

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Karl-Jaagup Rõngas 225280IADB

Laohaldussüsteemi infotahvel

Bakalaureusetöö

Juhendaja: Kersti Antoi
MSc

Tallinn 2023

Autorideklaratsioon

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Karl-Jaagup Rõngas

23.10.2022

Annotatsioon

Lõputöö eesmärgiks on luua interaktiivne infotahvli rakendus, mis annaks kiire ülevaate lao tööpäevast ning seda iseloomustavatest mõõdikutest ja oleks abivahendiks lao tööprotsesside juhtimisel ning koordineerimisel.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 27 leheküljel, 6 peatükki, 14 joonist, 1 tabeli.

Abstract

Dashboard for Warehouse Management System

The purpose of this thesis is to develop an interactive data dashboard application that provides the user a quick overview of daily processes and key metrics of the warehouse. The application provides up-to-date information to help with management and business decisions.

The thesis is in Estonian and contains 27 pages of text, 6 chapters, 14 figures, 1 table.

Lühendite ja mõistete sõnastik

AS	Aktsiaselts
CSS	<i>Cascading style sheets</i> , veebilehtede kujunduse märgistuskeel
CTE	<i>Common table expression</i> , ühine tabeli avaldis
DOM	<i>Document object model</i> , dokumendi objektimudel
EDI	<i>Electronic data interchange</i> , elektroonne andmevahetus
GET	HTTP protokoll meetod teabe pärimiseks
HEX	Kuueteistkümnendsüsteem
HTML	<i>Hypertext markup language</i> , veebilehtede sisu märgistuskeel
HTTP	<i>Hypertext transfer protocol</i> , protokoll teabe edastamiseks arvutivõrkudes
IT	Infotehnoloogia
JavaScript	Skriptimiskeel veebilehtede kasutajaliideste loomiseks
JSON	<i>JavaScript object notation</i> , andmevahetusvorming
PHP	Skriptimiskeel serverirakenduste loomiseks
POST	HTTP protokoll meetod teabe saatmiseks
SQL	<i>Structured query language</i> , struktuurpäringukeel
Table-valued funktsioon	Kasutaja defineeritud funktsioon andmebaasis, mis tagastab tabeli tüüpi tulemuse
WebGL	<i>Web graphics library</i> , JavaScripti rakendusliides graafika renderdamiseks veebibrauseris
WMS	<i>Warehouse management system</i> , laohaldustarkvara
XML	<i>Extensible markup language</i> , andmevahetusvorming

Sisukord

1 Sissejuhatus	10
2 Taustinformatsioon.....	11
2.1 Laohoone ja selle ülesehitus	11
2.2 Töökorraldus laos	12
2.3 Olemasolev laohaldustarkvara.....	12
2.3.1 Komplekteerimine	13
2.3.2 Sissetulnud kauba kontroll	14
2.3.3 Ladustamine	15
2.3.4 Kauba siire.....	15
2.3.5 Aktiivide täitmine	15
2.3.6 Inventuur.....	15
3 Probleemi püstitus ja projekti eesmärk.....	16
3.1 Ülesande püstitus	16
3.2 Lähtetingimused	16
3.3 Eesmärk	16
3.4 Metoodika.....	16
4 Probleemi analüüs	18
4.1 Andmemudeli kirjeldus	18
4.2 Lao tööd iseloomustavad mõõdikud ja nende olulisus.....	19
4.3 Nõuded loodavale lahendusele	20
4.3.1 Funktsionaalsed nõuded	20
4.3.2 Mittefunktsionaalsed nõuded.....	21
4.4 Olemasolevad lahendused	22
5 Lahenduse realiseerimine	23
5.1 Arendustöö käik.....	23
5.2 Kasutatud tehnoloogiad	23
5.3 Mõõdikute arvutamine.....	24
5.4 Serverirakendus mõõdikute publitseerimiseks	25
5.5 Kasutajaliidese lahendus.....	25

5.5.1 Koondvaade	25
5.5.2 Põrandaplaanide visualiseerimine	33
5.5.3 Lahenduse testimine	34
5.6 Kasutajate tagasiside ja võimalikud edasiarendused	35
5.7 Majandusliku kasu analüüs.....	35
Kokkuvõte	36
Kasutatud kirjandus	37
Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks	38
Lisa 2 – Fotod	39
Lisa 3 – Infotahvli koondvaade	42
Lisa 4 – Päringu näide	43

Jooniste loetelu

Joonis 1. Infrastruktuuri skeem	13
Joonis 2. Kontrollitud kauba etikett.....	14
Joonis 3. Lihtsustatud andmebaasiskeem	18
Joonis 4. Infokuvade selgitused - ikoonid koondvaate päises	26
Joonis 5. Infokuvade selgitused - sektordiagrammid koondvaate päises	27
Joonis 6. Infokuvade selgitused - komplekteerijate järjestus	28
Joonis 7. Infokuvade selgitused - kauba vastuvõtud	28
Joonis 8. Infokuvade selgitused - muud laotoimingud	29
Joonis 9. Infokuvade selgitused - kauba ladustamised	29
Joonis 10. Infokuvade selgitused - pooleliolevad ja alustamata noppelehed	30
Joonis 11. Infokuvade selgitused - määramata noppelehed	31
Joonis 12. Noppelehe määramise tegevusdiagramm	32
Joonis 13. Infokuvade selgitused - täidetud noppelehed	32
Joonis 14. I korruse põrandaplaan	34

Tabelite loetelu

Tabel 1. Võimalike lahenduste võrdlus	22
---	----

1 Sissejuhatus

Paljud ettevõtted ja organisatsioonid kasutavad infotahvleid andmete jälgimiseks, analüüsimiseks ning kuvamiseks. Infotahvlid annavad kiire ülevaate organisatsiooni, osakonna või spetsiifilise protsessi olukorrast. Infotahvlid ühendavad kokku erinevad mõõdikud ja andmekogud ning kuvavad neid lihtsasti arusaadaval viisil. Infotahvli kasulikkus peitub selle võimes pakkuda ajakohast informatsiooni ning seeläbi aidata äriliste otsuste tegemisel. [1]

Käesoleva lõputöö raames luuakse rakendus, mis annab kokkuvõtva ülevaate OG Elektra AS lao tööpäevast ning seda iseloomustavatest mõõdikutest. Loodud rakendus võimaldab lao personalil paremini juhtida aja- ja inimressursi kasutust.

Töö koosneb teoreetilisest ja praktilisest osast. Teoreetilises osas analüüsitakse hulgilao töökorraldust, laohoone ülesehitust ning kasutusel olevat ettevõttespetsiifilist laohaldustarkvara (WMS). Lisaks uuritakse andmete visualiseerimise parimaid praktikaid ning olemasolevaid lahendusi.

Töö praktiline osa on infotahvli rakenduse arendamine. Tulemusena valmis interaktiivne infotahvli veebirakendus, mis annab kiire ülevaate laos toimuvast. Lõppkasutajate tagasiside põhjal selgus, et rakendus täidab oma eesmärgi ning pakub reaalselt väärtust lao töö koordineerimisel.

2 Taustinformatsioon

OG Elektra AS on 1991. aastal asutatud kodumaine ettevõte, mille põhitegevusalaks on jaemüük spetsialiseerimata kauplustes, kus on ülekaalus toidukaubad, joogid ja tubakatooted. [2] Kontserni tugevuseks on ainulaadne ahel tootjast tarbijani, mis seisneb seakasvatuses, omatoodangus, logistikakeskuses ning kaupluste ketis. Selline ahel tagab paindlikkuse ja stabiilsuse ning annab võimaluse tarbija soovidega kiiresti kohaneda. Kontsern on tugev just tänu sünergiale, mis tekib erinevate ettevõtete koosmõjust. Kontsernis töötab üle 1600 inimese. Kokku on ettevõttel üle Eesti 72 toidu- ja esmatarbekauplust ning 2 tööstuskaupade kauplust. [3]

2.1 Laohoone ja selle ülesehitus

OG Elektra AS-i üks viimastest suurematest investeringutest on olnud täiesti uue logistikakeskuse rajamine Tobia külla, kus asuvad kontserni tootmishooned ning peakontor. Uue laohoone (Lisa 2) suletud netopindala on 13504.1m² [4] ning kaupade ladustamiseks on hoone kahel korrusel 14075 alusekohta. Kuna kaupade vastuvõtt ja väljastamine toimub esimesel korrusel, on kahe korruse vahel lift.

Iga alusekohta tähistab unikaalne laoaadress (Lisa 2), nagu näiteks *PI-A-001-01/I*, kus

- *PI* tähistab lao esimest korrust
- *A* sektorit (riiulite rivi)
- *001* sektori osa (riiuli moodulit riiulite rivis)
- *01* riiuli korrust (enamasti on riiuleid viiel korrusel)
- */I* alusekohta sellel korrusel (enamasti on ühel korrusel kolm alusekohta).

Laoaadresse on kahte tüüpi: aktiivaadressid ja passiivaadressid. Aktiivaadressid asuvad enamasti esimesel korrusel ning need on esimeseks aadressiks, kuskohast komplekteerimisel antud kaupa võetakse. Passiivaadressidel ladustatakse sama kauba varusid. Passiivaadressilt kauba kättesaamiseks ning aktiivaadressi täitmiseks kasutatakse kahveltõstukit.

Laos olevad kauba väljastus- ning vastuvõtualad omavad samuti unikaalset laoaadressi. Neid on 139 ning need tähistavad kindalt betoonsammast lao 20 laadimismaja juures.

Igale kauplusele on infosüsteemis määratud väljastusala, kuhu tema tellitud ning komplekteeritud kaup enne väljastamist koondatakse. Samuti asuvad eelnimetatud aadressidel kaubad, mis on lattu saabunud ning ootamas ladustamist riulisse.

Igale tootele või tootegrupile on määratud kindlad aktiiv- ja passiivaadressid. Tänu sellele säilib kaupade lattu vastuvõtmisel ja ladustamisel optimaalne asendiplaan kaupade komplekteerimiseks, kus on oluline, et näiteks õrnad kaubad ei satuks esimesena alusele, kuhu kaupu komplekteeritakse. Lao suurest pindalast tulenevalt kasutatakse komplekteerimiseks seisuplatvormiga elektrilisi kahvelkärusid.

2.2 Töökorraldus laos

Kauplused esitavad kesklaole elektroonselt tellimusi kasutades ettevõttes kasutusel olevat hulgi- ja jaekaubanduse infosüsteemi. Tellimusi komplekteeritakse viiel päeval nädalas, esmaspäeviti ning laupäeviti tegeletakse muude, peamiselt administratiivsete töödega. Olenevalt päevast töötab laos ligikaudu 15-40 inimest. Lisaks komplekteerijatele töötavad laos ka kauba vastuvõtjad, andmespetsialistid, tõstukijuhid ning lao juhataja.

Komplekteerimise tööpäevad võivad olla väga pikad ning kesta kuni südaööni. Komplekteeritud kaubad laetakse veoautodele ning läkitatakse teele järgmise päeva varahommikul. Komplekteerimisel kasutatakse kahte erinevat meetodikat:

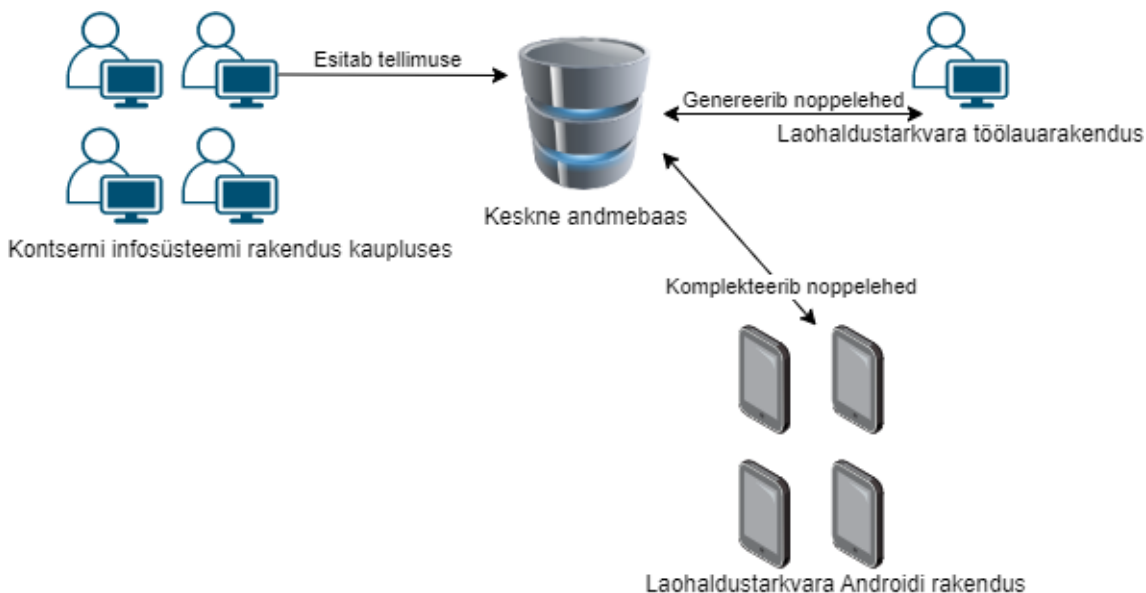
- objektipõhine komplekteerimine, kus korraga komplekteeritakse ühe kaupluse tellimus
- rühmkomplekteerimine, kus korraga komplekteeritakse mitme kaupluse tooteid mingist konkreetsest tootekategooriast.

Rühmkomplekteerimist kasutatakse näiteks tubakatoodete või maitseainete puhul, kus mahud on väikesed ning nendega tegelevad kindlad inimesed, kes neid kaupu väga hästi tunnevad. Lõplikult pakendatakse ühe kaupluse tellimus kokku sellele kauplusele määratud väljastusalas.

2.3 Olemasolev laohaldustarkvara

Uus laohoone on vanast kordades suurem ning erinevalt vanast laost käib töö uues laos elektrooniliselt. Kasutusel olev laohaldustarkvara pakett on rendatud spetsiaalselt uue

lao jaoks ning on liidestatud kontserni kaubanduse ja tootmise infosüsteemiga. Pakett koosneb Androidi rakendusest ning Windowsi töölauarakendusest. Mõlemad rakendused kasutavad sama andmebaasi. (Joonis 1)



Joonis 1. Infrastruktuuri skeem

Androidi rakendus sisaldab väga palju erinevat funktsionaalsust, olles töövahendiks nii komplekteerijatele, töstukijuhtidele kui ka kauba vastuvõtjatele. Rakenduse eri funktsioonide kasutamiseks peavad kasutajale olema määratud vastavad õigused.

2.3.1 Komplekteerimine

Komplekteerimise ettevalmistamine toimub töölauarakenduses, kus andmespetsialist töötleb kaupluste esitatud tellimused – korrigeerib tellitud kogused ning liidab tellimused kokku, kui üks kauplus peaks olema esitanud mitu tellimust.

Seejärel genereerib rakendus töödeldud tellimuste põhjal komplekteerimise noppelhed kummagi lao korruse jaoks, peale mida määratakse igale komplekteerijale hinnanguliselt poole tööpäeva jagu noppelhti – kõige suuremad noppelhed esmajärjekorras. Noppelhed, mida kohe hommikul ühelegi komplekteerijale ei määratud, määratakse komplekteerijale, kes esimesena lõpetab talle hommikul määratud tellimuste täitmise.

Komplekteerimiseks kasutatakse tahvelarvutit ning ribakoodilugejat. Ribakoodi lugejad fikseeritakse kasutaja randmele ja sõrmele selliselt, et mõlemad käelabad jäävad vabaks kaupade haaramiseks. (Lisa 2) Tahvelarvuti kinnitatakse kahvelkäru vastava hoidiku

külge või kantakse rihmaga kaelas. Peale Androidi rakendusse sisselogimist saab töötaja asuda komplekteerima talle määratud tellimusi. Tellimust täitma asudes soovib rakendus toodet, mida järgmisena komplekteerida, arvestades optimaalset teekonda läbi lao. Komplekteerimist võib alustada ka enda asukoha fikseerimisega, skaneerides mõne lähedalasuva riuliaadressi. Seejärel kuvatakse kasutajale järgmise toote aktiivaadresse, millel on positiivne jääk, noppimist vajavat kogust, toote nimetust, toote koodi, toote pilti ning hulgipakendi olemasolul hulgipakendi kogust. (Lisa 2) Kui aktiivaadressidel jääke pole, saab rakenduse kaudu teha „tõstuki kutse”, mis edastatakse parasjagu laos töötavatele tõstukijuhtidele rakenduse kaudu. Lisaks on graafiliselt esitatud aluse paiknemine riuli moodulis. Nopitud kauba kinnitamiseks ning valideerimiseks tuleb skaneerida selle ribakood või riuliaadress ning sisestada tegelik nopitud kogus. Kui noppel on maksimaalses võimalikus mahus komplekteeritud, viiakse kaup kauplusele määratud väljastusala laoaadressile ning märgitakse noppel täidetuks.

2.3.2 Sissetulnud kauba kontroll

Kauba saabumisel lattu on selle vastuvõtmise eelduseks saateleht ettevõtte infosüsteemis, millest enamasti saabub elektroonilisel kujul mõne EDI operaatori kaudu. Peale rakenduses saabunud kauba saatelehe valimist loob rakendus kauba kontrollimise akti, millel fikseeritakse kauba tegelik kogus. Kaupade lihtsamaks tuvastamiseks kasutatakse ribakoodilugejat. Kontrollitud kaubale trükitakse mobiilse printeriga etikett, millel on märgitud kauba nimetus, partii, säilivus ning kogus. Lisaks on etiketil *Data Matrix* kood, milles on kodeeritud samad andmed masinlugemise eesmärgil.



Joonis 2. Kontrollitud kauba etikett

Peale kauba kontrollimist kinnitatakse kontrollakt, fikseeritakse väljastusala aadress, kus kaup asub, ning suunatakse kaup ladustamisele.

2.3.3 Ladustamine

Ladustamist ootavate kaupade nimekiri ning asukohad on tõstukijuhile rakendusest näha. Ladustamise alustamiseks fikseerib tõstukijuht, millist kaupa ja millises koguses ladustama asub. Selleks skaneerib ta kaubale kleebitud etiketi, kasutades selleks spetsiaalset ribakoodilugejat, mis on võimeline ribakoode lugema mitme meetri kauguselt. Seejärel pakub rakendus laoaadresse - vastavalt toote kuuluvusele, kuhu kauba ladustama peaks. Peale kauba paigutamist riulisse kinnitatakse kauba ladustamine. Selleks tuleb rakenduses klõpsata laoaadressil või fikseerida laoaadress sihtides ribakoodilugeja riulil olevale kleepsule, millel on kodeeritud laoaadress. (Lisa 2)

2.3.4 Kauba siire

Kauba teisaldamiseks ühelt laoaadressilt teisele skaneeritakse kauba lähteaddress ning peale sihtaadressile paigutamist kinnitatakse sihtaaddress skaneerides riulil olev laoaadressi kleeps. Juhul kui rakenduse kasutajal pole füüsiliselt võimalik ise kaupa teisaldada, saab kauba siirde tõstukijuhile täitmiseks ette defineerida. Kauba ühelt aadressilt teisele teisaldamise põhjuseks võib olla näiteks kaupade paiknemise korrigeerimine.

2.3.5 Aktiivide täitmine

Tõstukijuhil on võimalik rakendusest vaadata täitmist vajavate aktiivaadresside nimekirja. Rakenduses kuvatakse väga väikese jäägiga või täiesti tühjad aktiivaadressid, millel oleval kaubal on passiivaadressil jääk olemas. See aitab ennetada olukordi, kus komplekteerijal ei ole tellitud kaupa võimalik noppida, sest kauba jääk ei asu talle kättesaadaval kõrgusel.

2.3.6 Inventuur

Rakenduses on ka inventuuri funktsionaalsus. Kasutajale kuvatakse valitud riulimooduli graafiline esitus ning igal alusekohal asuva kauba nimetus ning kogus. Kasutaja ülesanne on määrata iga alusekoha tegelik kogus.

3 Probleemi püstitus ja projekti eesmärk

Peatükis seatakse paika projekti eesmärgid ja meetoodika.

3.1 Ülesande püstitus

Selgitada välja kõige olulisemad lao tööpäeva iseloomustavad mõõdikud, kaardistada loodava rakenduse funktsionaalsed ning mittefunktsionaalsed nõuded. Luua rakendus, mis annaks kiire ja kokkuvõtva ülevaate lao tööpäevast ja seda iseloomustavatest mõõdikutest.

3.2 Lähtetingimused

Laos on kasutusel ettevõtte spetsiifiline laohaldustarkvara, millel puudub infotahvli funktsionaalsus, mis annaks kiire ülevaate lao tööpäevast. Töö käigus loodav rakendus on uus ja eraldiseisev programm, mis tuleb liidestada olemasoleva laohaldustarkvaraga ning kontserni infosüsteemiga. Rakenduse peamiseks kasutuskohaks saab laoruumi sissekäigu juures seinale kinnitatud kõrgeresolutsiooniline puuetundlik kuvar. (Lisa 2)

3.3 Eesmärk

Loodava rakenduse eesmärk on pakkuda lao personalile ajakohast informatsiooni ning aidata äriliste otsuste tegemisel. Rakendus aitab paremini juhtida aja- ning inimressursi kasutust, lihtsustab tööülesannete jagamist ning koostöö koordineerimist. Rakendus tõstab personali olukorratundlikust ning motiveerib neid rohkem panustama.

3.4 Meetoodika

Ülesande lahendamine toimub erinevates etappides:

- vajaduste ning nõuete kaardistamine
- sarnaste lahendustega tutvumine
- prototüübi kavandamine

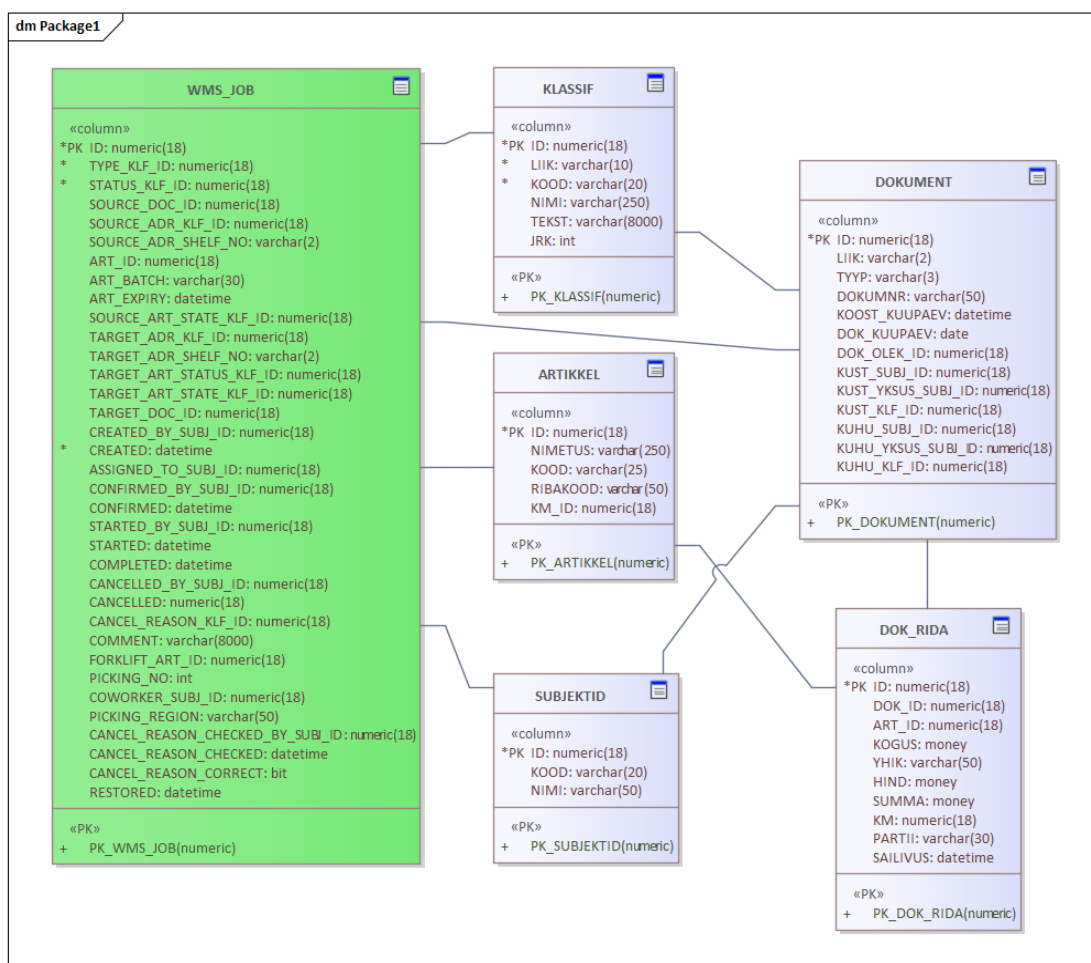
- prototüübi loomine
- sihtgrupilt tagasiside küsimine
- täienduste ning paranduste tegemine saadud tagasiside põhjal
- rakenduse juurutamine toodangu keskkonnas.

4 Probleemi analüüs

Peatükis analüüsitakse probleemi detailsemalt – kirjeldatakse laohaldustarkvara ning infosüsteemi andmemudelit, mille läbi selgitatakse välja, millised andmed on saadaval ning kuidas need võiksid kasulikud olla, fikseeritakse loodava lahenduse funktsionaalsed ja mittefunktsionaalsed nõuded ja kasutusjuhud ning võrreldakse analoogseid lahendusi.

4.1 Andmemudeli kirjeldus

Laohaldustarkvara andmemudeli põhiliseks olemiks on töökäsk, teisisõnu laokanne, mille peamiseks ülesandeks on kajastada kõiki kauba liikumisi laos. See viitab paljudele teistele olemitele kontserni infosüsteemis nagu näiteks artikkel (toode) ning subjekt. Subjekt võib olla füüsiline isik – lao töötaja, juriidiline isik – lattu kaupa tarninud hankija või üksus – kontserni üks kauplustest.



Joonis 3. Lihtsustatud andmebaasiskeem

Laokannete põhjal on teada igasugune kauba liikumine laoaadressi, partii, säilivuse ning kauba seisundi täpsusega. Riiulijääk on konkreetse laoaadressi sisse-välja kannete koguste summa. Lattu vastuvõetud kaubad pärinevad pseudoaadressidelt, mille üle arvestust ei peeta. Laost väljuvad kaubad liiguvad samuti pseudoaadressidele. Näiteks pärinevad lattu sissetulnud kaubad pseudoaadressilt nimega „P-Saateleht”. Lisaks on olemas ajatemplid laokande eri olekute ja nende olekutega seotud töötaja viide ning samuti viide dokumendile ettevõtte infosüsteemis, millega kaup lattu saabub või laost väljus.

Riiulite sektorid, sektorite osad, korrused ning lõplikud laoaadressid on defineeritud infosüsteemis klassifikaatoritena, millel on viide eelmise tasandi klassifikaatorile. See võimaldab esitada laoaadresside infot hierarhilisel kujul.

4.2 Lao tööd iseloomustavad mõõdikud ja nende olulisus

Asjaolu, et midagi saab mõõta, ei tähenda automaatselt, et seda peaks mõõtma. Mõõtmine maksab (antud juhul riistvaralist ressursi), seetõttu tuleb otsustada, mida mõõta, võttes arvesse selle mõõtmise oodatavat kasu. [3] Olulisemate mõõdikute leidmiseks jälgiti erinevaid lao tööprotsesse ning analüüsiti olemasolevat laohaldustarkvara. Lisaks küsiti lõppkasutajatelt, millist infot nad infotahvlil näha sooviksid.

Laohaldustarkvara põhiolemi, laokande, järgi saab leida väga palju kasulikku infot. Nagu näiteks

- Kui palju kaupu ootab ladustamist riiulisse ning kui palju on mingit sissetulekut või saatelehte juba riiulitesse ladustatud. Millise hankija kaup viimasena lattu tuli, kus see asub ning kes selle vastu võttis. See näitab olukorras, kus komplekteerimine ei ole võimalik kauba puuduse tõttu, kas lähiajal on oodata soovitud kaubale jäägi tekkimist.
- Kui palju on aktiivseid tõstuki kutseid ja kui palju aktiivaadresse vajavad lähiajal täitmist (aktiivaadressi jääk on alla mingi teatud koguse ning passiivaadressil on jääk olemas). Kui palju siirdeid on tegemata. Need aitavad otsustada kas suunata tööle rohkem tõstukijuhte või mitte.
- Kui palju on antud tööpäeval komplekteeritud objektipõhiseid ja rühmkomplekteerimise noppelheit nende kogumahust. Milliseid noppelheit ei ole

veel kellelegi määratud. See annab orienteeruva ülevaate tööpäeva lõppemise ajast ning aitab otsustada, kas suunata tööle rohkem komplekteerijaid.

- Kui palju on iga töötaja antud tööpäeval komplekteerinud (mitu laokannet täitnud), millal ta tööd alustas ning millal viimase noppe tegi. See aitab motiveerida ja annab ülevaate töötaja töökoormusest, aitab otsustada, kes võiks komplekteerida veel täitmata noppelehti.
- Kui palju on komplekteeritud konkreetset noppelehte ning kes seda tegi. Millised noppelehed on juba lõpuni komplekteeritud. Selle põhjal saab kontrollida, kas kõik tellitud kaubad on nopitud ning võib alustada väljastusalas aluste ettevalmistamist transpordiks.
- Kus keegi viimati asus ning mis ajahetkel see oli. See aitab leida suurelt pindalalt inimesi üles ja koordineerida koostööd. Lisaks annab viimase tegevuse ajatempel aimduse, kas inimene viibib laos või on läinud näiteks pausile.

4.3 Nõuded loodavale lahendusele

Peatükis tuuakse välja loodavale lahendusele püstitatud funktsionaalsed ja mittefunktsionaalsed nõuded. Peale olulisemate mõõdikute kaardistamist oli funktsionaalsete nõuete formuleerimine küllaltki kiire protsess, sest enamus nõudeid määratlevad, millist mõõdikut rakendus esitama peaks.

4.3.1 Funktsionaalsed nõuded

Loodavale rakendusele püstitati järgnevad funktsionaalsed nõuded:

- Rakendus peab kuvama riulisse ladustamist ootavate kaubaaluste kogust.
- Rakendus peab kuvama tööpäeva jooksul riulisse ladustatud kaubaridade kogust.
- Rakendus peab kuvama aktiivsete tõstukikutsete kogust.
- Rakendus peab kuvama tööpäeva jooksul täidetud tõstukikutsete kogust.
- Rakendus peab kuvama täitmist vajavate aktiivaadresside kogust.
- Rakendus peab kuvama tegemata kauba siirete kogust.
- Rakendus peab kuvama eraldi, kui palju noppeid on tehtud ja kui palju on neid veel teha ning kui palju on aega kulunud
 - I ja II korruse objektipõhises komplekteerimises
 - I korruse ja tubakatoodete rühmkomplekteerimises.

- Rakendus peab kuvama enim noppeid teinud töötajate järjestust, sh töötaja pilti, tema esimese ja viimase noppe aega ning antud päeval tehtud nopete koguarvu
- Rakendus peab kuvama teisi laotoiminguid (siirdamine, koguse parandamine, ladustamine) teinud töötajate järjestust, esimese ja viimase toimingu aega, viimase toimingu nimetust ning viimast asukohta
- Rakendus peab kuvama lattu viimati vastu võetud kaupa, sh kauba hankija nime, kontrollakti koostanud töötaja nime ja pilti, kauba asukohta ning kontrollakti koostamisele kulunud aega
- Rakendus peab kuvama sissetulnud kaupade ladustamised, kui palju ridu on juba ladustatud ning kui palju ridu on kokku, saabunud kauba saatelehe numbrit, hankija nime ning ladustamise algus- ja lõppaega
- Rakendus peab kuvama töötajatele määratud noppelehti kahanevas järjekorras nende täitmisprotsendi järgi, komplekteeriija nime ja pilti, tema viimast asukohta, tellija nime, komplekteerimise algus- ja lõppaega, nopitud ridade arvu, ridade arvu kokku
- Rakendus peab kuvama noppelehe numbrit ning selle ridade arvu, mida ei ole veel kellelegi täitmiseks määratud. I, II korruse kaupade ja tubakatoodete noppelehed peavad olema eristatavad. Kasutaja peab saama rakenduse kaudu noppelehe endale määrata. Vajalik on toimingu kinnitamine tahvelarvutis, laohaldustarkvara Androidi rakenduses. Kui määramata noppelehtede hulgas on prioriteediga noppelehti, peab olema lubatud ainult nende määramine kuni prioriteediga noppelehti nimekirjas enam ei ole.
- Rakendus peab kuvama täidetud noppelehti nende täitmisaja järgi kahanevalt, komplekteeriija nime ja pilti, tellija nime, komplekteerimise algus- ja lõppaega, viimase noppe laoaadressi
- Rakenduse kuvatavad andmed ja mõõdikud peavad uuenema iseseisvalt, ilma lõppkasutaja sisendita
- Rakendus peab kuvama I ja II korruse riulite põrandaplaani alusekoha täpsusega ja võimaldama põrandaplaani graafilisel esitusel andmete visualiseerimist.

4.3.2 Mittefunktsionaalsed nõuded

Loodavale rakendusele püstitati järgnevad mittefunktsionaalsed nõuded.

- Rakendus peab kasutama Microsoft SQL Server andmebaasi.

- Rakenduse kasutajaliides ning kuvatavad andmed peavad olema kergesti mõistetavad.
- Rakenduse konstantide ja keskkonnamuutujate nimetused peavad olema selgesti mõistetavad.

4.4 Olemasolevad lahendused

Infotahvlite ja andmete visualiseerimiseks on levinud näiteks Microsoft Power BI ja Tableau tarkvara kasutamine. Püstitatud funktsionaalsete nõuete kontekstis ei olnud nende kasutamine võimalik. Põhilisteks takistusteks oleks olnud noppelehtede määramise ja põrandaplaanide visualiseerimise funktsionaalsuse loomine. Esimene neist kätkeb rakendusloogikat, mille realiseerimiseks eelnimetatud tarkvarad mõeldud ei ole.

Tabel 1. Võimalike lahenduste võrdlus

	Komponendid andmete visualiseerimiseks (sektordiagrammid, edenemisribad)	Andmete automaatne värskendamine	Rakendus- ja äri loogika realiseerimine	Interaktiivse 2D või 3D graafika genereerimine
Microsoft Power BI	Jah	Jah	Ei	Ei
Tableau	Jah	Jah	Ei	Ei
Eraldi rakendus	Jah	Jah	Jah	Jah

Analoogseid erilahendusi uurides selgus asjaolu, et need ei ole reeglina eraldiseisvad tooted vaid üldjuhul mõne teise laohaldustarkvara (WMS) paketi osad ning tõenäoliselt oleks nende kohandamine antud ülesande spetsiifikale vastavaks olnud sama mahukas kui uue lahenduse arendamine. Olemasolevate lahenduste uurimine oli siiski kasulik, andes ettekujutuse ja ideid, milline loodav tulemus olla võiks.

5 Lahenduse realiseerimine

Töö autor otsustas ülesande lahendamisel veebirakenduse loomise kasuks, sest kasutajaliidese ehitamiseks kasutataval programmeerimiskeelel JavaScript on väga palju vabavaralitsentsiga teeke ja raamistikke andmete visualiseerimiseks ning graafika loomiseks. Lisaks on veebirakendusi lihtne uuendada ja kasutada erinevatel töökohtadel – puudub vajadus paigaldada rakendust erinevatele töökohtadele ning hoolitseda nende versiooniuuenduste pärast. Rakendusel on laialtlevinud arhitektuur, mis koosneb andmebaasist, serverirakendusest ning kasutajaliideseist.

5.1 Arendustöö käik

Esimene arendustööde etapp oli kasutajaliidese eskiiside loomine *Moqups* tarkvara abil. Eskiise väga detailselt lõpuni ei viimistletud, kuid need andsid hea alguspunkti kasutajaliidese ehitamiseks. Järgnes vajaminevate mõõdikute arvutamine ja serverirakenduse loomine, mis need mõõdikud loodavale kasutajaliidesele kättesaadavaks tegi. Seejärel algas rakenduse vaadete planeerimine tehnilise poole pealt – milliseid teeke ja komponente kasutada andmete paremaks visualiseerimiseks. Kuidas vaated struktureerida ja tagada komponentide soovitud paiknemine. Peale vaadete ehitamist olid viimasteks etappideks rakenduse testimine, dokumenteerimine ning kasutusjuhendite koostamine.

5.2 Kasutatud tehnoloogiad

Rakendus tervikuna moodustub Microsoft SQL Server andmebaasist, PHPs kirjutatud serverirakendusest ning kasutajaliideseist, mille loomisel on kasutatud HTMLi, CSSi ja JavaScripti.

Andmebaasi valiku dikteeris asjaolu, et loodud rakendus tuli liidestada olemasoleva laohaldustarkvaraga ja kontserni infosüsteemiga – need juba kasutavad Microsoft SQL Server andmebaasi.

Serverirakenduse loomisel otsustati PHP kasuks peamiselt selle tõttu, et kontsernis on varem loodud sellega erinevaid ettevõtte siseseid tööriistu. Sellest tulenevalt on ettevõtte IT osakonnal pädevus selliste rakenduste käitamisel ning tööshoidmisel. Rakenduse

serveerimiseks loodi uus Linuxi virtuaalmasin, millele paigaldati Apache veebiserver. Veebiserver on ligipääsetav vaid ettevõtte sisevõrgust. Olgugi, et PHP turuosa on viimasel ajal languses, on see jätkuvalt enimlevinud keel serverirakenduste loomiseks [5] ning sellel palju häid omadusi, nagu näiteks vabavaraline litsents, operatsioonisüsteemi agnostilisus, väga suur kogukond ja põhjalik dokumentatsioon. Tulenevalt antud serverirakenduse madalast keerukusest ning mahust ei kasutatud serverirakenduse arenduseks täiendavaid raamistikke või teeke.

Kasutajaliidese loomisel on kasutatud erinevaid vabavaralisi teeke ja komponente nagu näiteks Bootstrap 4 CSS raamistik, millest pärineb enamik kasutajaliidese elemente: hüpikaknad, nupud ja edenemisribad. Lisaks on kasutajaliidese erinevate komponentide paiknemine määratletud Bootstrapi *grid* süsteemi abil. JavaScripti teekidest on kasutatud sektordiagrammide loomiseks Chart.js teeki, pörandaplaanide visualiseerimiseks three.js teeki, kuupäevade vormindamiseks ja ajavahede arvutamiseks moment.js teeki ning lihtsamaks DOMi manipuleerimiseks jQuery teeki. jQuery kasutamise tingis osaliselt ka teiste teekide sõltuvus sellest.

5.3 Mõõdikute arvutamine

Rakenduse kuvatavad mõõdikud arvutatakse välja peamiselt laokande olemi põhjal. Arvutused teostatakse andmebaasi tasemel, sest andmebaasi serveriarvuti on riistvaraliselt väga võimekas ning SQL päringukeel optimaalne suurte andmehulkade töötamiseks ning agregeerimiseks. Kirjutatud päringud on defineeritud andmebaasis *table-valued* funktsioonidena. Funktsioonide kasutamine võimaldas arenduse ajal päringuid muuta ilma serverirakenduse koodi muutmata. Lisaks vähendavad funktsioonid päringu tegemisele kulunud aega – päringute täitmisplaane taaskasutatakse, salvestades need peale esmast koostamist andmebaasiserveri mällu. [5] Enamikes päringutes on kasutatud CTE avaldisi (Lisa 4), mis on paremini loetavad ja võivad mõningatel juhtudel pakkuda paremat päringu kiirust. Vastupidiselt alampäringule saab CTE avaldise tulemusi pärida mitmeid kordi ilma seda uuesti defineerimata. [6]

Pörandaplaanide visualiseerimiseks vajaminevad laoaadresside andmed pärinevad samuti andmebaasist – nende leidmiseks loodud päringud sisaldavad alampäringuid ning tulemuseks on hierarhilised andmed. Tulenevalt Microsofti SQL Server andmebaaside JSON andmevahetusvormingu toest on päringute tulemuseks olevad andmed

konverteeritud JSON formaati juba andmebaasi tasemel. Selline lähenemine võimaldas minimeerida andmebaasi ning serverirakenduse vahelist võrguliiklust. Microsoft SQL Server omab ka XML andmevahetusvormingu tuge, kuid kasutajaliidese ehitamisel kasutatud JavaScript programmeerimiskeelel ei ole sisseehitatud tuge XML vormingu töötlemiseks.

5.4 Serverirakendus mõõdikute publitseerimiseks

Loodud serverirakenduse maht ja keerukus on väike, sest selle peamiseks ülesandeks on kasutajaliidese tehtud HTTP päringute vastuvõtmine, nende töötlemine selgitamaks välja, millist mõõdikut päritakse, mõõdikule vastava andmebaasipäringu tegemine ning tulemuse kasutajaliidesele tagasisaatmine. Lisaks mõõdikute ja laoaadresside pärimiseks kasutatavate GET meetodi otspunktidele on realiseeritud üksainus POST meetodi otspunkt, mille kaudu lisatakse andmebaasi noppellehe haaramise kirje, mis hiljem tahvelarvutis kinnitatakse ja seeläbi noppelleht kasutajale täitmiseks määratakse. Tulenevalt asjaolust, et rakenduse serverimiseks kasutatud veebiserver on konfigureeritud ligipääsetavaks vaid kontserni sisevõrgust ei olnud vajalik ka juurdepääsupiirangute implementeerimine.

5.5 Kasutajaliidese lahendus

Kasutajaliidesele on kolm vaadet: koondvaade (Lisa 3), I korruse põrandaplaan ning II korruse põrandaplaan.

5.5.1 Koondvaade

Koondvaade koosneb päisest ja kuuest tulpast. Päises asuvad tööpäeva üldpilti iseloomustavad mõõdikud ning tulpades erinevad järjestatud andmed. Suure hulga andmete puhul on kõik tulbad eraldi üles-alla keritavad. Peale esmast renderdamist uuendatakse kuvatavaid andmeid ilma kasutaja sisendita iga 40-90 sekundi tagant, olenevalt andmete eripärast ning nende pärimise kulukusest. Uuendamise intervallid on defineeritud programmikoodis konstantidena. Iseseisev uuendamine on tagatud rekursiooniga, kus andmeid renderdav või uuendav (olenevalt kas on tegu esmase lehe laadimisega või mitte) funktsioon kutsub ennast ise välja konstandina defineeritud aja möödumisel kasutades JavaScripti `setTimeout` funktsiooni. Uuendamisel otsib funktsioonis olev algoritm DOMist üles serverirakenduselt päritud andmeplokile vastava

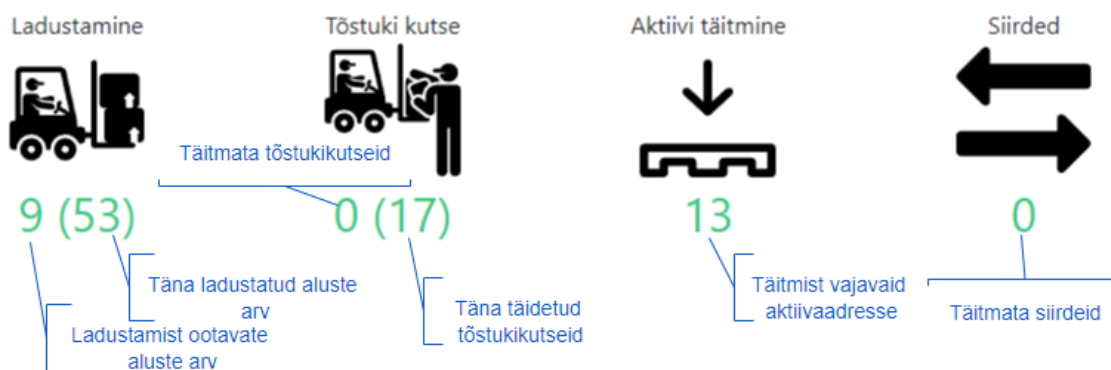
HTML komponendi ning uuendab selle komponendi poolt kuvatavaid andmeid. Tulpades kuvatavate andmete korrektne järjestus on tagatud CSS *order* atribuudiga. Vastupidiselt kogu vaate uuesti laadimisele tagab selline uuendamine katkematu kasutajakogemuse ja võimaldab sujuvate animatsioonidega anda märku andmete muutumisest.

Koondvaates kuvatakse ka andmetega seotud töötaja pilti selle olemasolul. Pildi saavad töötajad üles laadida laohaldustarkvara Androidi rakenduses, kasutades selleks tahvelarvuti esikaamerat. Pildi üleslaadimine ei ole kohustuslik. Pildi ümber oleva ümara raami värvus tähistab töötaja aktiivsust. Kui viimasest tegevusest on möödas rohkem kui 10 või 15 (olenevalt tegevuse tüübist) minutit, on raam oranži värvi, tähistamaks, et töötaja võib olla pausil või laost lahkunud. Pildi juures on kuvatud minutites möödunud aeg viimase toimingu tegemisest.

Lisaks kuvatakse koondvaates kuupäeva- ja kellaega ning otseteed infotahvli ja laohaldustarkvara veebipõhiste juhenditele.

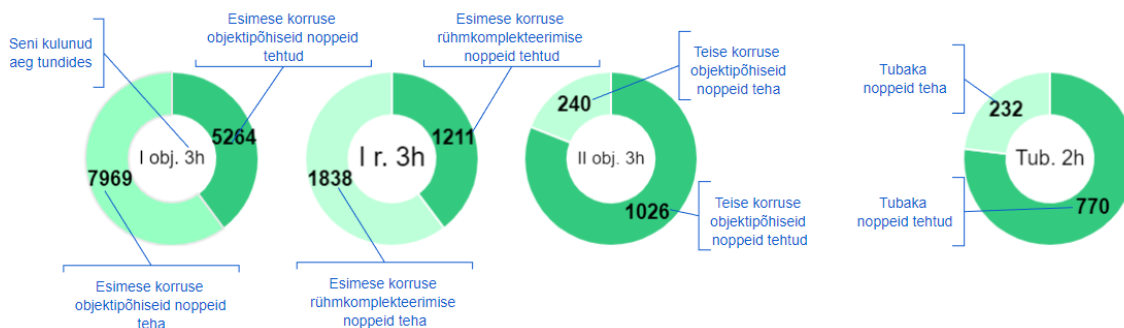
5.5.1.1 Koondvaatel kuvatavate andmete kirjeldused

Koondvaate päises kuvatakse lao tööd iseloomustavaid koondandmeid.



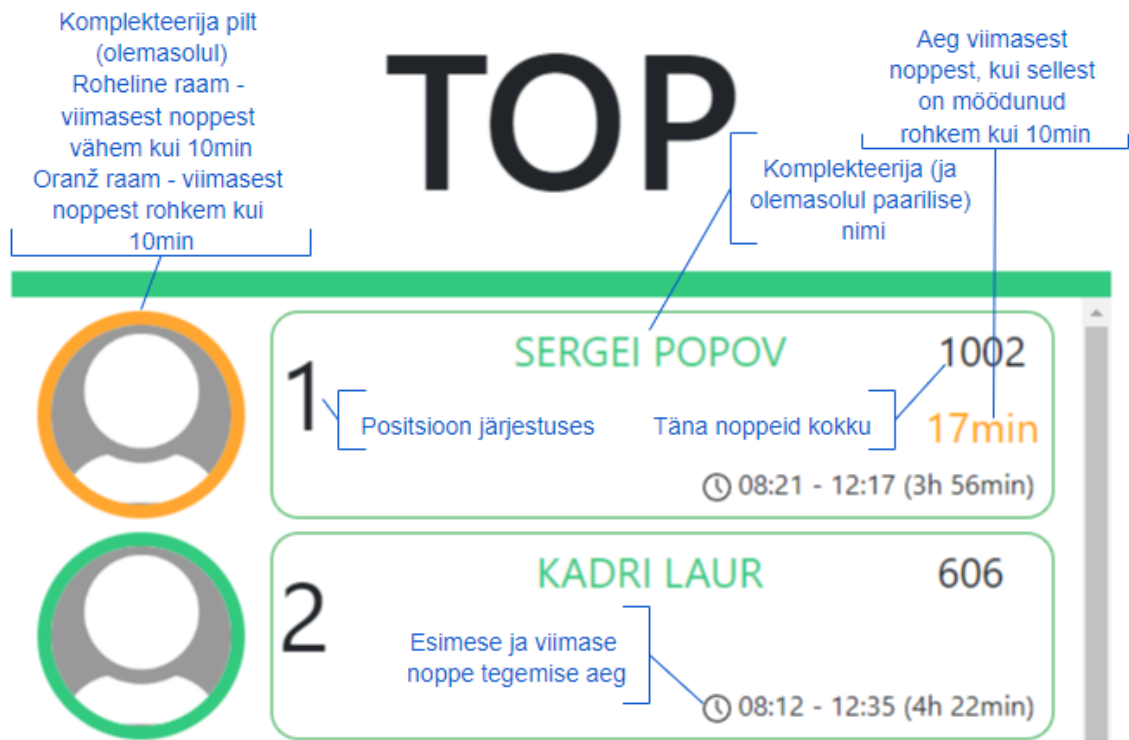
Joonis 4. Infokuvade selgitused - ikoonid koondvaate päises

Kui konkreetse sektordiagrammi kuvatav info antud tööpäeval puudub, näiteks kui antud päeval ei tehta üldse rühmkomplekteerimist, siis ei näidata vastavat sektordiagrammi üldse.



Joonis 5. Infokuvade selgitused - sektordiagrammid koondvaate päises

Koondvaate esimeses tulbas kuvatakse päeva kõik komplekteerijad nopete arvu põhjal järjestatuna. Kui kahel komplekteerijal on samapalju noppeid siis asetatakse ettepoole see kellel on kulunud vähem aega. Komplekteerija peale vajutades näidatakse noppelehtede tulpades (4. ning 6. tulp) ainult selle komplekteerija noppelehti. Uuesti sama komplekteerija peal vajutades filtreerimine lõpetatakse. Filtreerimine lõpetatakse ka automaatselt 90 sekundi möödumisel.



Joonis 6. Infokuvade selgitused - komplekteerijate järjestus

Teises tulbas kuvatakse kõik kauba vastuvõtmised lattu antud päeval koostatud kontrollaktide põhjal.



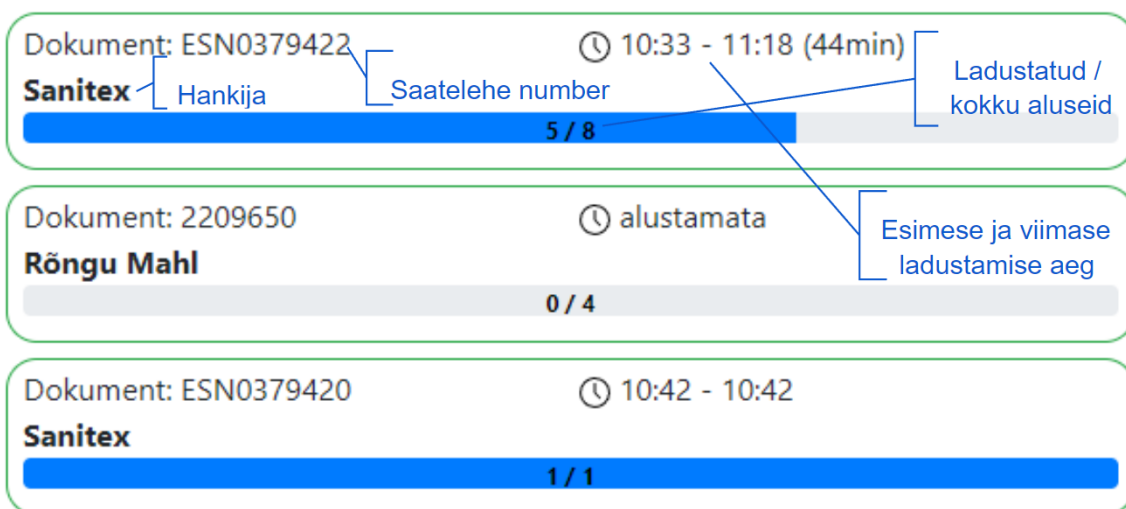
Joonis 7. Infokuvade selgitused - kauba vastuvõetud

Kolmandas tulbas kuvatakse töötajad, kes on päeva jooksul teinud muid laotoiminguid nagu näiteks siirdeid, ladustamist või koguse parandust. Tulemused on järjestatud toimingute koguarvu põhjal.



Joonis 8. Infokuvade selgitused - muud laotoimingud

Nende järel kuvatakse kolmandas tulbas ka sissetulnud kauba ladustamise edenemised saatelehtede kaupa. Lõpuni ladustatud kaubad on kõige all ning lõpule kõige lähemal üleval.



Joonis 9. Infokuvade selgitused - kauba ladustamised

Neljandas tulbas kuvatakse poolleiolevaid ning alustamata noppelehti, mis on määratud kindlale töötajale täitmiseks. Noppelehed on sorteeritud täitmisprotsendi järgi kahanevalt. Kui noppeleht saab täidetud, liigub ta viimasesse, täidetud noppelehtede tulpa.

Komplekteerija pilt (olemasolul)

Roheline raam - viimasest noppest vähem kui 10min

Oranž raam - viimasest noppest rohkem kui 10min

Tellija nimi või rühmanoppe puhul kaupluste arv

Komplekteerija nimi

Viimase noppe laoaadress

Nopitud / ridu kokku

Aeg viimasest noppest, kui sellest on möödunud rohkem kui 10min

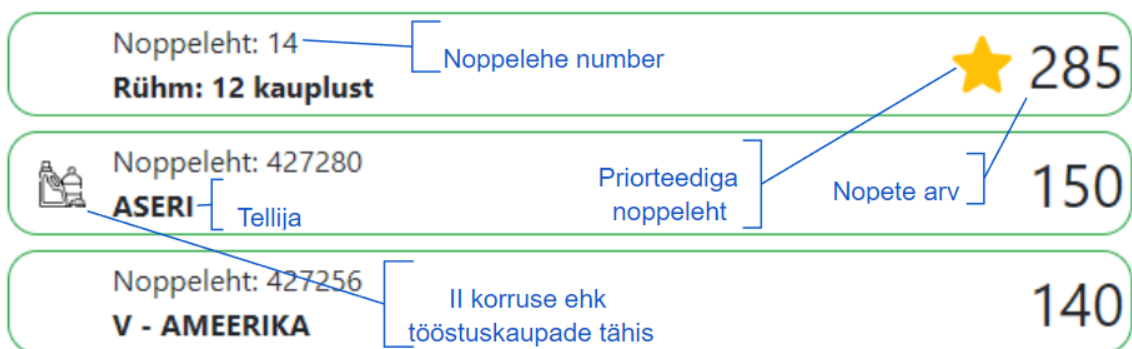
II korruse ehk tööstuskaupade tähis

Noppelehe esimese ja viimase noppe aeg

Employee Name	Group / Address	Count	Time
TAMARA LÕHMUS	Rühm: 15 kauplust Viimane asukoht: QX-003-02	296 / 330	08:15 - 10:59 (2h 44min)
NIKOLAI SILD	JÄRVEOTSA Viimane asukoht: OX-013-01	317 / 389	08:02 - 11:00 (2h 57min)
ENE PÕDER	JÄRVAKANDI Viimane asukoht: LT-005-01	42 / 52	09:59 - 10:25 (25min)
ALEXANDER MOROZOV	ENDLA Viimane asukoht: LT-007-01	37 / 49	10:32 - 11:00 (27min)

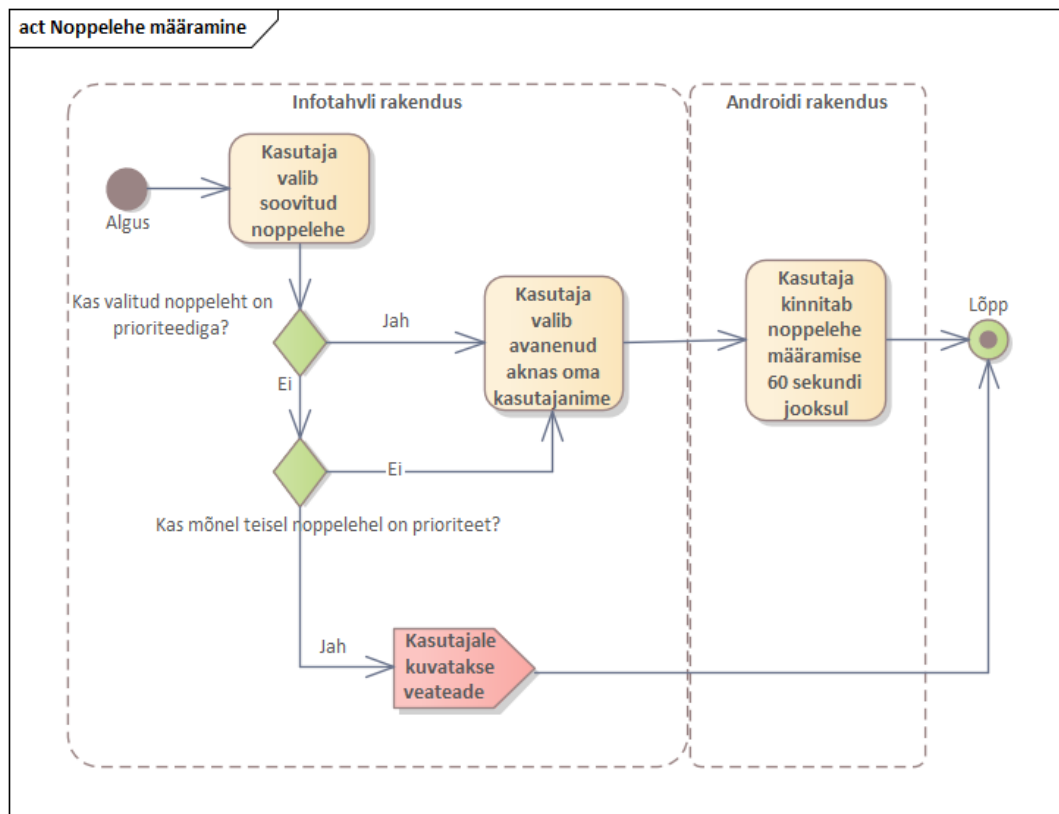
Joonis 10. Infokuvade selgitused - poolleiolevad ja alustamata noppelehed

Viendas tulbas kuvatakse noppelehed, mida ei ole veel kellelegi täitmiseks määratud. Noppelehed on sorteeritud prioriteedi ning ridade arvu järgi kahanevalt.



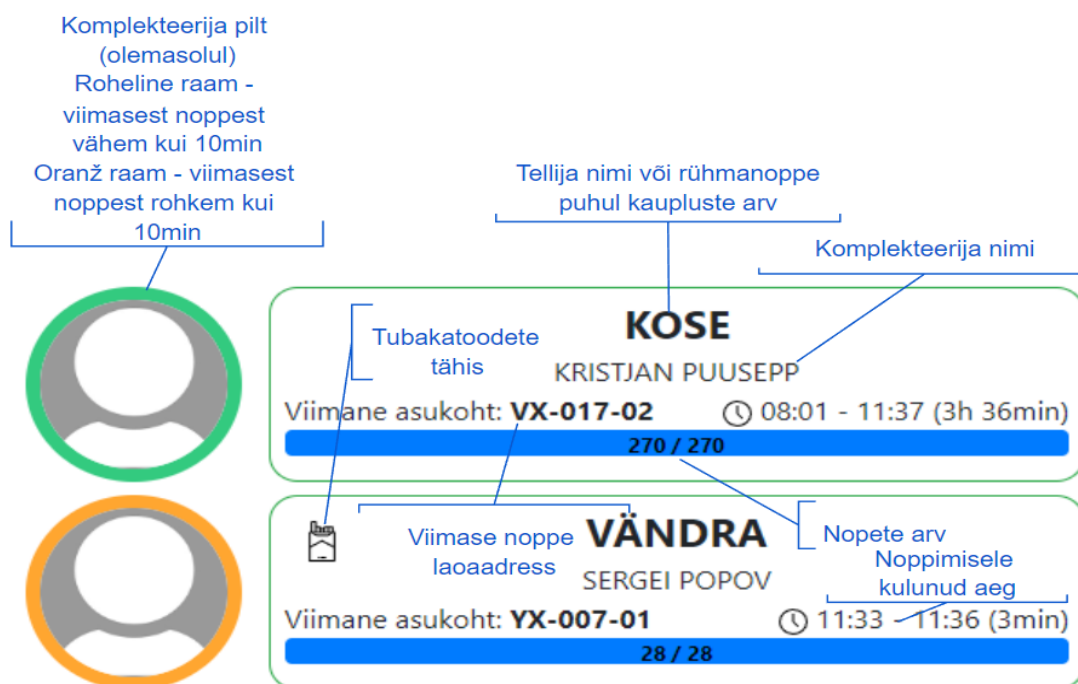
Joonis 11. Infokuvade selgitused - määramata noppelehed

Kui eelnevalt toimus noppelehtde määramine ainult laohaldustarkvara töölauearanduses, siis infotahvlilt saab noppelehele vajutades kasutaja selle ise endale täitmiseks määrata. Selleks tuleb avanenud hüppikaknas valida enda kasutajanimi ning vajutada "Valmis". Lisaks tuleb määramine tahvelarvutis 60 sekundi jooksul kinnitada. See välistab olukorra, kus keegi ekslikult määrab noppelehe kellelegi teisele. Kui määramist ei kinnitata, siis ilmub noppeleht paari minuti jooksul nimekirja tagasi. Kõrge prioriteediga noppelehe olemasolul teisi noppelehti valida ei saa, kuni kõik prioriteetsed noppelehed on kellelegi määratud.



Joonis 12. Noppelehe määramise tegevusdiagramm

Viimases koondvaate tulbas kuvatakse täidetud noppelehti viimase noppe tegemise aja järgi kahanevalt.



Joonis 13. Infokuvade selgitused - täidetud noppelehed

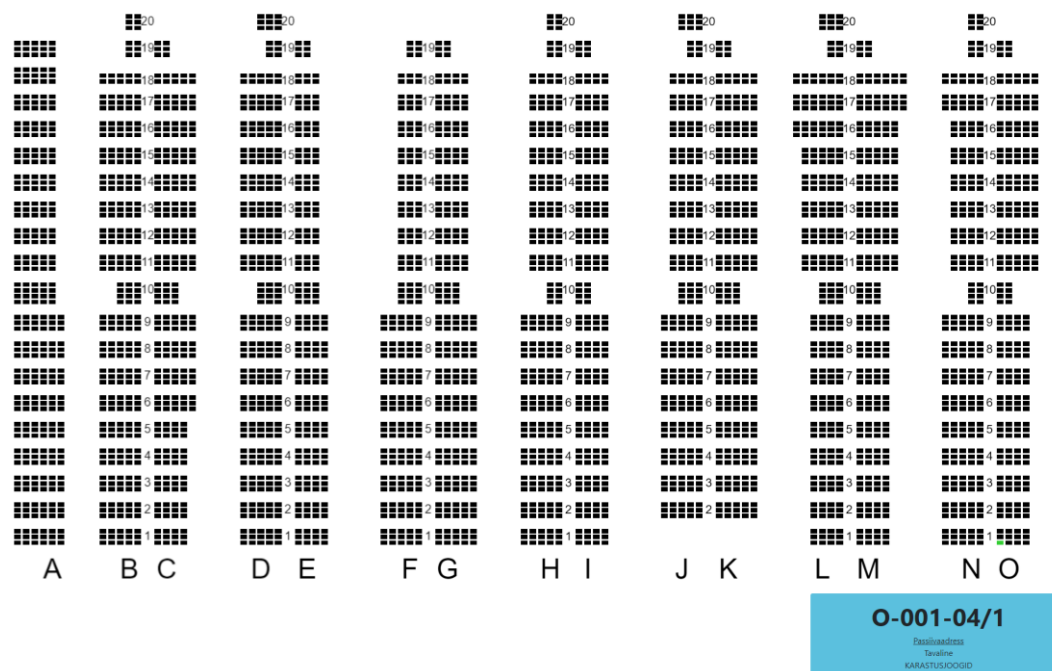
5.5.2 Põrandaplaanide visualiseerimine

Põrandaplaanide visualiseerimiseks on kasutatud JavaScripti teeki three.js. Three.js on teek, mida kasutatakse animeeritud 3D-arvutigraafika loomiseks ja kuvamiseks veebibrauserites. Three.js aluseks on JavaScripti WebGL rakendusliides, mille käitamiseks kasutatakse arvuti graafikakaarti. [7] Millegi kujutamiseks on vaja defineerida kolm objekti: stseen, kaamera ning renderdaja. Stseenile saab lisada erinevaid geomeetrilisi kujundeid nagu näiteks kuubikuid, keraid ja silindreid. Kujundeid on võimalik defineerida ka ise, kirjeldades nende kuju erinevate joonte ja parameetrite abil. Kujunditele saab määrata nende väljanägemise, kasutades tekstuuri- ja materjaliobjekte. Renderdaja joonistab stseeni HTML DOMis *canvas* elemendile ning kuvab seda kaamera objekti loomisel defineeritud kauguse ja nurga alt. [8]

Antud lahenduses on laoaadressid visualiseeritud kahemõõtmelisena, olgugi et tehniliselt on tegu kolmemõõtmelise stseeniga. Kahemõõtmelisus on tagatud sellega, et kaamera polaarnurk vaadeldavate objektide suhtes on fikseeritud 0 radiaanile ning selle muutmine keelatud. Kaamera asimuut vaadeldavate objektide suhtes on samuti fikseeritud – vaate „pööramine” ei annaks mingit mõõdetavat kasu. Kahemõõtmelisus tagab, et kõiki laoaadresse on võimalik korraga näha, tagades seeläbi parema ülevaate kogupildist ja võimalikest tendentsidest. Kaamera paiknemist x ja y teljel saab muuta ekraanil lohistades. Asukohta z teljel muudetakse kas hiire rullikuga või kahe sõrmega sisse ja välja suumides.

Iga alusekoht laos on kujutatud ristkülikuna, millel vajutades värvitakse see roheliseks ning kuvatakse selle laoaadress, aadressi tüüp ning aadressile määratud tooted või tootegrupid. Kuna tegu on kahemõõtmelise joonisega, on riiulite eri korrused kujutatud kas vasakult-paremale või vastupidi, olenevalt sellest, kummalt poolt riiul füüsiliselt ligipääsetav on. Joonisel on kujutatud sektorite tähised nende alguses ning sektoriosade numbrid. Lisaks on võimalik joonisel visualiseerida erinevaid andmeid, nagu näiteks tühjade aadresside asukohad või konkreetse toote asukoht laos. Visualiseeritavate andmete tüübid ja neile vastavad päringud on defineeritud andmebaasi protseduurides. Uute andmetüüpide ja päringute lisamiseks ei pea serverirakenduse või kasutajaliidese koodi muutma – piisab andmebaasi kirjetest ja nende vastavate päringute loomisest.

Päringu tulemuses peab olema nimekiri laoadressidest ning värvikood HEX süsteemis, millega see tähistada.



Joonis 14. I korruse põrandaplaan

Põrandaplaani visualiseerimise algoritmi sisendiks on sektorite, sektoriosade, korruste ning alusekohtade hierarhiline andmestik JSON formaadis. See andmestik ei sisalda alusekohtade koordinaate või muud nende füüsilist paiknemist kirjeldavat infot. Seetõttu on alusekohtade paiknemise tavapärased defineeritud algoritmis endas. Näiteks vahed erinevate sektoriosade vahel ja fakt, et iga paarisarvulise järjekorranumbriga sektori järel on järgmine sektor ning mitte vahekäik. Need eripärad ja paiknemisreeglid tuli lao kahe eri korruse kohta autoril ise kokku koguda. Peamiselt tähendas see laos ringikõndimist ja märkmete tegemist ning ehitusjooniste analüüsi. Sellise lahenduse kasuks otsustati, sest laos olevate riulite füüsilise paiknemise muutumine on vähetõenäoline.

5.5.3 Lahenduse testimine

Laohaldustarkvara arenduse ajaks oli ülesse seatud paralleelne andmebaas arenduse ja testimise jaoks. Testimise andmebaas oli toodangu andmebaasi mõne päeva vanune koopia, kuid kuna seal andmebaasis reaajas tööd ei tehtud, tuli rakenduse vaadete uuenemist testida kasutades toodangu andmebaasi, mille andmetes toimub tööpäeva jooksul pidevalt muutusi. Ülejäänud funktsionaalsus sai enne rakenduse toodangu keskkonda paigutamist testitud selleks ettenähtud andmebaasi kasutades.

5.6 Kasutajate tagasiside ja võimalikud edasiarendused

Infotahvel võeti kasutajate poolt hästi vastu. Kasutajad olid juba eelnevalt hästi meelestatud ja vastuvõtlikud uutele asjadele ja lahendustele, sest antud lahenduse kasutusele võtmine toimus vahetult peale uue laohaldustarkvara (WMS) paketi üldist juurutamist. Peale esmase versiooni avalikustamist sai kasutajate tagasiside põhjal parandatud mitmed väiksemad vead, nagu näiteks asjaolu, et infotahvel ei tohiks tööpäeva lõppenuks lugeda kuupäeva muutumisel, sest üksikutel juhtudel jätkub töö laos ka paaril tunnil peale südaööd. Lisaks vigade avastamisele andsid kasutajad ka tagasisidet, kuidas rakendust paremaks muuta – näiteks esimeses tulbal komplekteerijal vajutades toimuv filtreerimine oli ühe lõppkasutaja idee.

Peale lahenduse loomist selgus, et infotahvli soovitakse harvadel juhtudel kasutada ka mujal - näiteks lao juhataja kabinetis olevas lauaarvutis. Rakendus on arendatud 3840 x 2160 pikslite arvuga kuvareid silmas pidades. Väiksema eraldusvõimega kuvaritel on mõned kasutajaliidese elemendid proportsioonidest väljas, sest kohati on kasutatud elementide suuruste ja paiknemise kirjeldamisel staatilisi väärtusi. Seega oleks üks võimalik edasiarendus rakenduse vaadete optimeerimine ka väiksematele kuvaritele.

5.7 Majandusliku kasu analüüs

Enne infotahvli juurutamist sai noppellehti komplekteerijale täitmiseks määrata vaid lao andmespetsialist laohaldustarkvara töölaarakendusest. Komplekteerijal tuli endast märku andmiseks minna andmespetsialisti ruumidesse, kuhu võisid tekkida järjekorrad. Infotahvli rakenduses realiseeritud noppellehtede määramise funktsionaalsus aitas nende järjekordade tekkimist vähendada, pakkudes alternatiivset moodust tööülesannete jagamiseks.

Kokkuvõte

Käesoleva lõputöö eesmärk oli luua interaktiivne infotahvli rakendus, mis annaks kiire ülevaate lao tööpäevast ning seda iseloomustavatest mõõdikutest, loodud rakendus on abivahend lao tööprotsesside juhtimiseks ning koordineerimiseks. Rakendus vastab töös püstitatud funktsionaalsetele ning mittefunktsionaalsetele nõuetele.

Eesmärgi saavutamiseks analüüsiti laohoone ülesehitust, töökorraldust laos ning laos kasutusel olevat ettevõttespetsiifilist laohaldustarkvara, millega loodud rakendus liidestati. Seejärel selgitati välja lao tööd iseloomustavad mõõdikud ning nende olulisus ja kasulikkus. Rakenduse arendamiseks ja käitamiseks kasutati valdavalt väga laialt levinud ning ennast tõestanud tehnoloogiaid ja raamistikke. Osaliselt oli tehnoloogia valik dikteeritud kontsernis olemasoleva infrastruktuuri poolt.

Töö praktiline osa hõlmas andmebaasipäringute kirjutamist ning optimeerimist, serverirakenduse ja kasutajaliidese loomist, lahenduse testimist ning juhendite loomist. Praktilise poole pealt oli suurimaks ajakuluks ning õppetunniks lao pörandaplaanide visualiseerimine.

Peale lahenduse valmimist lansseeriti see toodangu keskkonnas. Kasutajad võtsid rakenduse hästi vastu ning andsid häid ideid kuidas rakendust paremaks muuta. Tagasiside põhjal tehti muudatusi ning parandusi ja rakenduse lõplik lahendus on täitnud seatud eesmärged ning töötanud ilma märkimisväärsete tõrgeteta juba mõnda aega. Peale lõpliku lahenduse valmimist koostati rakenduse kasutusjuhendid, mis on lihtsasti kättesaadavad rakenduse enda kaudu. Infotahvel on tihti esimene asi, mis haarab inimese tähelepanu, kes antud ladu varem külastanud ei ole.

Kasutatud kirjandus

- [1] „What is a data dashboard?“, Microsoft, [Võrgumaterjal]. Available: <https://powerbi.microsoft.com/en-us/data-dashboards/>.
- [2] „Ehitisregister“, [Võrgumaterjal]. Available: <https://ehr.ee/>.
- [3] A. Janes, A. Sillitti ja G. Succi, „Effective Dashboard Design“, *Cutter IT Journal*, 2013.
- [4] „World Wide Web Technology Surveys“, W3Techs, [Võrgumaterjal]. Available: <https://w3techs.com/>.
- [5] „User-defined functions“, Microsoft, [Võrgumaterjal]. Available: <https://learn.microsoft.com/en-us/sql/relational-databases/user-defined-functions>.
- [6] D. Sarka, M. Radivojevic ja W. Durkin, *SQL Server 2016 Developer's Guide*, Packt Publishing Ltd, 2017.
- [7] J. Dirksen, *Learning Three.js: the JavaScript 3D library for WebGL*, Packt Publishing Ltd, 2013.
- [8] „Three.js documentation“, [Võrgumaterjal]. Available: <https://threejs.org/docs/>.
- [9] „Ettevõtte tutvustus“, OG Elektra AS, [Võrgumaterjal]. Available: <http://grossitoidukaubad.ee/ettevottest/>.
- [10] „E-äriregister“, [Võrgumaterjal]. Available: <https://ariregister.rik.ee/>.

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks¹

Mina, Karl-Jaagup Rõngas

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

30.11.2022

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

Lisa 2 – Fotod



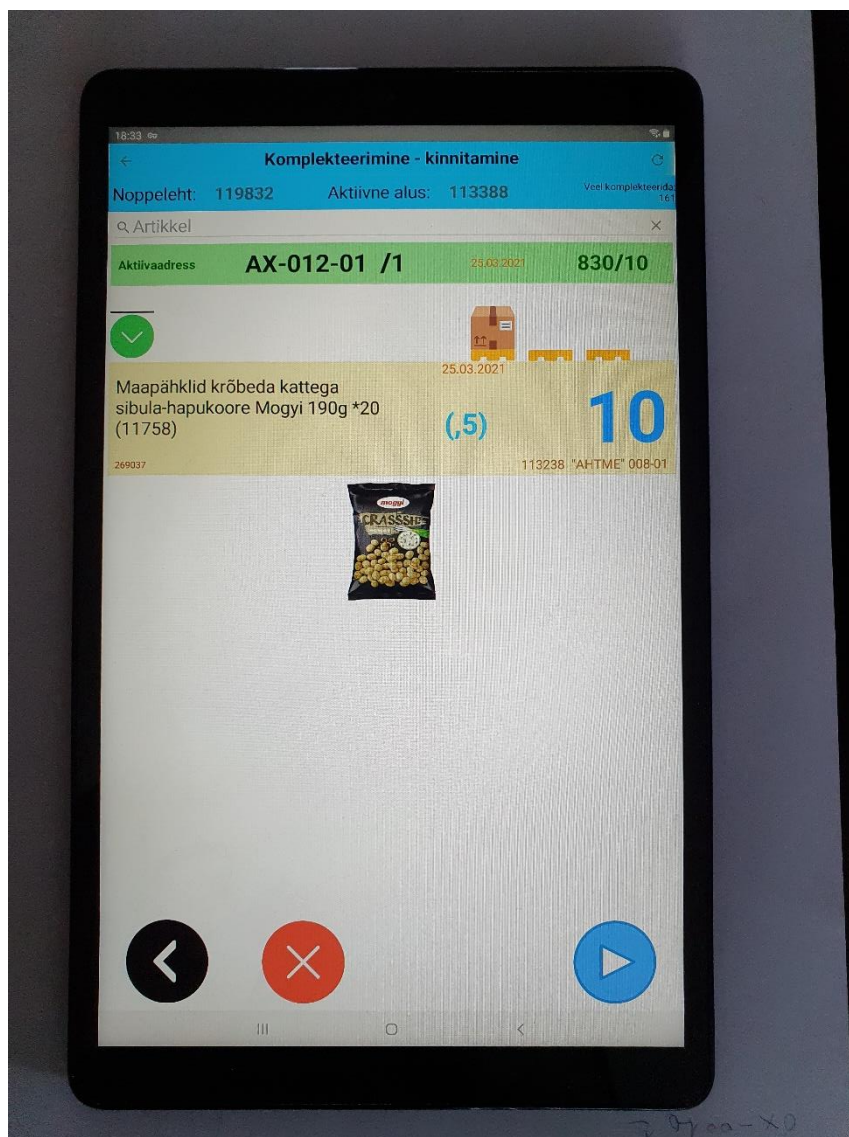
Uus laohoone



Laoaadress



Ribakoodilugeja



Komplekteerimise vaade Androidi rakenduses



Infotahvli peamine kasutuskoht

Lisa 3 – Infotahvli koondvaade

TOP		Lokuminte		Hoidulokuminte		Alles sõbraste		Juhendid		9:43:27	
		18 (56)		1 (8)		2		Laopäev I		01.12.2022	
1	SERGEI FOROV	282	Forans Eesti	4000000000	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000
2	LAURA LIIV	269	Mobex	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000
3	PETER KULL	265	Sanitec	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000
4	LIDIMILA STEPANOV	248	Sanitec	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000
5	KADRI LAUR	221	Sanitec	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000
6	NIKOLAI SILD	215	Kroonpuu	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000
7	VIKTOR RIIJAN	214	Coormlander	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000
8	SANDER SARAFUJ	198									
9	REIN SOKOLOV	190									
10	TAMARA LÖHMUS	184									
11	KRISTIAN PLUSEPP	179									
12	MEELIS MIICHAJEV	163									
13	TOOMAS OTS	151									
14	INDREK JAKOVLEV	140									
15	PRIT PAJU	125									
	ARTUR ANDREJEV	119									
	TIINA VASSILJEV										
	ÜLE KLISK										
	MARIA REBANE										
	SIRJE OJA										
	KRISTINA KARU										
	MARGUS LEPPIK										
	SVETLANA MÄGI										
	MAJAKA										
	JÕGEVA PARGI										
	KADRINA										
	JÄRVA-JAANI										
	TURU KAUBAMAJA										
	GEA										
	NARVA 2 PÄHELMÄE										
	MUSTAMÄE										
	RANDVEE										
	V - MAARJA										
	V - AMERIKA										
	TAPA KAUBANDUSKESKUS										
	Rühm: 20 kauplust										
	RAJA										
	AHTME										
	Rühm: 17 kauplust										
	KALEV KOHTLA-JÄRVE										
	KUNDA										
	PILLE-RIIN										
	SAUE										
	MUSTVEE										
	KADRINA										
	KELA										
	GEA										
	JÄRVA-JAANI										
	EVE										
	KADRINA										
	KUNDA										
	PILLE-RIIN										
	TURU KAUBAMAJA										
	TAPA KAUBANDUSKESKUS										
	V - MAARJA										

Lisa 4 – Päringu näide

```
WITH TANA AS
(
    SELECT
        WJ.[STARTED],
        S.ID AS SUBJ_ID,
        S.NIMI AS SUBJ_NAME,
        HANKIJA.NIMI AS SUPPLIER,
        K_SOURCE_ADR.KOOD AS SOURCE_ADR,
        K_TARGET_ADR.KOOD AS TARGET_ADR,
        K_TYPE.KOOD AS JOB_TYPE
    FROM dbo.WMS_JOB WJ (NOLOCK)
    INNER JOIN dbo.KLASSIF K_TYPE (NOLOCK)
    ON WJ.TYPE_KLF_ID = K_TYPE.ID
    INNER JOIN dbo.KLASSIF K (NOLOCK)
    ON WJ.TARGET_ART_STATUS_KLF_ID = K.ID
    INNER JOIN dbo.SUBJEKTID S (NOLOCK)
    ON WJ.STARTED_BY_SUBJ_ID = S.ID
    INNER JOIN dbo.KLASSIF K_SOURCE_ADR (NOLOCK)
    ON WJ.SOURCE_ADR_KLF_ID = K_SOURCE_ADR.ID
    INNER JOIN dbo.KLASSIF K_TARGET_ADR (NOLOCK)
    ON WJ.TARGET_ADR_KLF_ID = K_TARGET_ADR.ID
    OUTER APPLY
    (
        SELECT S.NIMI
        FROM dbo.DOKUMENT D (NOLOCK)
        INNER JOIN dbo.SUBJEKTID S (NOLOCK)
        ON D.KUHU_SUBJ_ID = S.ID
        LEFT JOIN dbo.SUBJEKTID S2 (NOLOCK)
        ON D.KUHU_YKSUS_SUBJ_ID = S2.ID
        WHERE D.ID = WJ.SOURCE_DOC_ID
    ) HANKIJA
    WHERE CAST(WJ.COMPLETED AS DATE) >= CAST(DATEADD(HOUR, -3, GETDATE()) AS DATE)
    AND K_TARGET_ADR.KOOD NOT LIKE 'RIIULPV%'
    AND K.KOOD = 'ARTOLEK_LDSTATUD'
    AND K_TYPE.KOOD IN ('WMSJOBLLIK_LIIKUMINE', 'WMSJOBLLIK_TEGKOG')
)
SELECT
    ROW_NUMBER() OVER (ORDER BY S.ACTIONS DESC, DATEDIFF(MINUTE, S.[START], S.[END])) AS [RANK],
    S.SUBJ_ID,
    TRIM(SUBSTRING(S.SUBJ_NAME, CHARINDEX(' ', S.SUBJ_NAME), 50)) AS FIRST_NAME,
    TRIM(SUBSTRING(S.SUBJ_NAME, 1, CHARINDEX(' ', S.SUBJ_NAME))) AS LAST_NAME,
    S.ACTIONS,
    S.[START],
    S.[END],
    LAST_ACTION.JOB_TYPE,
    LAST_ACTION.TARGET_ADR,
    LAST_ACTION.SUPPLIER
FROM
(
    SELECT
        T.SUBJ_ID,
        T.SUBJ_NAME,
        COUNT(*) AS ACTIONS,
        (
            SELECT MIN([STARTED])
            FROM TANA
            WHERE SUBJ_ID = T.SUBJ_ID
        ) AS [START],
        (
            SELECT MAX([STARTED])
            FROM TANA
            WHERE SUBJ_ID = T.SUBJ_ID
        ) AS [END]
    FROM TANA T
    GROUP BY T.SUBJ_ID, T.SUBJ_NAME
) S
CROSS APPLY
(
    SELECT TOP(1) SUPPLIER, JOB_TYPE, SOURCE_ADR, TARGET_ADR
    FROM TANA
    WHERE SUBJ_ID = S.SUBJ_ID
    ORDER BY [STARTED] DESC
) LAST_ACTION
```