importida kaubaartiklite põhiaandmeid Exceli failist, määratleda inventuuritsoone, luua ja hallata inventuurilehti ning eksportida inventuuritulemusi Exceli faili.
- Lugejatel peab olema võimalus vaadata kättesaadavaid inventuurilehti, sisestada inventuurikandeid konkreetse lehe kohta ning vaadata enda poolt sisestatud kandeid.
- Rakendus peab võimaldama volitatud kasutajatel importida kaubaartiklite nimekirja, kasutades spetsiaalselt vormindatud ERP (ettevõtte ressursside planeerimise) süsteemist eksporditud Exceli tabelleid.
- Rakenduses peab olema funktsionaalsus inventuuride läbiviimiseks eraldi määratletud tsoonides.
- Volitatud kasutajatel peab olema võimalik algselt luua uus inventuur, määrates sellele nimetuse ja sidudes selle vajadusel konkreetse tsooniga.
- Lugejatel peab olema kasutajaliides (UI), kus nad saavad valida artikli konkreetse inventuurilehe põhinimekirjast ja sisestada füüsiliselt loendatud koguse, millele rakendatakse sisendandmete valideerimist.
- Rakendus peab pakkuma mehhanismi, mille abil lugejad saavad kasutada seadme kaamerat artiklikoodide lugemise, et kiirendada artiklite tuvastamist loenduskannete sisestamise faasis.
- Lugejad peavad nägema kinnitust enda tehtud loenduskannetest; halduritel peab olema laiem ülevaade kõikidest sisestatud kannetest konkreetse inventuurilehe raames.
- Volitatud kasutajatel peab olema võimalik eksportida kogutud loendusandmed täiendatud inventuurilehelt struktureeritud ja eelnevalt määratletud Exceli vormingusse.
- Rakendus peab olema disainitud modulaarselt, et tulevikus oleks võimalik selle integreerimine ERP-süsteemi moodulina.

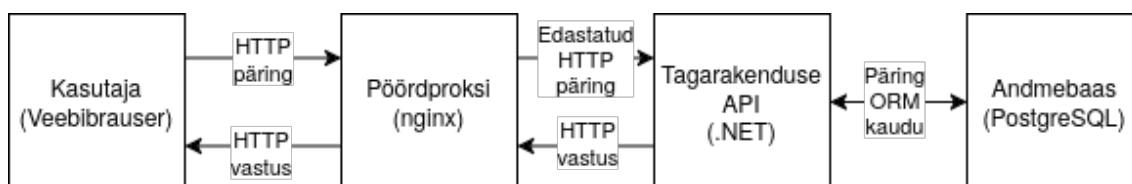
### **2.2.2 Mittefunktsionaalsed nõuded**

Lisaks põhifunktsioonidele seati rakendusele järgmised kvaliteedinõuded:

- UI peab olema põhitegevuste täitmiseks loogiline ja lihtsasti õpitav. Erilist tähelepanu tuleb pöörata loendusliidese kasutatavusele erinevates seadmetes, tagades selle reageerimisvõime.
- Kasutajate paroolid peavad olema andmebaasis turvaliselt räsituna salvestatud [43].
- Rollipõhine autoriseerimine peab olema järjepidevalt jõustatud tagarakenduse API lõpp-punktides.
- Rakenduse põhilised töövood peavad toimima stabiilselt ja usaldusväärselt. Rakenda-da tuleb veakäsitlus ja kasutajate informeerimine sobimatute sisendite või süsteemi-probleemide korral.
- Koodibaas peab järgima valitud tehnoloogiate standardseid praktikaid. Kogu aren-dusprotsessi vältel tuleb kasutada versioonihaldussüsteemi.
- Rakendus peab tunduma reageeriv tüüpiliste interaktsioonide ajal ning suutma taluda mitme samaaegse kasutaja tegevust ilma olulise jõudluse halvenemiseta.
- Rakendusega peab kaasas olema selge ja arusaadav kasutusjuhend lõppkasutajatele, mis katab peamised töövood.

## 2.3 Rakenduse arhitektuur ja tehnoloogiate valik

Rakenduse arhitektuuriks valiti kolmekihiline klient-server mudel, mis on levinud N-tasemelise arhitektuuri realisatsioon [44], [45]. Selline arhitektuurne valik toetab komponentide sõltumatut arendust ja testimist, võimaldades eri kihtide kallal samaaegset tööd [46], [47], parandab skaleeritavust, kuna iga kihti saab vajadusel eraldi skaleeri-da [48], [44], ning pakub paindlikumat tehnoloogiate valikut igale kihile. Tehnoloogiate valiku aluseks olid projekti nõuete täitmine, ettevõtte arendajate varasem kogemus ja avatud lähtekoodiga tarkvara kättesaadavus. Joonis 1, mis illustreerib rakenduse üldist arhitektuuri, kihtidevahelisi seoseid ning komponentide paiknemist Dockeri konteinerites, on visualiseeritud kasutades draw.io [49] veebirakendust.



Joonis 1. Rakenduse arhitektuur ja peamised komponendid Dockeri keskkonnas.

### 2.3.1 Tagarakendus

Tagarakendus, mis vastutab äriloogika, andmete valideerimise ja andmebaasisuhtluse eest, realiseeriti .NET-il, kasutades C# programmeerimiskeelt. See valik tehti tänu .NET platvormi turvalisuse, usaldusväärsuse ja kõrge jõudluse omadustele API-de implementeerimisel [1], ASP.NET Core raamistiku kõrgjõudluslikule arhitektuurile [4], Entity Framework Core-i (EF Core) kergekaalu ja laiendatavale disainile andmebaasioperatsioonide haldamisel [11], ettevõtte arendajate eelnevatele kogemusele antud tehnoloogiatega ning .NET 8 LTS-toele (*Long-Term Support*), mis tagab lahenduse stabiilsuse. Tagarakendus implementeeriti REST-põhise (*Representational State Transfer*) API-na, mille andmevahetusformaadiks on JSON (JavaScript *Object Notation*). API lepingute defineerimiseks ja interaktiivseks testimiseks rakendati OpenAPI spetsifikatsiooni, kasutades Swagger tööriistu.

Kasutajate autentimiseks ja API päringute autoriseerimiseks kasutati ASP.NET Core Identity teeki. Kaubaartiklite nimekirjade importimiseks Exceli failidest ja inventuuritulemuste eksportimiseks samasse formaati integreeriti ClosedXML [50] teeki. Tagarakenduse korrektse toimimise tagamiseks implementeeriti põhjalikud ühiktestid. Ühiktestid on tarkvaraarenduses kriitilise tähtsusega, kuna need aitavad varakult avastada vigu koodi väikseimates osades, lihtsustavad refaktoormist ja tagavad koodi stabiilsuse pikaajaliselt. Testimiseks kasutati levinud .NET testimisraamistikku xUnit [51]. Sõltuvuste isoleerimiseks ja testitavate ühikute eraldamiseks kasutati Moq [52] teeki.

### 2.3.2 Eesrakendus

Eesrakendus, mis tegeleb UI kuvamise ja kasutajate interaktsioonide haldamisega, arendati kasutades Vue.js raamistikku ja TS (TypeScript) programmeerimiskeelt. Vue.js valiti tänu selle komponendipõhisele programmeerimismudelile, mis aitab efektiivselt arendada mistahes keerukusega kasutajaliideseid, ning automaatsele reaktiivsussüsteemile, mis jälgib oleku muutusi [53]. TS-i kaasamine tagas staatilise tüübikontrolli, mis aitab tuvastada tüüбивigu kompileerimise ajal, parandades sellega koodi kvaliteeti ja vähendades jooksmis-aegsete vigade teket [54]. Arendustsükli kiirendamiseks kasutati Vite ehitustööriista, mis pakub kiiret kuuma mooduli asendust arendusrežiimis ning optimeeritud staatilist varade pakkimist tootmiskeskkonnas [55].

Eesrakendus suhtleb tagarakendusega asünkroonsete HTTP (*Hypertext Transfer Protocol*) päringute kaudu, kasutades selleks Axios teeki. Rakenduse globaalset olekut, sealhulgas sisselogitud kasutaja andmeid ja aktiivset inventuurilehte, hallati Pinia teegi abil. Artiklikoodide lugemiseks seadme kaamera abil integreeriti html5-qrcode [56] JS teek. Eesrakenduse kvaliteedi tagamiseks implementeeriti automatiseeritud testid nii peamistele kasutajavoo- gudele kui ka individuaalsetele komponentidele, kasutades Vitest [57] testimisraamistikku, mis on optimeeritud Vite ehitussüsteemi ja Vue.js rakenduste jaoks. Vue.js komponentide isoleeritud testimine võimaldab kontrollida nende korrektset renderdamist, atribuutide käitlemist ja sündmuste väljastamist, tagades rakenduse UI stabiilsuse nii iseseisvate kom- ponentide kui ka nende koostoime tasandil.

### **2.3.3 Andmebaas**

Andmete püsivaks salvestamiseks valiti PostgreSQL. Otsus põhines selle avatud lähtekoo- digi objekt-relatsioonilise andmebaasisüsteemi tuntud töökindlusel, tehingute atomaarsuse, konsistentsuse, isolatsiooni ja vastupidavuse toel, heal jõudlusel mitme samaaegse kasu- taja korral ning võimel hallata keerukaid andmestruktuure, mis teeb sellest sobiva valiku projekti andmete jaoks [58].

Tagarakendus suhtleb andmebaasiga, nagu eelnevalt mainitud, EF Core vahendusel.

### **2.3.4 Pidevintegratsioon ja pidevvalmidus**

Projekti arendustsükli automatiseerimiseks rakendati CI/CD praktikaid, kasutades GitHub Actions platvormi. See lähenemine võimaldab tarkvaraarenduse elutsükli automatiseeritud haldamist, alates koodimuudatuste integreerimisest kuni testimise ja juurutamiseni [59], mille tulemusena valmivad rakenduse uuendused kiiremini ja usaldusväärsemalt [60].

Iga koodimuudatuse korral peamises repositooriumi harus käivitab GitHub Actions au- tomaatse töövoo, mis sisaldab rakenduse komponentide ehitamist ning automaatsete käivitamist.

### **2.3.5 Konteineriseerimine ja juurutamine**

Rakenduse kõik komponendid, eesrakendus koos nginx veebiserveriga, tagarakenduse API ja PostgreSQL andmebaas, pakendati ja juurutati kasutades Dockeri konteineriseeri-

mist. Konteineriseerimine võimaldab rakenduse ja selle sõltuvuste isoleerimist, tagades ühtse käituskeskkonna erinevates süsteemides ning lihtsustades juurutamist ja skaleerimist [61], [62].

Eesrakenduse staatiliste failide serveerimiseks ning päringute suunamiseks tagarakendusele Dockeri konteineris rakendati nginx pöördproksi. Mitme konteineri kooskõlastatud haldamiseks ja käivitamiseks, eriti arenduskeskkonnas, kasutati Docker Compose-i. Kogu projekti lähtekood ja arendusega seotud dokumentatsioon hallatakse GitHubi repositooriumis.

## 2.4 Alternatiivsete lahenduste analüüs

Nõute analüüsi oluline osa oli olemasolevate inventuurihaldustarkvarade hindamine, et tagada valitud kohandatud arenduse lähenemise otstarbekus projekti eesmärkide saavutamiseks. Analüüsiti mitmeid avatud lähtekoodiga lahendusi ning ühte SaaS (*Software as a Service*) lahendust. Tabelis 1 esitatakse valitud lahenduste võrdlev analüüs.

Tabel 1. Alternatiivsete lahenduste võrdus.

| Lahendus                        | Litsents   | Tehnoloogia                         | Fookusala                                               | Piirangud kontekstis                                                                                          |
|---------------------------------|------------|-------------------------------------|---------------------------------------------------------|---------------------------------------------------------------------------------------------------------------|
| Stock and Inventory Online [63] | SaaS       | Ei ole teada                        | Üldine laovarude haldamine                              | Avatud lähtekoodi paindlikkuse puudumine, SaaS-mudeliga seotud potentsiaalsed kulud, kasutajate arvu piirang. |
| Ralph NG [64]                   | Apache-2.0 | Python (Django), MySQL/- PostgreSQL | IT-varade haldamine                                     | Fookus IT-spetsiifilistele varadele, Dockeri implementatsioon katsejärgus [65].                               |
| GLPI Project [66]               | GPLv3+     | PHP, MySQL/- MariaDB                | IT-varade ja -teenuste haldus                           | Tugev orienteeritus IT-toe protsessidele.                                                                     |
| OpenBoxes [67]                  | EPL-1.0    | Java (Grails), MySQL/- MariaDB      | Tarneaahela haldamine (tervishoid, humanitaarlogistika) | Laiem logistika fookus.                                                                                       |

Jätkub...

Tabel 1 – Jätkub...

| Lahendus       | Litsents | Tehnoloogia                   | Fookusala                                         | Piirangud kontekstis                              |
|----------------|----------|-------------------------------|---------------------------------------------------|---------------------------------------------------|
| Snipe-IT [68]  | AGPLv3   | PHP (Laravel), MySQL/-MariaDB | IT-varade (riistvara, litsentsid) haldamine       | Ei ole mõeldud mitte-IT-objektide inventuurideks. |
| InvenTree [69] | MIT      | Python (Django)               | Elektroonika ja tootmisdetailide varude haldamine | Spetsiifiline fookus.                             |

Analüüsitud lahendustel ilmnes ühine joon: nad kõik vajaksid vähemalt teatud määral kohandamist, et vastata täielikult käesoleva projekti spetsiifilistele vajadustele, nagu näiteks andmete import ja eksport kindlas Exceli formaadis ning projektis kasutatavate koodide tugi. Pealegi on enamik neist loodud keerukamate varude haldamise stsenaariumide jaoks, samas kui antud projekt keskendub lihtsama ja sihipärasema lahenduse väljatöötamisele, mis võiks tulevikus potentsiaalselt integreeruda olemasolevasse ERP-süsteemi.

Seetõttu otsustati kohandatud tarkvaraarenduse kasuks, valides tehnoloogiavirnaks .NET koos C#-ga tagarakenduse jaoks, Vue.js koos TS-iga eesrakenduseks ning PostgreSQL andmebaasiks. See lähenemisviis võimaldab luua lahenduse, mis on täpselt kohandatud defineeritud nõuetele, hõlmates spetsiifilist inventuuri töövoogu, integreeritud koodilugemist ja rollipõhist juurdepääsukontrolli, kasutades ettevõtte arendajatele tuttavat tehnoloogiavirna.

#### 2.4.1 Arendusmetoodika

Käesoleva projekti arendusprotsess järgis iteratiivset lähenemist, mis on inspireeritud agiilsetest tarkvaraarenduse põhimõtetest. Arvestades projekti ulatust ja seda, et tegemist oli peamiselt individuaalse tööga, ei rakendatud ranget Scrumi või muud formaalset agiilset raamistikku täies mahus. Selle asemel keskenduti järgmistele printsiipidele:

- Töö jaotati loogilisteks etappideks ja funktsionaalsuseplokkideks, mis vastasid eelnevalt koostatud arendusplaanis kirjeldatud faasidele. Iga iteratsiooni eesmärk oli realiseerida konkreetne osa rakendusest, alustades tuumikfunktsionaalsusest ja lisades järk-järgult täiendusi.

- Kuigi formaalseid sprinditsükleid ei kasutatud, toimus regulaarne progressi hindamine ja vajadusel plaanide kohandamine, põhinedes tagasisidel antud tootmisettevõtte esindajalt, kes pakkus praktilist sisendit rakendusele.
- Nagu eelnevalt mainitud, rakendati API-keskset lähenemist, kus tagarakenduse API lepingud defineeriti varakult. See võimaldas ees- ja tagarakenduse arendusel toimuda teatud määral paralleelselt ja vähendas integratsiooniprobleeme.
- Koodi versioonide haldamiseks kasutati järjepidevalt Giti GitHub-il. Uute funktsioonide ja paranduste arendamiseks kasutati harustrateegiaid.
- Arendustegevused prioritseeriti vastavalt projekti eesmärkidele, keskendudes esmalt kriitiliste funktsionaalsuste valmimisele.

## 2.5 Valideerimine

Lisaks tehnilisele testimisele viidi läbi ka loodud rakenduse esialgne hindamine, kaasates potentsiaalseid lõppkasutajaid tagasiside kogumiseks. Tagasisidet koguti lugeja rollis mitte-tehnilistelt testijatelt, kellel oli üldine arvutikogemus, kuid puudusid sügavad tehnilised teadmised. Lisaks saadi tagasisidet halduri rollis ettevõtte esindajalt, kellel oli ülevaade reaalsest inventuuri protsessidest ja ärinõuetest.

Testkasutajatel paluti rakenduses täita eelnevalt määratletud ülesandeid. Peamised ülesanded hõlmasid sisselogimist, määratud laoseisulehe valimist, kolme artikli koguse käsitsi sisestamist ja kolme artikli koguse sisestamist koodilugemise abil. Pärast ülesannete täitmist viidi läbi lühike poolstruktureeritud intervjuu, et saada üldmuljeid kasutajaliidese intuiitiivsuse, õpitavuse ja kasutusmugavuse kohta.

Ettevõtte esindajale näidati rakenduse kõiki peamisi funktsionaalsusi, sealhulgas halduri ja lugeja rollide töövooge. Demonstratsioonile järgnes põhjalikum poolstruktureeritud intervjuu, mis keskendus rakenduse funktsionaalsuse vastavusele tegelikele ärivajadustele ja üldisele kasutajakogemusele administraatori vaatenurgast.

## **3 Tulemused**

Käesolev peatükk esitab peatükis 2 kirjeldatud arendusprotsessi tulemusena valminud lahenduse. Peatükis kirjeldatakse loodud veebipõhise inventuurirakenduse UI-st ja põhi-funktsionaalsusi ning esitatakse rakenduse valideerimiseks läbi viidud testimise tulemused.

### **3.1 Loodud rakenduse ülevaade**

Arendustöö tulemusena valmis inventuuri läbiviimise veebirakendus. Rakendus koosneb Vue.js põhinevast UI-st, .NET põhinevast REST API-st ning andmete hoiustamiseks kasutatavast PostgreSQL andmebaasist. Rakendus realiseerib projekti põhieesmärgid, pakkudes rollipõhist ligipääsu ning tuge inventuuriprotsessi põhietappidele, alates kaubaandmete impordist kuni tulemuste ekspordini. Järgnevates alapeatükkides tutvustatakse rakenduse UI-st ja funktsionaalsusi rollide kaupa ning kirjeldatakse teostatud testimise tulemusi.

### **3.2 Kasutajaliides ja funktsionaalsus**

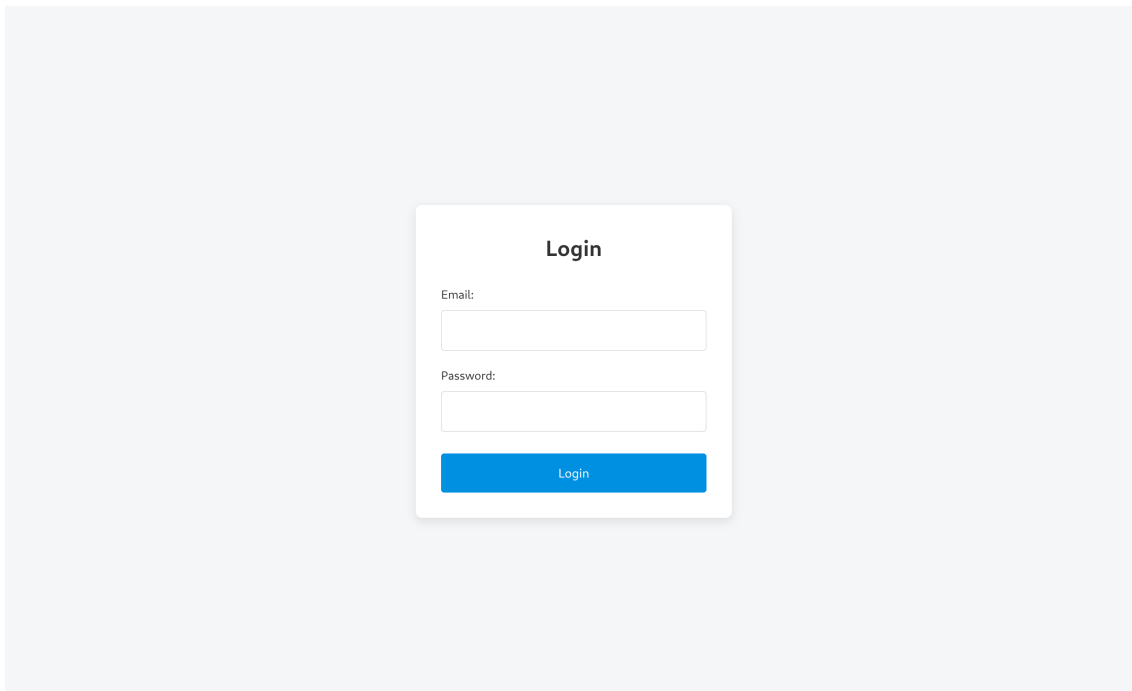
Rakenduse UI on loodud eesmärgiga olla võimalikult intuitiivne vastavalt kasutaja rollile määratud ülesannetele. Järgnevalt kirjeldatakse peamisi kasutajavoogusid ja liidese elemente.

#### **3.2.1 Autentimine**

Kõik rakenduse kasutajad peavad end autentima läbi sisselogimisliidese, mis on kujutatud Joonisel 2.

Kasutaja sisestab oma e-posti aadressi ja parooli. Pärast andmete edastamist valideerib tagarakendus need ning eduka autentimise korral suunatakse kasutaja vastavalt tema rollile rakenduse põhilehele. Turvalisuse tagamiseks kasutatakse tokeni põhiseid autentimis- ja autoriseerimismehhanisme, kusjuures autentimistoken salvestatakse kliendipoolset turvalises küpsises.



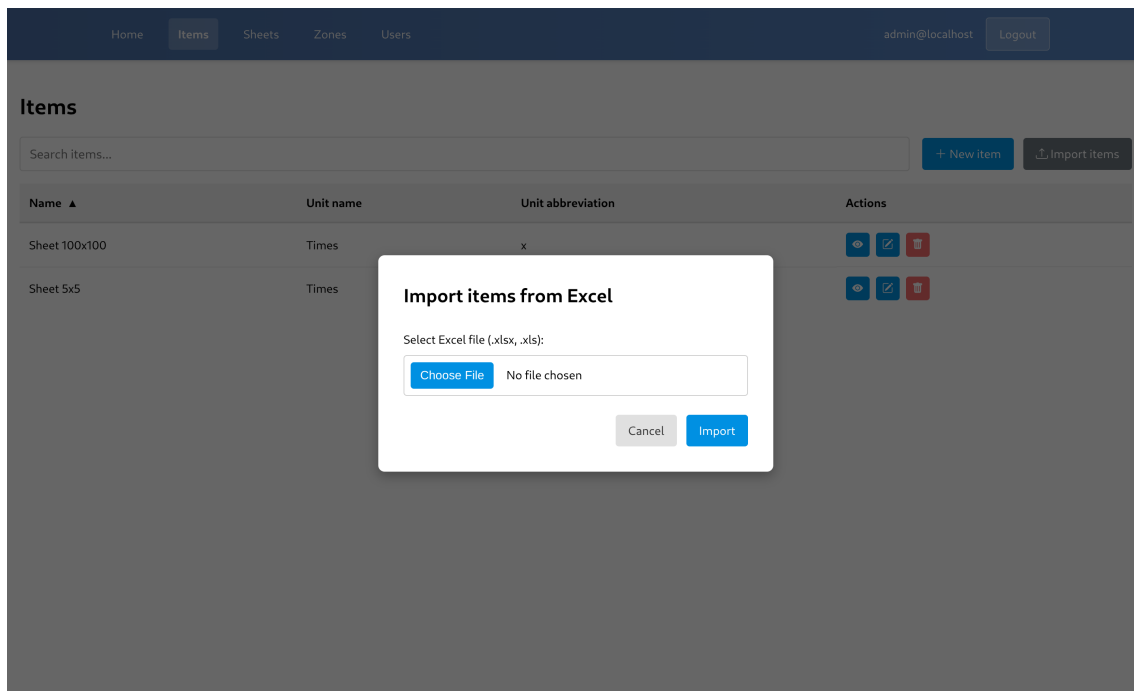


Joonis 2. Sisselogimisliides.

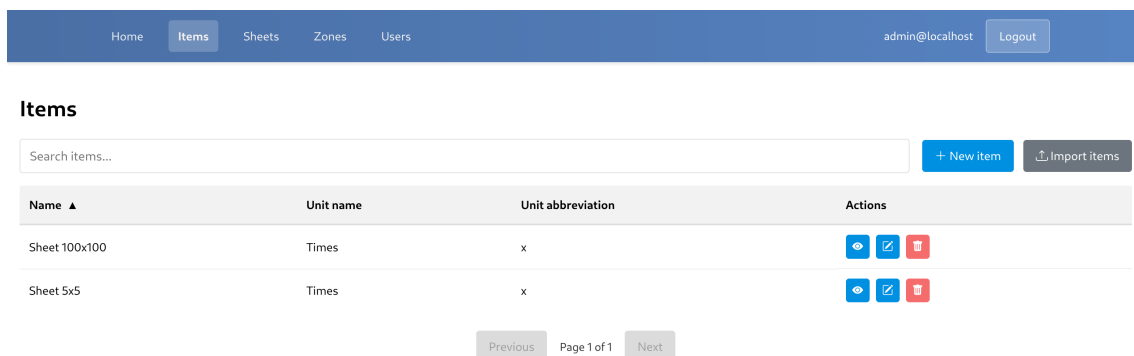
### 3.2.2 Artikliandmete import

Pärast sisselogimist navigeerib administraator või haldur haldusliideses kaubaartiklite impordi sektsiooni. Kasutades failivaliku liidest (vt Joonis 3), valib kasutaja eelnevalt ettevalmistatud, ERP-süsteemist eksporditud Excel-faili kaubaartiklite nimekirjaga.

Faili üleslaadimisel töötleb tagarakendus selle sisu, lisades või uuendades artikleid rakenduse andmebaasis. UI kuvab seejärel teate impordi õnnestumise või ebaõnnestumise kohta. Eduka impordi tulemusena on imporditud artiklid nähtavad rakenduse artiklite nimekirja vaates. (vt Joonis 4).



Joonis 3. Artikliandmete importimise UI.



Joonis 4. Artiklite nimekiri.

### 3.2.3 Inventuurilehe loomine ja haldamine

Administraator või haldur navigeerib inventuurilehtede haldamise vaatesse (vt Joonis 5).

Uue inventuuri alustamiseks valib kasutaja „*New stocktaking*“ nupu, mis avab inventuu-

rilehe loomise vormi (vt Joonis 6). Vormil tuleb täita kohustuslikud väljad, nagu inventuurilehe nimi ja valida inventuuri ala. Pärast vormi esitamist luuakse rakenduses uus inventuurileht, mis ilmub hallatavate inventuurilehtede nimekirja.

Home

Items

Sheets

Zones

Users

admin@localhost

Logout

# Stocktakings

Search stocktakings...

+ New stocktaking

| Name ▲   | Zone | Actions                                      |
|----------|------|----------------------------------------------|
| W1Z1-May | W1Z1 | <div><div></div><div></div><div></div></div> |
| W2Z4-May | W2Z4 | <div><div></div><div></div><div></div></div> |

Previous

Page 1 of 1

Next

Joonis 5. Inventuurilehte haldamise vaade.

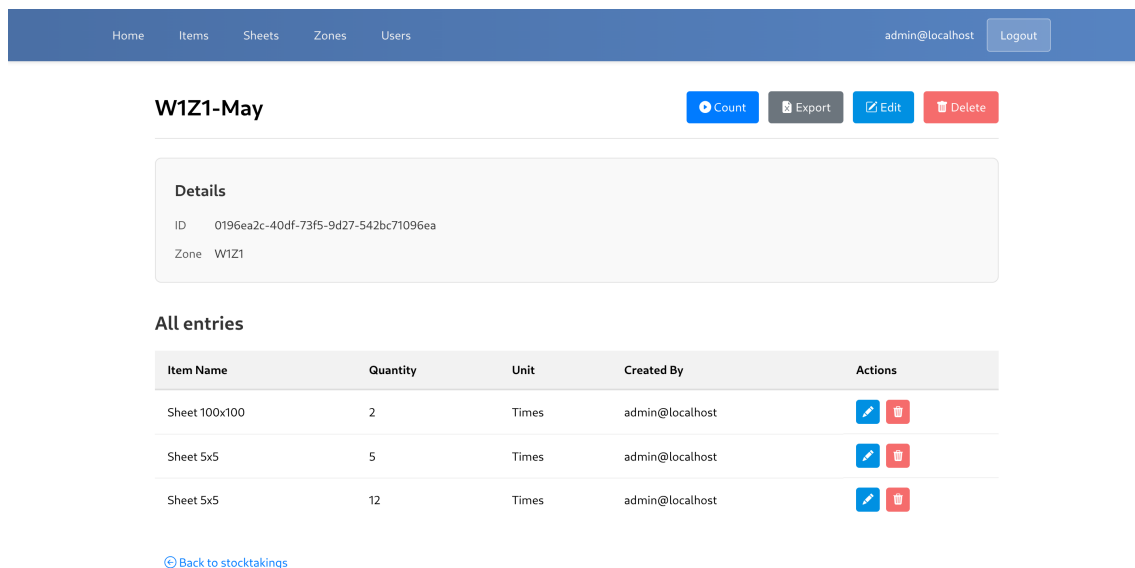
|                                      |  |  |                        |  |  |  |
|--------------------------------------|--|--|------------------------|--|--|--|
| Home Items <b>Sheets</b> Zones Users |  |  | admin@localhost Logout |  |  |  |
| <b>New stocktaking</b>               |  |  |                        |  |  |  |
| Name:                                |  |  |                        |  |  |  |
| <input type="text"/>                 |  |  |                        |  |  |  |
| Zone:                                |  |  |                        |  |  |  |
| <div>W1Z1</div>                      |  |  |                        |  |  |  |
| <div>Cancel</div> <div>Create</div>  |  |  |                        |  |  |  |

Joonis 6. Uue inventuurilehe loomise vorm.

### 3.2.4 Inventuuriandmete eksport

Kui inventuuri loendus on lõpetatud ja vastav inventuurileht on valitud haldusliidesest, saab administraator või haldur algatada andmete ekspordi.

Vajutades „Export“ nupule (selle asukoht on nähtav Joonisel 7), genereerib tagarakendus nõutud formaadis Excel-faili. See fail sisaldab kõiki valitud inventuurilehe raames sisestatud kirjeid eelnevalt määratletud struktuuris. Pärast genereerimist algatab kasutaja veebibrauser automaatselt faili allalaadimise.



The screenshot shows a web application interface for inventory management. At the top, there is a navigation bar with links: Home, Items, Sheets, Zones, Users. On the right, it shows the user 'admin@localhost' and a 'Logout' button. Below the navigation bar, the main content area is titled 'WIZ1-May'. To the right of the title are four buttons: 'Count' (blue), 'Export' (grey), 'Edit' (blue), and 'Delete' (red). Below the title bar, there is a 'Details' section with the following information: ID: 0196ea2c-40df-73f5-9d27-542bc71096ea, Zone: WIZ1. Below the details section, there is a table titled 'All entries'. The table has five columns: Item Name, Quantity, Unit, Created By, and Actions. The table contains three rows of data. At the bottom of the table, there is a link: '© Back to stocktakings'.

| Item Name     | Quantity | Unit  | Created By      | Actions                                     |
|---------------|----------|-------|-----------------|---------------------------------------------|
| Sheet 100x100 | 2        | Times | admin@localhost | <a href="#">Edit</a> <a href="#">Delete</a> |
| Sheet 5x5     | 5        | Times | admin@localhost | <a href="#">Edit</a> <a href="#">Delete</a> |
| Sheet 5x5     | 12       | Times | admin@localhost | <a href="#">Edit</a> <a href="#">Delete</a> |

Joonis 7. Inventuurilehe andmete vaade.

### 3.2.5 Inventuurilehe valimine

Pärast edukat sisselogimist kuvatakse kasutajale nimekiri aktiivsetest inventuurilehtedest, millele tal on ligipääs (vt Joonis 8).

Lugeja valib sellest nimekirjast inventuurilehe, millega ta soovib tööd alustada.

Home

Items

Sheets

user@localhost

Logout

Stocktakings

Search stocktakings...

| Name ▲   | Zone | Actions |
|----------|------|---------|
| W1Z1-May | W1Z1 |         |
| W2Z4-May | W2Z4 |         |

Previous

Page 1 of 1

Next

Joonis 8. Inventuurilehe valiku vaade lugejale.

### 3.2.6 Käsitsi koguse sisestamine

Pärast inventuurilehe valimist avaneb kasutajale peamine loendusliides (vt Joonis 9).

Loendatava kaubaartikli valimiseks kasutab ta otsingufunktsiooniga rippmenüüd või auto-maattäitega välja. Seejärel sisestab ta vastavale numbriväljale füüsiliselt loendatud koguse. Rakendus teostab sisestatud kogusele kliendipoolse valideerimise. Vajutades „Add entry“ nupule, saadetakse andmed tagarakendusse salvestamiseks. Eduka salvestamise korral kuvatakse kasutajale kinnitusteadet ning äsja sisestatud kirje ilmub lehe alaosas olevas „Your entries“ nimekirjas, pakkudes kohest tagasisidet (nähtav Joonisel 9).

## Counting: W1Z1-May

Zone: W1Z1

### Add entry

Scan

Search for item:

Quantity:

+ Add entry

### Your entries

No entries yet.

Joonis 9. Inventuurikirje käsitsi sisestamise liides, näidates artikli valikut ja koguse sisestust.

### 3.2.7 Skaneerimisega koguse sisestamine

Loendusliideses (vt Joonis 9) on kasutajal võimalus vajutada „Scan“ nupule. Loendusliideses on kasutajal võimalus vajutada „Scan“ nupule.

Esimesel kasutuskorral küsib veebibrauser luba seadme kaamera kasutamiseks. Pärast loa andmist avab veebibrauser seadme kaamera (vt Joonis 10). Näidatud kaamera suunatakse kaubaartikli koodile, mille kasutaja soovib skaneerida. JS teek dekodeerib koodi. Eduka

dekodeerimise järel täidetakse automaatselt kaubaartikli valikuväli vastava tootega. Seejärel sisestab kasutaja käsitsi loendatud koguse ja vajutab „Add entry“ nuppu. Edasine töövoog jätkub sarnaselt käsitsi sisestamisega, kus kirje salvestatakse ja kuvatakse esitatud kirjete nimekirjas.

user@localhostLogout

## Counting: W1Z1-May

Zone: W1Z1

### Add entry

Stop scanning



Search for item:  
Type to search for items...

Quantity:  
1

Joonis 10. Loendusliides, kus on valitud artikli skaneerimise nupp.

### **3.3 Testimine ja valideerimine**

Valminud rakenduse testimine keskendus põhiliste funktsionaalsuste ja kasutajavoogude valideerimisele, et tagada rakenduse vastavus püstitatud eesmärkidele. Testimine viidi läbi mitmetasandiliselt: manuaalsete läbiv testimise stsenaariumite abil, simuleerides iga kasutaja rolli tüüpilisi tegevusi rakenduses; tagarakenduse ühiktestide kaudu, mis valideerisid äri loogika komponentide õiget toimimist; ning eesrakenduse automatiseeritud UI testidega, mis kontrollisid UI elementide funktsionaalsust ja reageerimist kasutaja sisenditele. Lisaks kontrolliti andmete korrektsust andmebaasis ja eksporditud failides protsessi kriitilistes punktides.

Rakenduse kasutatavuse esialgseks hindamiseks viidi läbi ülesandepõhine testimine nelja mittetehnilise testkasutajaga, kes simuleerisid lugeja rolli. Testkasutajatel paluti sooritada lihtsaid loendamise ülesandeid, sealhulgas esemete käsitsi sisestamist ja koodide skannimist. Kõik testkasutajad suutsid neile määratud ülesanded edukalt täita. Intervjuude käigus kogutud tagasiside näitas, et kasutajaliides on üldiselt intuitiivne ja hõlpsasti kasutatav. Kasutajad märkisid, et eseme valimine ja koguse sisestamine on loogiline ning skannimisfunktsioon töötab hästi. Kasutajaliidese elementide paigutus ja navigeerimine olid selged ning kasutajad leidsid vajalikud funktsioonid kiiresti üles.

Rakendust demonstreeriti ka ettevõtte esindajale, et saada tagasisidet funktsionaalsuse vastavuse kohta tegelikele äri vajadustele administraatori vaatenurgast ja üldise äri protsessi seisukohast. Hinnang oli positiivne ja ettevõtte esindaja leidis, et rakendus katab suure osa nende vajadustest.

Ettevõtte esindaja oli rahul rakenduse funktsionaalsusega ja sellega, kuidas see toetab inventuuri protsessi. Impordi- ja ekspordifunktsioonid toimisid ootuspäraselt ning rakenduse kasutajaliides oli selge ja kasutajasõbralik. Kõik vajalikud funktsioonid olid rakenduses olemas ja töötasid korrektselt. Arutati ka võimalikke tulevasi täiustusi, kuid need jäävad käesoleva lõputöö ulatusest välja.

#### **3.3.1 Funktsionaalse testimise tulemused**

Kõik eelnevalt alapeatükis 3.2 kirjeldatud põhilised kasutajavood testiti edukalt nii eesrakenduse tasemel kui ka tagarakenduse osas, viimaste puhul täiendavalt valideerides nende



korrektset toimimist ka ühikutestidega. Nii manuaalsed läbivtestid testid, tagarakenduse ühiktestid kui ka eesrakenduse automatiseeritud kasutajaliidese testid kinnitasid rakenduse põhiliste funktsionaalsuste ootuspärast toimimist.

Administraatorid ja haldurid said edukalt importida kaubaartikleid, luua ja hallata inventuurilehti ning eksportida tulemusi. Lugejad said samuti edukalt sisse logida, valida neile määratud inventuurilehti, sisestada arvestuskirjeid nii käsitsi kui ka skaneerimise abil ning näha enda poolt sisestatud kirjeid.

### **3.3.2 Kasutatavuse ja kasutajaliidese vaatlused**

Manuaalse testimise käigus tehtud vaatlused näitasid, et põhiülesannete täitmiseks mõeldud liidesed on üldjoontes loogilise ülesehitusega ja arusaadavad. Rollipõhine funktsionaalsuse eristus oli selge. Loendusliidese funktsionaalsust testiti erinevatel füüsilistel seadmetel, sealhulgas tahvelarvutitel ja nutitelefonidel, ning leiti, et see toimib ootuspäraselt. Kasutajakogemus, kaasa arvatud väiksematel ekraanidel, on hea ja elementide paigutus on sobiv. Navigeerimine peamiste vaadete vahel oli üldiselt sujuv ja kiire.

### **3.3.3 Koodilugemise jõudluse tulemused**

Rakenduse lugemisfunktsiooni testiti kasutades erinevatel mobiilsetel seadmetel kaamerat. Testimisel kasutati koodi, mis olid prinditud tavalisele paberile.

Heades valgustingimustes tuvastas teek koodid usaldusväärselt ja täitis kaubaartikli välja korrektset.

### **3.3.4 Andmete valideerimise tulemused**

Andmete terviklikkuse kontrollimisel veenduti, et imporditud Exceli faili kaubaartiklite andmed kajastusid korrektset rakenduse andmebaasis. UI kaudu sisestatud kogused, nii käsitsi kui skaneerides, salvestati täpselt vastavatesse andmebaasi tabelitesse koos korrektsete viidetega artiklile ja inventuurilehele.

Lõplik eksporditud Excel-fail sisaldas kõiki vastava inventuurilehe kirjeid ning faili struktuur ja veergude järjestus vastasid algselt määratletud nõuetele. Põhiliste töövoogude testimisel ei tuvastatud olulisi andmete kadumise või moonutamise probleeme.

## 4 Analüüs ja arutelu

Käesolev peatükk esitab põhjaliku analüüsi ja arutelu peatükis 3 tutvustatud tulemuste kohta, mis käsitlevad väljatöötatud veebipõhist inventuurirakendust. Peatükis hinnatakse kriitiliselt projekti eesmärkide saavutamist, tõlgendades valideerimistulemusi ja rakendatud funktsioone. Analüüs liigub kaugemale lihtsatest funktsionaalsuskontrollidest, hinnates loodud infotehnoloogilise artefakti praktilist kasulikkust ja sellega kaasnevaid piiranguid. Lisaks toob peatükk esile valitud arhitektuurist ja tehnoloogiatest tulenevad olulised tugevused ning implementeeritud lahenduse piirangud, asetades need traditsiooniliste inventuurimeetodite ja võimalike parenduste konteksti. Samuti sisaldab peatükk üksikasjalikke mõtisklusi arendusprotsessi enda kohta, tunnistades konkreetseid väljakutseid, hinnates tehnoloogiavalikuid ja identifitseerides peamised õppetunnid. Lõpetuseks sünteesitakse lõputöö peamised tulemused ja panus, visandades konkreetseid ja põhjendatud potentsiaalsed suunad edasisteks täiustusteks ning rakenduse edasiseks arendamiseks ja uurimistööks.

### 4.1 Tulemuste analüüs

Käesoleva lõputöö peamine eesmärk, nagu märgitud peatükis 1, oli funktsionaalse, veebipõhise rakenduse arendamine, mis on mõeldud sageli töömahuka ja vigaderohke inventuuri protsessi sujuvamaks muutmiseks. Eesmärk oli saavutada see määratletud tehnoloogiavirral, mis hõlmas .NET tagarakenduse, Vue.js UI-d ja PostgreSQL andmebaasi. Tuginedes peatükis 3 kirjeldatud ja valideeritud põhitöövoogude edukale implementeerimisele, mis hõlmab kaubaartiklite andmete importimist Excelist, inventuurilehtede loomist ja haldamist, füüsilise loendamise tuge nii manuaalse sisestuse kui ka kaamerapõhise koodilugemise kaudu, kuni lõpptulemuste genereerimiseni Exceli ekspordi näol, võib kindlalt järeldada, et see peaeesmärk saavutati olulisel määral. Tulemuseks on elujõuline digitaalne lahendus, mis demonstreerib põhilisi võimekusi, mis on vajalikud fundamentaalse inventuuritsükli haldamiseks, pakkudes käegakatsutavat edasiminekut võrreldes puhtalt manuaalsete või lihtsate tabelarvutusmeetoditega.

Rakenduse edukuse detailsem hindamine konkreetsete funktsionaalsete nõuete (jaotis 2.2.1) alusel kinnitab nende täitmist ning annab ühtlasi ülevaate rakenduse operatiivsetest omadustest. Erinevate kasutajarollide implementeerimine koos turvaliste sisselogimismehhanismidega täidab olulise nõude kontrollitud juurdepääsu ja tööülesannete jaotuse osas mitme kasutajaga inventuurirakenduses, mida kinnitas ka testimine. .NET identiteediraamistiku ja hoolikalt konfigureeritud API otspunktide autoriseerimise kasutamine ennetab levinud probleeme, nagu lugejate poolt andmete juhuslik muutmine või volitamata juurdepääs administratiivfunktsioonidele. Need on riskid, mis on omased vähem struktureeritud rakendustele. Kuigi implementeeritud rollid katavad põhitöövoos jaoks vajalikud eristused, võimaldab rakenduse arhitektuur tulevikus laiendada seda detailsemate lubade mudelitele, kui seda nõuavad keerukamad organisatsioonilised struktuurid.

Funktsionaalselt kinnitati ka rakenduse võimekus edukalt importida kauba põhiaandmeid eelnevalt määratletud Exceli formaadis. See on operatiivselt elutähtis funktsioon, pakkudes vajalikku silda olemasolevate laohaldusrakenduste või põhiaandmeallikatega, vähendades märkimisväärselt esialgset seadistusaega ja minimeerides andmesisestusvigu võrreldes kaubanimistute käsitsi sisestamisega. Tagarakenduse valideerimine tagab impordi ajal elementaarse andmete terviklikkuse. Siiski, praegune implementatsioon eeldab ühtset sisendvormingut; selle tundlikkus väikeste variatsioonide või rikutud failide suhtes kujutab endast praktilist piirangut (mida käsitletakse täpsemalt jaotises 4.3), võrreldes kindlamate API-põhiste integratsioonidega.

Arendatud UI võimaldab volitatud kasutajatel luua, jälgida ja lõpetada inventuurilehti koos seotud kaubanimistute ja kasutajate määramistega, nagu detailsemalt kirjeldatud jaotises 3.2. See struktureeritud digitaalne lähenemine asendab manuaalse inventuurilehtede jaotamise või mitteplaneeritud tabelarvutuste loomise ebaselgust ja potentsiaalset korratust, pakku- des paremat selgust, planeerimisvõimalusi ning ülevaadet käimasolevatest ja tulevastest inventuuritegevustest.

Rakenduse usaldusväärne implementatsioon toetab nii otsest manuaalset koguste sisestamist kui ka kaamerapõhist koodilugemist, pakkudes operatiivset paindlikkust loendustegevustes. Integreeritud kaamerapõhine koodilugemine kiirendab kaupade identifitseerimisprotsessi selgelt kodeeritud kaupade puhul sobivates keskkondades, toetades füüsilise loenduse kiiruse suurendamise eesmärki. Manuaalne sisestus toimib mitte ainult kodeerimata kaupade

puhul, vaid ka varuvariandina, kui lugemistingimused on ebasoodsad. UI on kavandatud nende režiimide vahel sujuvaks vahetamiseks ning lugemise jõudlus osutus arendustestimise käigus standardse kooditüübiga rahuldavaks.

Lõpuks vastab valmis loendustulemuste eksportimise funktsionaalsus määratud Exceli formaati otsesele vajadusele aruandluse, arhiveerimise ja andmete edasise kasutamise järele. See standardiseeritud digitaalne väljund pakub märkimisväärselt suuremat usaldusväärsust ja tõhusust võrreldes töömahuka ja vigaderohke ülesandega konsolideerida käsitsi täidetud loenduslehti. Ekspordi struktuur on hetkel fikseeritud, kuid seda saab tulevastes iteratsioonides muuta konfigureeritavamaks, et toetada erinevaid aruandlusvajadusi.

Mittefunktsionaalsete nõuete (jaotis 2.2.2) hindamine, lähtudes [42] raamistiku evalvatsiooni kriteeriumidest, annab täiendavat konteksti rakenduse omadustele. Ülesannetepõhise testimise käigus mittetehniliste kasutajatega lugeja rollis ja rakenduse demonstreerimisel ettevõtte esindajale halduri rollis sai selgeks, et loodud kasutajaliidesed on üldiselt intuitiivsed põhiülesannete täitmiseks. See kinnitab kasutatavust, mis oli üks peamisi kvaliteedinõudeid. Funktsionaalsed ja üldiselt intuitiivsed UI-d loodi kõigi peamiste kasutajarollide töövoogude jaoks, panustades artefakti kasutatavusele. Erilist rõhku pandi loendusliidese mobiilsele reageerimisvõimele, vastates otseselt nõudele kasutada tahvelarvuteid või nutitelefone. Selle valideerimine toimus peamiselt arenduskeskkonnas ja piiratud testimisega tegelikel seadmetel. Kasutajatestimise käigus ilmnis, et UI on üldiselt intuitiivne ja lihtne omandada. Mittetehnilistelt kasutajatelt saadud tagasiside kinnitas navigeerimisloogika arusaadavust ja põhifunktsionaalsuste leitavust. Äriperspektiivi hinnang näitas, et rakendus vastab peamistele protsessuaalsetele vajadustele, toetades olulisi inventuuri töövoogusid ja pakkudes praktilisi lahendusi senistele takistustele. Kvalitatiivne tagasiside kinnitas, et rakenduse põhiline töövoog on arusaadav ning vastab ootustele. Kuigi rakendus on kasutatav ja vastab põhilistele intuitiivsuse nõuetele, viitab ulatuslikuma kasutajakogemuse testimise puudumine erinevate kasutajagruppide ja reaalsete töötingimuste lõikes vajadusele edasise iteratiivse täiustamise järele. Tulevikus oleks vajalik naturalistlikum evalvatsioon, et töövooge optimeerida.

Rakenduste turvalisuse aluseks on põhiliste turvameetmete rakendamine. See hõlmab ASP.NET Core Identity pakutavaid paroolide räsamise mehhanisme ning raamistiku tasemel kaitset levinud veebihaavatavuste vastu. API tasemel rakendati rangeid rollipõhiseid

autoriseerimiskontrolle. Kuigi need praktikad loovad kindla turvalisuse aluse, on oluline märkida, et formaalseid turvaauditeid ja läbistustestimist ei teostatud (detailsemalt käsitletud jaotises 4.3), mis tähendab, et artefakti ranget valideerimist keerukate rünnakute vastu tuleks edasises töös käsitleda.

Põhilised töövood osutuvad funktsionaalsete testimisstsenaariumide käigus usaldusväärseteks. Implementeeritud elementaarne veatöötlus, sealhulgas kliendipoolne valideerimine ja serveripoolne erinditöötlus, on osa artefakti kvaliteedi tagamisest, mille eesmärk oli parandada kasutajakogemust ja rakenduse töökindlust. Keerukamad strateegiad, nagu katkendlike võrguprobleemide käsitlemine või detailne vigade logimine, mis panustaksid põhjalikumasse hindamisse reaalses kasutustingimustes, jäid esialgselt ulatusest välja.

Valitud arhitektuurilised lahendused ja kaasaegne tehnoloogiavirn panustavad rakenduse hooldatavusele, mis on oluline kvaliteedimõõdik. Selle hindamine on pigem pidev protsess, kuid hea disaini otsused, näiteks selge komponentide eraldatus, võimaldavad artefakti lihtsamalt tulevikus täiendada, mis on kooskõlas disainiteaduse eesmärgiga luua kestlikke lahendusi. Süsteemi jõudluse ja reageerimisvõime hindamine oli esialgses etapis peamiselt kvalitatiivne ja formatiivne. Põhjalikum summeeriv-kunstlik jõudlustestimine, näiteks koormustestid, või summeeriv-naturalistlik hindamine reaalkasutajatega suuremahulise samaaegse kasutuse korral oleks vajalik enne laiemat kasutuselevõttu.

Kokkuvõttes kinnitab tulemuste põhjalik analüüs disainiteaduse ja [42] raamistiku perspektiivist, et väljatöötatud rakendus täidab edukalt lõputöö põhiülesande, luues kasuliku IT-artefakti. Teostatud evalvatsioon, ehkki peamiselt formatiivne ja kunstlikes tingimustes, demonstreeris artefakti tuumikfunktsionaalsuste toimivust. Rakendus pakub funktsionaalset digitaalset tööriista, mis tegeleb paljude traditsiooniliste, integreerimata inventuurimeetoditega kaasnevate peamiste ebatõhusustega.

## **4.2 Lahenduse tugevused**

Arendatud veebipõhine inventuurirakendus, võrreldes otse taustauuringu käigus tuvastatud levinud manuaalsete või lihtsate tabelarvutustest sõltuvate protseduuridega, omab mitmeid kaalukaid tugevusi, mis pakuvad käegakatsutavat operatiivset kasu.

Kõige otsesem eelis seisneb otsast lõpuni töövoos sujuvamaks muutmises. Andmete impor-

timise, loenduslehtede määramise, tulemuste konsolideerimise ja aruannete eksportimise automatiseerimisega kõrvaldab rakendus mitmed manuaalse sekkumise punktid. Näiteks tulemuste konsolideerimine arvukatelt paber kandjal lehtedelt või eraldi digitaalsetest failidest, mis suurte inventuuride puhul võib võtta tunde, väheneb ekspordifunktsiooni abil sekunditele. Lisaks sihib integreeritud koodilugemise funktsioon otseselt aeganõudvat protsessi esemete käsitsi tuvastamiseks ja koguste otsimiseks paberloenditest, potentsiaalselt suurendades oluliselt füüsilise loenduse faasi kiirust, vähendades samal ajal ka lugejate kognitiivset koormust.

Manuaalsed protsessid on kurikuulsalt alid inimlikele vigadele, sealhulgas transkriptsioonivead, arvutusvead, loetamatu käekiri või aegunud kaubanimistute kasutamine. Tabelarvutuslahendused, kuigi paremad, on endiselt vastuvõtlikud valemivigadele, valedel lahtriviidetele või andmete rikkumisele juhusliku kustutamise või sortimisvigade tõttu. See rakendus maandab neid riske oluliselt, tsentraliseerides andmed, vähendades manuaalset transkriptsiooni impordi, ekspordi ja skannimise kaudu, jõustades andmetüüpe ja põhilist valideerimist sisestuspunktides ning teostades serveripoolseid arvutusi järjepidevalt. Selle tulemuseks on usaldusväärsemad laoseisuandmed, mis mõjutavad otseselt operatiivotsuste tegemist, finantsilist täpsust ja auditeeritavust.

PostgreSQL andmebaas toimib ühtse, autoriteetse hoidlana kogu inventuuriteabe jaoks, sealhulgas kauba põhiaandmed, loenduslehed, kasutajate määramised ja loendustulemused. See ühtne tõellikas kõrvaldab olulised väljakutsed, mis on seotud potentsiaalselt kümnete või sadade hajutatud, versioonihallatud tabelarvutusfailide haldamisega, mis on sageli salvestatud mitmesse asukohta või e-kirjavahetusse. Tsentraliseerimine lihtsustab andmete hankimist aruandluseks, muudab ajaloolise analüüsi teostatavaks ja loob selge auditijälje aluse, kuigi detailne logimine vajab edasist arendust.

Rollipõhine juurdepääsukontrolli mudel pakub granulaarset kontrolli selle üle, kes milliseid toiminguid teha saab, näiteks seda, et ainult haldurid saavad luua loenduslehti. Turvaline kasutaja autentimine tagab, et rakendusele pääsevad juurde ainult volitatud töötajad. See kujutab endast märkimisväärset edasiminekut turvalisuses ja protsessikontrollis võrreldes jagatud, sageli halvasti kaitstud tabelarvutuste või paberdokumentidega. Lisaks saab rakenduses tehtud toiminguid potentsiaalselt logida üksikute kasutajate vastu.

.NET, Vue.js koos TS-iga ja PostgreSQL valik tagab, et rakendus kasutab ajakohaseid, kindlaid ja laialt levinud tehnoloogiaid. See toob kaasa arvukalt eeliseid, nagu juurdepääs ulatuslikule veebidokumentatsioonile ja suurtele arendajakogukondadele toe saamiseks, arvukate kolmandate osapoolte teekide kättesaadavus funktsionaalsuse laiendamiseks, regulaarsed turvapaigad ja uuendused tootjatelt ning arendusparadigma, mis edendab koodikvaliteeti ja lihtsustab pikaajalist hooldust ja funktsioonide täiustusi. See vähendab oluliselt tehnoloogilise vananemise riski võrreldes vanemate või vähem levinud platvormidega.

Olles veebirakendus, muudab rakenduse olemuslikult kättesaadavaks mis tahes seadmest, mis on varustatud kaasaegse veebilehitsejaga, olenemata aluseks olevast operatsioonisüsteemist. See võimaldab halduritel kasutada seadistamiseks ja analüüsiks laua- või sülearvuteid, samas kui lugejad saavad kasutada kaasaskantavamaid tahvelarvuteid või isegi nutitelefone otse lao- või laoruumis. See kõrvaldab vajaduse paigaldada ja hallata spetsiaalset tarkvara igas kliendiseadmes.

Need kombineeritud tugevused positsioneerivad väljatöötatud rakenduse mitte pelgalt funktsionaalse prototüübina, vaid platvormina, mis pakub olulisi, mõõdetavaid edusamme tõhususes, täpsuses, turvalisuses ja hallatavuses võrreldes traditsiooniliste inventuuri meetoditega, mida see asendada püüab.

### **4.3 Lahenduse piirangud**

Kuigi ettevõtte esindajatelt ja mittetehnilistelt kasutajatelt saadud tagasiside oli positiivne, tuleb arvestada, et hindamine viidi läbi kontrollitud keskkonnas, mitte põhjaliku kasutajatestina reaalses töökeskkonnas. Seetõttu ei pruugi kõik potentsiaalsed kasutatavusprobleemid või töövoogude kitsaskohad olla veel ilmnenu.

Rakenduse praegune sõltuvus konkreetsetest Exceli failistruktuuridest nii põhiantmete importimisel kui ka tulemuste eksportimisel kujutab endast olulist operatiivset haprust. Igasugune muudatus lähte failide struktuuris, näiteks muudetud veergude järjekord või nimed ERP ekspordiuuendusest, põhjustaks tõenäoliselt impordi ja ekspordi tõrkeid, nõudes arendaja sekkumist rakenduse parsimisloogika uuendamiseks. See tihe sidumine konkreetse failivorminguga vastandub teravalt API-põhise integratsiooni pakutavale kindlusele, mis suudab andmestruktuuri muutustega paindlikumalt toime tulla versioonimise ja läbirää-

kimiste kaudu. See piirang tähendab, et praegune rakendus nõuab ranget kinnipidamist määratletud mallidest, lisades töövoogu manuaalse sammu ja potentsiaalse tõrkepunkti.

Kuigi funktsionaalne, tugineb kaamerapõhine skannimine brauseri API implementatsioonidele ja JS teekidele dekodeerimiseks. Selle reaalne jõudlus on seetõttu vastuvõtlik paljudele välistele muutujatele, mis ei ole rakenduse kontrolli all, nagu seadme kaamera kvaliteet ja teravustamisvõime, ümbritseva valguse tase inventuurirakenduses, koodide suurus ja trükikvaliteet ning isegi kasutatav konkreetne brauser. Selle muutujate matriksi ulatuslik testimine oli projekti raames võimatu. Järelikult on olemas stsenaariume, kus skannimine võib olla aeglane, ebatäpne või täielikult ebaõnnestuda, sundides kasutajaid tagasi manuaalse sisestusele ja põhjustades potentsiaalselt frustratsiooni. Lisaks toetab implementatsioon praegu ainult üht kindlat, eelnevalt määratletud koodisümboolikat; kui organisatsioon kasutab mitut sümboolikat, oleks skannimisfunktsioon nende kaupade jaoks kasutuskõlbmatu.

Rakendus läbis funktsionaalse valideerimise peamiselt ühe kasutaja või madala samaaegsusega stsenaariumides. Selle võimet säilitada jõudlust ja reageerimisvõimet märkimisväärse koormuse all, mida oodatakse täismahus inventuuri ajal suures organisatsioonis, nt 30+ kasutajat sisestavad samaaegselt loendustulemusi mitmele loenduslehele, ei ole kontrollitud spetsiaalse koormustestimisega. Potentsiaalsed kitsaskohad võivad tekkida erinevates valdkondades, nagu samaaegsed andmebaasi kirjutamisoperatsioonid, mis põhjustavad lukustamist või konkurentsi, .NET tagarakenduse API, mis pingutab paljude samaaegsete päringute käsitlemisel, põhjustades suurenenud latentsusaega, või isegi Vue.js UI muutub aeglaseks suurte kaupade või loenduslehtede loendite renderdamisel. Ilma koormustestimiseta on oht jõudluse halvenemisele või ebastabiilsusele tippkasutusperioodidel.

Kuigi rakendatakse elementaarseid turvameetmeid, ei ole lõputöö raames väljatöötatud rakendust allutatud ametlikele turvaaudititele, haavatavuste lugemisele ega läbistustestimisele. See tähendab, et potentsiaalseid haavatavusi võib esineda.

Praegune nõue, et administraator peab käsitsi looma kõik kasutajakontod ja tegelema paroolide lähtestamisega, kujutab endast selget halduslikku kitsaskohta. Organisatsioonides, kus on palju inventuuri töötajaid või suur personalivoolavus, muutub see manuaalne protsess ebatõhusaks ja võib põhjustada viivitusi rakendusele juurdepääsu andmisel. Kuigi see



manuaalne protsess oli projekti jaoks teadlik lihtsustus, takistab see oluliselt rakenduse kasutatavust ja skaleeritavust administraatorite jaoks suuremates seadetes.

Nende piirangute tunnistamine on oluline. Need rõhutavad, et kuigi põhikontseptsioon on tõestatud, on selle rakenduse täielikult usaldusväärseks, skaleeritavaks ja turvaliseks inventuurilahenduseks muutmiseks vajalik märkimisväärne edasine arendus, testimine ja tugevdamine.

## **4.4 Mõtiskelu arendusprotsessist**

Selle inventuurirakenduse arendustekond, mis oli küll peamiste eesmärkide saavutamisel lõppkokkuvõttes edukas, pakkus arvukalt väärtuslikke õpikogemusi ja rõhutas mitmeid tarkvaraarenduse praktilisi aspekte, eriti konkreetse kliendiga töötades ja erinevaid tehnoloogiaid integreerides. Protsess järgis suures osas kavandatud iteratiivseid vahe-eesmärke, võimaldades paindlikkust ja kohanemist pideva õppimise ja tagasiside põhjal. Väärtuslikuks osutus tagasiside kogumine ettevõtte esindajatelt ja mittetehnilistelt kasutajatelt, mis suunas tähelepanu olulistele kasutajakesksetele aspektidele, mida tehnilisest vaatepunktist ei pruukinud esialgu märgata.

### **4.4.1 Ilmnenud väljakutsed**

Peamine väljakutse tulenes kliendi operatiivteadmiste ja mõnikord kaudsete nõuete tõlkimisest konkreetseteks tehnilisteks spetsifikatsioonideks. Varasemad iteratsioonid vajasid täpsustamist, kuna ühine arusaam soovitud töövoost arenes.

Kolmanda osapoole JS teegi integreerimine koodiskannimiseks Vue.js raamistikku oli tehniliselt nõudlik; spetsiifilised takistused hõlmasid kaameravoo juurdepääsu usaldusväärset haldamist ning kaameraõiguste küsimist ja käsitlemist järjepidevalt erinevates brauserites. Suhtlusvoog kliendiga vajab samuti kohandamist; tehniliste küsimuste selge sõnastamise õppimine ja vastuste saamise aja mõistmine ärikeskkonnas olid projekti käigus lihvitud peamised pehmed oskused. Lisaks nõudis selgelt määratletud ja vastupidava API lepingu loomine ja säilitamine eraldiseisvalt arendatud .NET tagarakenduse ja Vue.js UI vahel hoolikat spetsifitseerimist ja struktureeritud suhtlusprotokolle. See muutus eriti kriitiliseks, kui ühe komponendi muudatustel olid otsesed tagajärjed teisele, nõudes seeläbi sünkroniseeritud kohandusi sujuva koostalitlusvõime tagamiseks.

#### 4.4.2 Tehnoloogiavirna hindamine

Valitud tehnoloogiavirna osutus suures osas tõhusaks ja kinnitas esialgseid valikukriteeriume. .NET koos EF Core-iga pakkus produktiivset keskkonda tagarakenduse API ehitamiseks, pakkudes küpseid teeke andmekäitluseks, turvalisuseks ja sõltuvuste süstimiseks. PostgreSQL toimis usaldusväärse ja tõhusa relatsioonandmebaasina, mida oli lihtne Dockeriga hallata. UI-s lihtsustasid Vue.js komponendipõhine arhitektuur, reaktiivsussüsteem ning TS-i, püüdes varakult kinni potentsiaalsed tüübivead, ja Vite-i, pakkudes peaaegu kohest moodulite kiirvahetust, pakutav täiustatud arendajakogemus oluliselt UI arendust.

Docker oli hindamatu järjepideva arendus- ja testimiskeskkonna tagamisel, eriti andmebaasi jaoks, lihtsustades seadistamist erinevatel masinatel. Lisaks hõlbustasid Dockeri kontaineriseerimisvõimalused rakenduse lihtsat juurutamist. API-keskne lähenemine hõlbustas tagarakenduse loogika fookuseeritud testimist ja võimaldas teatud määral paralleelset UI ja tagarakenduse tööd, kuigi see nõudis kindlat API dokumentatsiooni ja suhtlust.

#### 4.4.3 Õppetunnid

See projekt tugevdas jõuliselt mitmeid tarkvaraarenduse põhiprintsiipe. Esiteks, iteratiivse täiustamise ja pideva klienditagasiside absoluutne vajalikkus projektides, kus nõuded ei pruugi alguses täielikult määratletud olla; esialgne prototüüp arenes oluliselt kliendi sihendi põhjal. Teiseks, seadme kaameratega integreerimine veebistandardite kaudu, kuigi võimas, lisab keerukuse ja varieeruvuse kihte, mis on seotud brauseri implementatsioonide, kasutajaõiguste ja riistvara kvaliteediga, mida tuleb hoolikalt hallata. Kolmandaks, tõhus suhtlus ei tähenda ainult info edastamist, vaid ka ühise arusaama tagamist, eriti ärinõuete ja tehniliste lahenduste vahel tõlkides. Viimaks demonstreeris projekt kaasaegsete, hästi dokumenteeritud raamistike valimise praktilisi eeliseid, mis kiirendavad arendust sisseehitatud funktsioonide ja tugeva kogukonnatõe kaudu paratamatute takistuste ilmnemisel.

Sisuliselt, kuigi tehnilised takistused ja suhtluskohandused olid protsessi osa, võimaldas metoodika nende lahendamist ning tehnoloogiavalikud pakkusid vajalikke tööriistu soovitud rakenduse tõhusaks ehitamiseks käesoleva lõputöö projekti piirangute raames.

## **4.5 Tulevased tööd**

Selle inventuurirakenduse edukas arendamine avab arvukalt võimalusi tulevasteks täiustusteks, eesmärgiga arendada sellest terviklikum, kindlam ja integreeritum ärintööriist. Tuvastatud piirangute ja potentsiaalsete lisaväärtuste põhjal saab tulevased tööd liigitada kolme peamisse valdkonda.

### **4.5.1 Funktsionaalsed täiustused**

Praegu tugineb analüüs andmete Excelisse eksportimisele. Visualiseerimisteede integreerimine konfigureeritavate töölaudade loomiseks rakenduses endas annaks halduritele kohese, reaajas ülevaate loenduse edenemisest, hälvete analüüsist, asukohapõhisest täpsusest ja ajaloolistest trendidest, suurendades oluliselt tööriista analüütilist väärtust ja vähendades sõltuvust välisest tarkvarast.

Oluline laiendus on lisada funktsionaalsust kaugemale lihtsast lugemisest. See hõlmab oodatavate laoseisude importimist ja töövoo implementeerimist lahknevuste esiletõstmiseks oodatud ja loetud koguste vahel. Funktsioonid võiksid hõlmata märkmete lisamist, uurimisülesannete määramist ja mehhanismide pakkumist korrigeerimiste ametlikuks kinnitamiseks, toetades seeläbi täielikku laovarude võrdlusprotsessi, mis on sageli inventuuri lõppeesmärk.

Pideva võrguühenduse nõude piirangu lahendamiseks oleks kasulik loendusmooduli refaktorimine progressiivseks veebirakenduseks. See võimaldaks lugejatel laadida oma määratud ülesandeid, teostada loendusi ja salvestada andmeid lokaalselt isegi võrguühenduseta, andmete automaatse sünkroniseerimisega serverisse ühenduse taastumisel. See parandab oluliselt praktilist kasutatavust ja usaldusväärsust keerulistes füüsilistes keskkondades.

Andmemudeli täiustamine, et see sisaldaks konkreetseid kauba hoiukohti, nagu vahekäik, riiul või kast, ja nende sidumine loenduslehtedega võimaldaks sihipärasemaid inventuure. See võiks võimaldada selliseid funktsioone nagu rakenduse poolt soovitatud optimeeritud loendustekonnad või kaupade grupeerimine asukoha järgi loendusliideses, potentsiaalselt suurendades veelgi lugeja tõhusust.

#### **4.5.2 Tehnilised täiustused**

Enne laiemat kasutuselevõttu on hädavajalik süstemaatiline jõudluse ja koormuse testimine. See tuvastaks potentsiaalsed kitsaskohad suure samaaegse kasutajakoormuse ja suurte andmemahtude korral, tagades rakenduse stabiilsuse ja reageerimisvõime inventuuri tipptegevuste, nagu aasta lõpu inventuurid, ajal. Tulemused suunaksid vajalikke optimeerimisi andmebaasi indekseerimises, päringute tõhususes, API otspunktide loogikas või vahemälustrateegiates.

Kriitiline samm tootmisvalmiduse suunas on põhjalike turvahinnangute läbiviimine. See hõlmab väliste ekspertide kaasamist läbistustestimiseks, koodi haavatavuse analüüsiks, sõltuvuste haavatavuste skannimiseks ning autentimis- ja autoriseerimisloogika ülevaatamiseks, et tuvastada ja parandada potentsiaalsed turvaaukud enne tundlike operatiiv- või finantsandmete käsitlemist.

Täiustatud koodiskannimise võimekusi võiks lisada, integreerides toe mitmele koodisümboolikale ja uurides täpsemaid skannimisteeke või konfiguratsioone, et parandada usaldusväärsust erinevates seadmetes ja tingimustes.

Lõpuks on kasutajahalduse täiustamine märkimisväärse halduskoormuse leevendamiseks oluline kaalutlus. Kuigi täielik iseregistreerimine võib vajada hoolikat kaalumist turvalisuse ja kinnitamise töövoogude osas, jääks kasutajate loomine haldusülesandeks või seda võiks tulevikus integreerida olemasolevate ettevõtte autentimisteenustega, et konto loomist sujuvamaks muuta.

#### **4.5.3 Juurutus- ja integratsiooniparandused**

See kujutab endast vaieldamatult kõige olulisemat lisaväärtust loovat täiustust. Kindlate, kahesuunaliste API integratsioonide arendamine siht ERP-süsteemi või spetsiaalsete laohaldussüsteemidega kõrvaldaks täielikult sõltuvuse manuaalsest Exceli failivahetusest. See võimaldaks kauba põhiaandmete, oodatavate laoseisude ja kinnitatud laokorrigeerimiste reaalajas sünkroniseerimist, luues tõeliselt integreeritud ja tõhusa otsast lõpuni digitaalse protsessi. See nõuaks tõenäoliselt spetsiifiliste konnektorite või vahevara adapterite arendamist iga siht ERP-süsteemi jaoks.

Juurutuste jaoks, mis nõuavad kõrget käideldavust, automaatset skaleerimist vastavalt koormusele ning lihtsustatud uuenduste väljalaskeid ja tagasivõtmisi, pakuks Dockeripõhise juurutuse migreerimine konteinerite orkestreerimisplatvormile olulisi operatiivseid eeliseid, eriti kui rakendus muutub selle ettevõtte laohaldusprotsessi kriitiliseks komponendiks.

Nende tulevaste sammude prioritseerimine sõltuks konkreetsetest ärivajadustest, kuid ERP integratsiooni, turvalisuse tugevdamise ja jõudluse valideerimisega tegelemine on tõenäoliselt kõige kriitilisemad sammud rakenduse edasiseks täiustamiseks kindlateks, ettevõtte tasemel inventuuri operatsioonideks.

## 5 Kokkuvõte

Käesolevas bakalaureusetöös käsitleti inventuuriprotsesside optimeerimise väljakutset, keskendudes traditsiooniliste manuaalsete meetoditega kaasnevale ebaefektiivsusele ja vigade potentsiaalile. Töö peamiseks eesmärgiks oli disainiteaduse uurimismetoodikat rakendades kavandada ja arendada veebipõhine tarkvararakendus, mis suudaks inventuuri läbiviimist oluliselt tõhustada ja digiteerida.

Selle eesmärgi saavutamiseks loodi IT-artefakt, mis kasutab .NET platvormil põhinevat tagarakenduse API, Vue.js ja TypeScriptiga arendatud eesrakendust ning PostgreSQL andmebaasi haldussüsteemi. Rakenduse disainimisel järgiti API-keskset lähenemist. Implementeeritud funktsionaalsus hõlmab turvalist kasutajahaldust koos erinevate rollidega, kaubaartiklite andmete importi Exceli failidest, inventuuriseansside loomist ja haldamist, loenduskannete sisestamist nii käsitsi kui ka integreeritud kaamerapõhise koodiskaneerimise abil ning inventuuritulemuste eksporti struktureeritud Exceli formaadis. Serveripoolne äriloogika jõustamine tagab andmete korrektsuse ning arendus- ja juurutusprotsessides kasutati CI/CD GitHub Actionsi kaudu, koos Dockeri konteineriseerimisega, kusjuures nginx teenindab eesrakendust.

Valminud veebirakendus implementeerib edukalt inventuuri põhilised töövood, mida kinnitab läbiv testimine ja esmane valideerimine. Arendatud rakendus pakub seeläbi elujõulist ja efektiivset alternatiivi manuaalsetele inventuurimeetoditele, tsentraliseerides andmehalduse, vähendades inimlike vigade ohtu ning pakkudes mobiilse interaktsiooni võimalusi täpsuse ja operatiivse tõhususe parandamiseks. Lõputöö käigus saavutati seatud eesmärgid, luues tugeva tehnoloogilise ja kontseptuaalse vundamendi edasisteks täiustusteks, mille hulka kuuluvad näiteks täpsemad analüütika- ja aruandlusvõimalused ning otsene integratsioon ERP-süsteemidega.

## Kasutatud kirjandus

- [1] Microsoft. *.NET*. Kasutatud: 11.05.2025. URL: <https://dotnet.microsoft.com/en-us/learn/dotnet/what-is-dotnet>.
- [2] AKIT. *API*. Kasutatud: 11.05.2025. URL: <https://akit.cyber.ee/term/3088>.
- [3] AKIT. *Arhitektuur*. Kasutatud: 28-04-2025. URL: <https://akit.cyber.ee/term/2673>.
- [4] Microsoft. *ASP.NET Core*. Kasutatud: 11.05.2025. URL: <https://learn.microsoft.com/en-us/aspnet/core/introduction-to-aspnet-core>.
- [5] Microsoft. *ASP.NET Core Identity*. Kasutatud: 11.05.2025. URL: <https://github.com/dotnet/AspNetCore/tree/main/src/Identity>.
- [6] Axios. "What is Axios?" Kasutatud: 16.05.2025. URL: <https://axios-http.com/docs/intro>.
- [7] AKIT. *CI/CD*. Kasutatud: 11.05.2025. URL: <https://akit.cyber.ee/term/16056>.
- [8] Ecma International. *C# Language Specification*. Kasutatud: 11.05.2025. URL: <https://ecma-international.org/wp-content/uploads/ECMA-334-7th-edition-december-2023.pdf>.
- [9] Git. *Git*. Kasutatud: 11.05.2025. URL: <https://git-scm.com/>.
- [10] The Tech Terms Computer Dictionary. *Frontend*. Kasutatud: 11.05.2025. URL: <https://techterms.com/definition/frontend>.
- [11] Microsoft. *Entity Framework Core*. Kasutatud: 11.05.2025. URL: <https://learn.microsoft.com/en-us/ef/core/>.
- [12] AKIT. *ERP*. Kasutatud: 11.05.2025. URL: <https://akit.cyber.ee/term/9223>.
- [13] The Tech Terms Computer Dictionary. *Excel*. Kasutatud: 11.05.2025. URL: <https://techterms.com/definition/excel>.
- [14] Docker. *Docker*. Kasutatud: 11.05.2025. URL: <https://docs.docker.com/get-started/docker-overview/>.
- [15] Cambridge Dictionary. *Stocktaking*. Kasutatud: 01.05.2025. URL: <https://dictionary.cambridge.org/dictionary/english/stocktaking>.
- [16] AKIT. *Iteratiivarendus*. Kasutatud: 16.05.2025. URL: <https://akit.cyber.ee/term/15534>.
- [17] AKIT. *JavaScript*. Kasutatud: 11.05.2025. URL: <https://akit.cyber.ee/term/3662>.
- [18] AKIT. *Kestustugi*. Kasutatud: 11.05.2025. URL: <https://akit.cyber.ee/term/2331>.

- [19] AKIT. *Läbivestimine*. Kasutatud: 11.05.2025. URL: <https://akit.cyber.ee/term/12075>.
- [20] F5. *nginx*. Kasutatud: 11.05.2025. URL: <https://nginx.org/en/>.
- [21] AKIT. *ORM*. Kasutatud: 11.05.2025. URL: <https://akit.cyber.ee/term/8533>.
- [22] E. S. M. Morote. *Pinia*. Kasutatud: 16.05.2025. URL: <https://pinia.vuejs.org/introduction.html>.
- [23] PostgreSQL. *PostgreSQL*. Kasutatud: 11.05.2025. URL: <https://www.postgresql.org/>.
- [24] AKIT. *Pöördproksi*. Kasutatud: 11.05.2025. URL: <https://akit.cyber.ee/term/4886>.
- [25] AKIT. *REST*. Kasutatud: 11.05.2025. URL: <https://akit.cyber.ee/term/10843>.
- [26] AKIT. *Tarkvara teenusena*. Kasutatud: 14.05.2025. URL: <https://akit.cyber.ee/term/3229>.
- [27] AKIT. *Scrum*. Kasutatud: 16.05.2025. URL: <https://akit.cyber.ee/term/16778>.
- [28] The Tech Terms Computer Dictionary. *Backend*. Kasutatud: 11.05.2025. URL: <https://techterms.com/definition/backend>.
- [29] Heap. *Tech Stack*. Kasutatud: 13.05.2025. URL: <https://www.heap.io/topics/what-is-a-tech-stack>.
- [30] Microsoft. *TypeScript*. Kasutatud: 11.05.2025. URL: <https://www.typescriptlang.org/>.
- [31] AKIT. *Kasutajaliides*. Kasutatud: 11.05.2025. URL: <https://akit.cyber.ee/term/6889>.
- [32] VoidZero. *Vite*. Kasutatud: 11.05.2025. URL: <https://vite.dev/>.
- [33] E. You. *Vue.js*. Kasutatud: 11.05.2025. URL: <https://vuejs.org/>.
- [34] RGIS. "Why is Stocktaking Important?" Kasutatud: 15.05.2025. URL: <https://www.rgis.my/industry-perspective/why-is-stocktaking-important/>.
- [35] 8Stock. "Top 10 Common Inventory Problems and How to Solve Them". Kasutatud: 22.04.2025. URL: <https://8stock.co/tips/top-10-common-inventory-problems-and-how-to-solve-them/>.
- [36] M. Ahmed. "Importance of Stocktaking". URL: <https://www.scmdojo.com/importance-stocktaking-purpose-of-stocktaking/>.
- [37] A. Jenkins. "20 Common Inventory Management Challenges". Kasutatud: 17.04.2025. URL: <https://www.netsuite.com/portal/resource/articles/inventory-management/inventory-management-challenges.shtml>.
- [38] O. Munro. "Stocktaking". URL: <https://www.unleashedsoftware.com/inventory-management-guide/stocktaking/>.
- [39] R. Butler. "How inventory management software is solving the 3 big problems of manual stock control". Kasutatud: 12.04.2025. URL: <https://harmoniq.com.au/inventory-management-software-solving-manual-stock-control-problems/>.



- [40] M. Mülla. "Some guidelines for stock-taking". Kasutatud: 22.04.2025. URL: <https://www.grantthornton.ee/en/insights1/some-guidelines-for-stock-taking/>.
- [41] A. R. Hevner *et al.* „Design Science in Information Systems Research“. *MIS Quarterly* 28.1 (2004), lk. 75–105. URL: <https://www.jstor.org/stable/25148625>.
- [42] J. Venable, J. Pries-Heje ja R. Baskerville. „FEDS: a Framework for Evaluation in Design Science Research“. *European Journal of Information Systems* 25.1 (2016), lk. 77–89. URL: <https://doi.org/10.1057/ejis.2014.36>.
- [43] OWASP. *Password Storage Cheat Sheet*. URL: [https://cheatsheetseries.owasp.org/cheatsheets/Password\\_Storage\\_Cheat\\_Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/Password_Storage_Cheat_Sheet.html).
- [44] E. Karataş. *Understanding N-Tier Architecture: Building Robust and Scalable Applications*. URL: <https://medium.com/@segekaratas/understanding-n-tier-architecture-building-robust-and-scalable-applications-62db30a40b5>.
- [45] Microsoft. "N-tier architecture style". URL: <https://learn.microsoft.com/en-us/azure/architecture/guide/architecture-styles/n-tier>.
- [46] Stackify. "What is N-Tier Architecture? How It Works, Examples, Tutorials, and More". URL: <https://stackify.com/n-tier-architecture/>.
- [47] IBM. "What is three-tier architecture?" URL: <https://www.ibm.com/think/topics/three-tier-architecture>.
- [48] Indeed Editorial Team. "N-Tier Architecture Style: Definition and Advantages". URL: <https://www.indeed.com/career-advice/career-development/n-tier-architecture>.
- [49] JGraph. *draw.io*. Kasutatud: 11.05.2025. URL: <https://github.com/jgraph/drawio>.
- [50] M. Leon *et al.* *ClosedXML*. Kasutatud: 03.06.2025. URL: <https://github.com/ClosedXML/ClosedXML>.
- [51] .NET Foundation. *xUnit.net*. Kasutatud: 03.06.2025. URL: <https://xunit.net/>.
- [52] Clarius, Manas ja InSTEDD. *Moq*. Kasutatud: 03.06.2025. URL: <https://github.com/devlooped/moq>.
- [53] Vue. "What is Vue?" URL: <https://vuejs.org/guide/introduction>.
- [54] Chaewonkong. "TypeScript, What Good For?" URL: <https://medium.com/@chaewonkong/typescript-what-good-for-8240dc9173c7>.
- [55] Vite. "Vite: Overview". URL: <https://vite.dev/guide/>.
- [56] A. V. Minhaz. *Html5-QRCode*. Kasutatud: 03.06.2025. URL: <https://github.com/mebjas/html5-qrcode>.
- [57] Anthony Fu ja Matías Capeletto. *Vitest*. Kasutatud: 03.06.2025. URL: <https://vitest.dev/>.
- [58] PostgreSQL. "PostgreSQL: About". URL: <https://www.postgresql.org/about/>.
- [59] RedHat. "What is CI/CD?" URL: <https://www.redhat.com/en/topics/devops/what-is-ci-cd>.

- [60] V. Silverthorne. "*10 Reasons why your business needs CI/CD*". URL: <https://about.gitlab.com/blog/2022/02/15/ten-reasons-why-your-business-needs-ci-cd/>.
- [61] S. Susnjara. "*What is containerization?*" URL: <https://www.ibm.com/think/topics/containerization>.
- [62] Amazon Web Services. "*What is containerization?*" URL: <https://aws.amazon.com/what-is/containerization/>.
- [63] Xaltos Technologies. *Stock and Inventory Online*. Kasutatud: 05.05.2025. URL: <https://chester-sw.com/>.
- [64] Allegro. *Ralph*. Kasutatud: 02.04.2025. URL: <https://github.com/allegro/ralph/>.
- [65] Allegro. *Ralph*. Kasutatud: 02.04.2025. URL: <https://ralph-ng.readthedocs.io/en/stable/installation/installation/#docker-installation-experimental>.
- [66] INDEPNET. *GLPI Project*. Kasutatud: 02.04.2025. URL: <https://glpi-project.org/>.
- [67] OpenBoxes. *OpenBoxes*. Kasutatud: 02.04.2025. URL: <https://openboxes.com/>.
- [68] Grokability. *Snipe-IT*. Kasutatud: 02.04.2025. URL: <https://snipeitapp.com/>.
- [69] InvenTree. *InvenTree*. Kasutatud: 02.04.2025. URL: <https://inventree.org/>.

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

Mina, Anton Meenov

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Veebi-põhise inventuurirakenduse arendamine“, mille juhendaja on Siim Rebane
  - 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 veebikeskonna 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.

04.06.2025

---

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