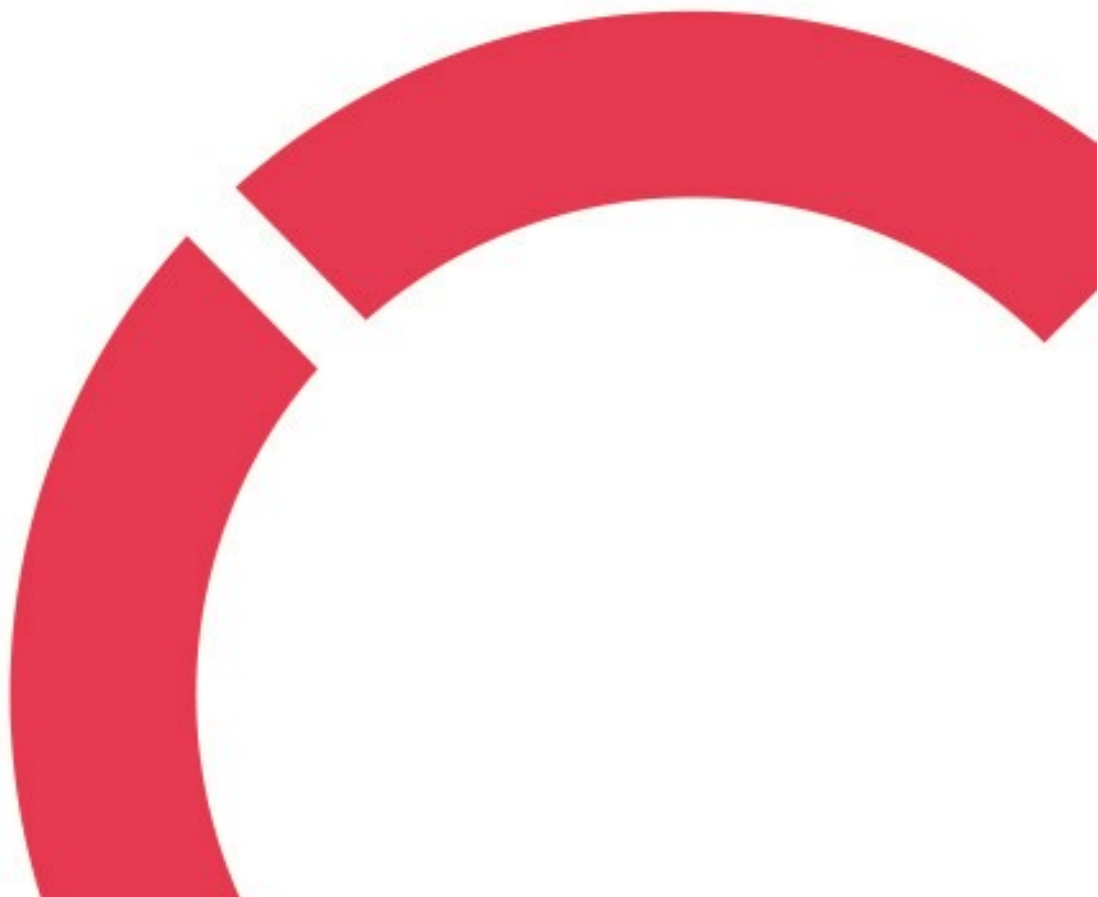


Anton Saarenpää

**DATAN TUONTI- JA VIENTITYÖKALUN RAKENTAMINEN
OSAKSI REFOX-TOIMINNANOHJAUSJÄRJESTELMÄÄ**

CASE: ATK Lillhonga Oy

**Opinnäytetyö
CENTRIA-AMMATTIKORKEAKOULU
Tieto- ja viestintäteknikan koulutus
Huhtikuu 2022**



TIIVISTELMÄ OPINNÄYTETYÖSTÄ

Centria-ammattikorkeakoulu	Aika Huhtikuu 2022	Tekijä/tekijät Anton Saarenpää
Koulutus Tieto- ja viestintäteknikka, insinööri AMK		<input checked="" type="checkbox"/> AMK <input type="checkbox"/> YAMK
Työn nimi DATAN TUONTI- JA VIENTITYÖKALUN RAKENTAMINEN OSAKSI REFOX-TOIMINNAN-OHJAUSJÄRJESTELMÄÄ. CASE: ATK Lillhonga Oy.		
Työn ohjaaja Tero Niemi		Sivumäärä 32 + 0
Työelämäohjaaja Petter Lillhonga		
<p>Opinnäytetyössä kuvataan datan tuonti- ja vientityökalun rakentamista osaksi Refox-järjestelmää. Refox-järjestelmä on ATK Lillhonga Oy:n valmistama ohjelmisto yritysten liiketoiminnan hallintaa varten. Opinnäytetyössä kuvataan työkalun rakennus kokonaisuudessaan, joka sisältää käyttöliittymän, validointi-, tallennus-, sekä laskentataulukoiden generointi järjestelmän.</p> <p>Esitellään yritys, jolle opinnäytetyö suoritetaan. Esitellään myös yrityksen kehittämä ohjelmisto, jonka osaksi työkalu rakennetaan. Näiden lisäksi esitellään eri tekniikat, joita on käytetty ohjelmiston ja työkalun toteuttamiseen.</p> <p>Kuvataan ohjelmistoprojektin aikataulun suunnittelua. Tämän lisäksi kuvataan itse laskentataulukon otsikoiden rakenteen tärkeys ja niiden suunnittelu. Kerrotaan myös käyttöliittymän suunnittelun tärkeydestä.</p> <p>Opinnäytetyössä kuvataan työkalun käyttämä tietokantarakenne. Lisäksi kerrotaan, kuinka datan hallinta on rakennettu jaettuna kolmeen eri vaiheeseen, vastaanottoon, validointiin ja tallentamiseen. Lisäksi kuvataan työkalun osuutta virheiden- ja vahinkojenhallinnasta. Kerrotaan työkalun integroinnista muuhun järjestelmään, sekä käyttöliittymän toteutuksesta.</p>		
Asiasanat Excel, laskentataulukot, PHP, toiminnanohjausjärjestelmät, validointi		

ABSTRACT

Centria University of Applied Sciences	Date April 2022	Author Anton Saarenpää
Degree programme Bachelor of Engineering, Information Technology		
Name of thesis CREATING IMPORT AND EXPORT TOOL THAT WILL BE PART OF REFOX ENTERPRISE RESOURCE PLANNING SYSTEM. CASE: ATK Lillhonga Oy.		
Centria supervisor Tero Niemi	Pages 32 + 0	
Instructor representing commissioning institution or company Petter Lillhonga		
<p>The thesis report describes the production of a data import and export tool that became part of Refox-system. Refox-system is a software created by ATK Lillhonga Oy for handling companies' business strategies. In this thesis report the creation of the tool is described. The thesis report also includes the creation of user interface, validation, saving and generation system for spreadsheets.</p> <p>The thesis report also includes an introduction of the company that commissioned the thesis. The software developed by the company is also presented, into which the tool will be built. Different technologies which were used in the creation of the software and the tool will be also introduced.</p> <p>The software project's timeline planning will be described. The importance of the spreadsheet headers and their structure will be described, as well as the importance of planning the user interface.</p> <p>The database structure used by the tool will be described. Also, the building of the data handling in the tool will be described. Data handling is divided in three separate phases: receiving, validating, and saving. Also, error and accident handling are described. The integration of the tool with the rest of the system will be described, as well as the implementation of the user interface.</p>		
Key words enterprise resource planning system, Excel, PHP, spreadsheets, validation		

KÄSITTEIDEN MÄÄRITTELY

ERP

ERP tulee englannin kielen sanoista Enterprise Resource Planning, joka tarkoittaa toiminnanohjausjärjestelmää.

PHP

PHP on rekursiivinen akronyympi englannin kielen sanoista Hypertext Preprocessor. Tämä on ohjelmointikielen nimi.

Laravel

Laravel on PHP-ohjelmointikielelle kehitetty ohjelmistokehys.

Refox

Refox on ATK Lillhonga Oy:n kehittämän toiminnanohjausjärjestelmän nimi.

Prebeo

Prebeo on Refox-järjestelmään liitettävä verkkokauppa-moduuli.

TIIVISTELMÄ
ABSTRACT
KÄSITTEIDEN MÄÄRITTELY
SISÄLLYS

1 JOHDANTO	1
2 ESITTELY	3
2.1 Opinnäytetyön aihe	3
2.2 Yrityksen esittely	3
2.3 Ohjelmistojen esittely	4
2.3.1 Refox-toiminnanohjausjärjestelmä	5
2.3.2 PHP	7
2.3.3 Laravel-ohjelmistokehys	8
3 SUUNNITTELU	9
3.1 Aikataulu	9
3.2 Laskentataulukko	10
3.3 Toimintatapa	13
3.3.1 Toimintatapa ennen työkalua	15
3.3.2 Toimintatapa työkalun kanssa	17
3.4 Käyttöliittymä	17
4 TOTEUTUS	19
4.1 Tietokantarakenne	19
4.2 Datat hallinta	21
4.2.1 Vastaanotto	22
4.2.2 Validointi	23
4.2.3 Tallentaminen	24
4.3 Virheidenhallinta	24
4.4 Vahinkojen hallinta	25
4.5 Tapahutumien kirjaaminen	25
4.6 Integrointi	26
4.7 Käyttöliittymä	26
5 JOHTOPÄÄTÖKSET	30
6 LÄHTEET	32

1 JOHDANTO

Opinnäytetyössä käydään läpi ohjelmiston suunnitteluprosessia, toteutusta ja lopputuloksia. Tavoitteena on käydä läpi ohjelmistoprojektin eri vaiheet, sekä kuvailla rakennettavaa työkalua. Sekä käydä läpi valintoja työkalun toteutuksen osalta, että perustella tehtyjä valintoja. Tarkastellaan myös projektin aikana tehtyjä virheitä ja käydään läpi ajatuksia, kuinka virheet olisi voitu välttää.

Opinnäytetyön aiheena on luoda ja integroida Refox-ohjelmistoon työkalu, jolla voidaan hallita dataa laskentataulukoiden avulla. Työkalu tulee osaksi asiakasyrityksen kehittämää ohjelmistoa. Ohjelmisto on ERP-järjestelmä (Enterprise Resource Planning), eli toiminnanohjausjärjestelmä. Toiminnanohjausjärjestelmä on yrityksille myytävä ohjelmisto, jonka kautta voidaan pitää kirjaa yrityksen ostoista ja myynneistä, sekä suunnitella yrityksen tulevaisuutta. Järjestelmästä saadaan myös raportteja, joiden avulla voidaan tarkastella yrityksen kannattavuutta.

Tarve datan hallinnalle syntyy ohjelmiston käyttäjien toimesta. Käyttäjillä on hallussaan paljon dataa laskentataulukko muodossa, sekä monille se on tutuin tapa manipuloida dataa. Tärkeää on myös saada dataa sisään järjestelmään tai ulos järjestelmästä. Laskentataulukkojen peruskäyttö on hallussa useimmilla käyttäjillä, joten tämä tapa datan manipuloinnille on tuttu jo suurelle osalle käyttäjiä. Tärkeää datan siirtämisessä on sen validointi. Järjestelmään ei pidä olla mahdollista tuoda dataa, joka rikkoisi muita ominaisuuksia. Mikäli käyttäjän tuomassa datassa on virheitä, ja ne päästetään järjestelmään, voi se aiheuttaa suurta vahinkoa käyttäjän yrityksen liiketoiminnalle. Iso osa opinnäytetyöstä keskittyykin datan validointiin, sekä virheiden- ja vahinkojen hallintaan.

Käyttäjillä on myös tarve toimittaa dataa kumppaneille, joten työkaluun rakennetaan myös mahdollisuus tuoda dataa ulos järjestelmästä. Kaikilla ohjelman käyttäjillä tulee olemaan mahdollisuus käyttää työkalua oman datan hallintaan. Tämä on tärkeä lisä jo kattaviin ominaisuuksiin, jotka löytyvät järjestelmästä.

Käyttäjien laskentataulukoiden avulla tuoma data täytyy olla tarkistettua, eli validoitua ennen kuin se tallennetaan järjestelmään. Tämä validointi on helppo suorittaa käytettävän ohjelmistokehyksen Laravel-ohjelmistokehyksen sisältämällä ominaisuudella, joka on rakennettu datan validointia varten.

Opinnäytetyössä tarkastellaan myös yrityksen nykyisiä toimintatapoja, joiden rinnalla rakennettava työkalu tulee toimimaan. Käyttöliittymä suunnitellaan ja rakennetaan mahdollisimman intuitiiviseksi, mutta silti järjestelmän yleisasuun sopivaksi.

2 ESITTELY

Opinnäytetyö toteutetaan yritykselle ATK Lillhonga Oy. Opinnäytetyö tulee olemaan osa yrityksen kehittämää Refox toiminnanohjausjärjestelmää. Tarve opinnäytetyön työkalulle on syntynyt Refox:n käyttäjien toimesta. Käyttäjät tarvitsevat helpon ja turvallisen tavan tehdä suuriakin muutoksia Refox:n sisältämään tuotekatalogiin tai muihin tietoihin.

2.1 Opinnäytetyön aihe

Opinnäytetyön aiheena on luoda Refox-ohjelmistoon työkalu, jolla voidaan muokata dataa laskentataulukoiden avulla. Toimintaperiaate työkalulle on yksinkertainen, ja se voidaan jakaa kolmeen vaiheeseen. Ensimmäisenä käyttäjä lataa haluamansa tiedot ulos järjestelmästä. Esimerkiksi kaikki tuotteet joiden hinta on alle 10e. Työkalu generoi käyttäjälle olemassa olevan datan perusteella laskentataulukon. Toisessa vaiheessa käyttäjä tekee muutokset laskentataulukkoon haluamallansa ohjelmalla. Kun käyttäjä on saanut tehtyä tahtomansa muutokset, voidaan sama laskentataulukko ottaa vastaan työkalussa. Kolmannessa vaiheessa käyttäjä lataa laskentataulukon takaisin Refox:iin työkalun avulla, ja työkalu tarkistaa datan. Tämän jälkeen työkalu ilmoittaa käyttäjälle mahdollisista löytämistään virheistä ja muutoksista, joita se aikoo suorittaa. Sitten käyttäjä voi hyväksyä muutokset ja tallentaa datan.

Työkalun toiminta rajataan opinnäytetyön osuudelta koskemaan vain tuotteita. Työkalu kuitenkin rakennetaan tulevaisuus mielessä ja työkalun toiminta-alueen helppo laajentaminen koskemaan muitakin osia toiminnanohjausjärjestelmää on tärkeää. Rajaus määritellään tuotteisiin, koska tuotteiden muokkaaminen on yleisin tarve, joka ohjelmiston käyttäjillä esiintyy.

2.2 Yrityksen esittely

ATK Lillhonga on vuonna 1990 perustettu yritys. Yritys aloitti toimintansa tarjoten ATK-palveluita ja toimien nimellä ATK-Lillhonga T:mi. Yrityksen perustajana toimi Johan Lillhonga. Vuonna 1993 yritys toimitti ensimmäisen ohjelmiston Foxadmin, joka oli toimi työkaluna teollisessa myynnissä. Tämän jälkeen ohjelmiston kehitys jatkui ja vuonna 1997 tuli Visual Foxadmin. Visual Foxadmin on Windows-pohjainen ERP-järjestelmä. Vuonna 2003 yritykseen otettiin ensimmäinen työntekijä, samoihin aikoihin

julkaistiin SQLFoxadmin, ohjelmisto, joka perustui SQL-kyselykieleen. Vuonna 2005 yritys toimitti ensimmäisen verkkokaupan, tämä oli rakennettu PHP-ohjelmointikielellä, käyttäen itse rakennettua ohjelmistokehystä. Tämän jälkeen yritys jatkoi kasvuansa ja vuonna 2007 yritys muuttui osakeyhtiöksi. Yrityksen tärkeimpiä asiakkuuksia solmittiin vuonna 2008, kun yritys aloitti yhteistyön Varustelean kanssa. (Refox 2022). Varustelega on suomalainen verkkokauppa, joka on ulkoilun ja armeijajalijäämän erikoisliike. (Varustelega 2022).

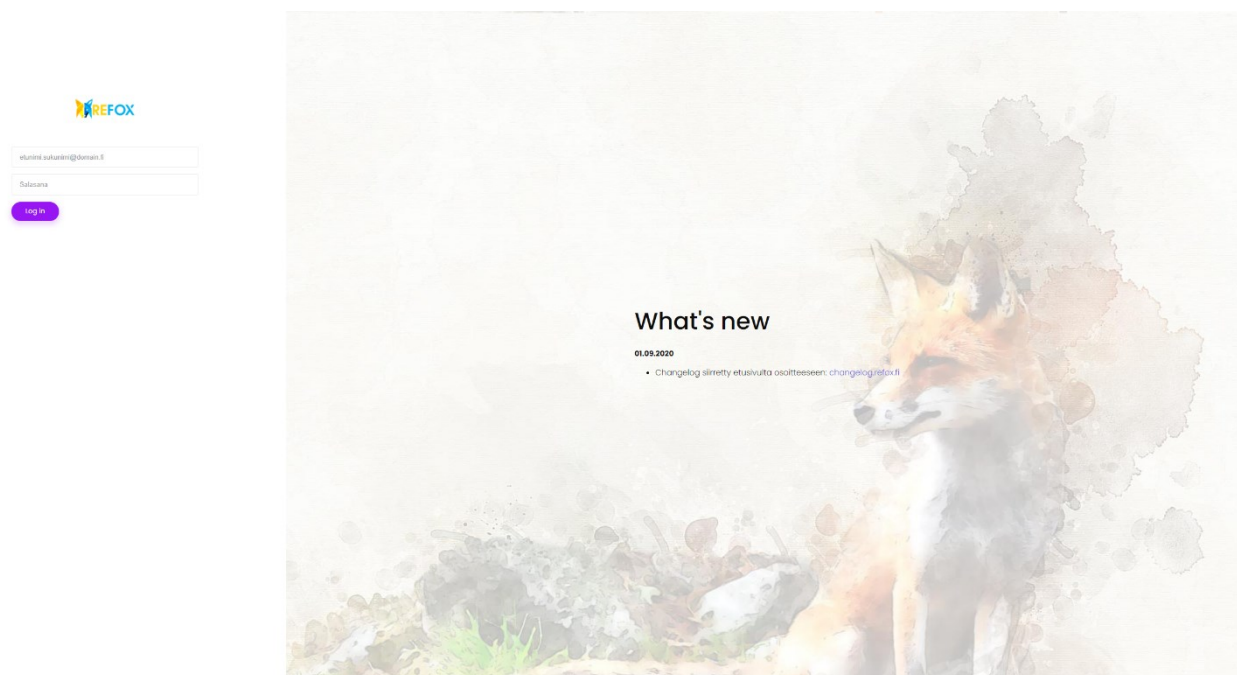
Vuonna 2013 yritys tuli käännekohtaan, kun Johan Lillhonga, yrityksen perustaja jättäytyi pois yrityksestä. Yrityksen omistajana ja toimitusjohtajana jatkoi Petter Lillhonga. Vuoteen 2013 mennessä yrityksessä oli henkilöstöä neljä. Refox-järjestelmän suunnittelutyö aloitettiin vuonna 2014. Kaikki vanha ohjelmisto jätettiin taakse, ja päätettiin aloittaa rakentaminen puhtaalta pöydältä. Ensimmäisen Refox-järjestelmän toimitus tapahtui vuonna 2016. Refox-järjestelmän kehitys tahti kiihtyi, kun järjestelmä oli aktiivisessa käytössä. Vuonna 2016 myös uudistettiin brändiä ja suunniteltiin tulevaisuutta. Jo vuonna 2017 Refox:sta tuli Lillhonga Oy:n pääasiallinen ohjelmisto. Vanhoja asiakkaita, jotka vielä käyttivät aktiivisesti SQLFoxadmin-ohjelmistoa, siirrettiin käyttämään uutta Refox-järjestelmää. (Refox 2022).

Yrityksen kasvu jatkoi kovaa tahtia, vuonna 2017 Refox-järjestelmiä oli jo useita kymmeniä aktiivisessa käytössä. Ohjelmiston myynnin kasvun ohessa myös henkilöstö määrä kasvoi, 2017 yrityksen henkilöstö määrä oli seitsemän. Kasvu jatkoi kiihtymistään ja vuonna 2019 liikevaihto kasvoi 20 %. Yritys on jatkanut kasvuaan ja vuonna 2022 henkilöstöä on yrityksessä jo neljätoista. (Refox 2022).

2.3 Ohjelmistojen esittely

Seuraavaksi esitellään eri ohjelmistot, joiden avulla Refox-toiminnanohjausjärjestelmä ja tähän tiukasti integroitu Prebeo-verkkokauppa ovat rakennettu. Ohjelmisto kokonaisuus, johon opinnäytetyössä rakennettava työkalu tulee osaksi, koostuu Refox-toiminnanohjausjärjestelmästä ja Prebeo-verkkokaupasta. Refox:n avulla ohjataan verkkokaupan toimintaa, tämä sisältää tuotekatalogit ja kaiken tarvittavan sisällön verkkokauppaa varten. Lisäksi käydään läpi myös ohjelmistoa, jonka avulla on rakennettu molemmat Refox, kuin Prebeo-verkkokaupakin.

2.3.1 Refox-toiminnanohjausjärjestelmä



KUVA 1. Refox-järjestelmän kirjautumissivu.

Ohjelmisto, jota työkalu tulee osaksi, on nimeltään Refox. Refox on toiminnanohjausjärjestelmä, jonka kehittäjänä toimii ATK Lillhonga Oy. Ohjelmisto toimii selaimessa, joten se ei vaadi erillistä asennusta ja se toimii lähes jokaisella modernilla laitteella, jossa on verkkoyhteys. Refox on suunniteltu käytettäväksi tietokoneella, mutta toimii myös tablet- ja mobiililaitteilla. Kuvassa 1 voidaan nähdä Refox-järjestelmän kirjautumissivu. Jokainen Refox-järjestelmä on asiakaskohtainen, eli ei ole olemassa yhtä Refox-järjestelmää, jota kaikki ohjelmiston käyttäjät hyödyntävät. Jokaisella Refox:n käyttäjällä on käytössään oma instanssi Refox:sta. Jokaiseen instanssiin voidaan sitten luoda käyttäjiä eri käyttötarkoituksiin. Eri käyttäjien oikeuksia ohjelmiston sisällä voidaan hallita rajaamalla ominaisuuksia vain tietyille käyttäjäryhmille.

Toiminnanohjausjärjestelmällä yleisesti tarkoitetaan järjestelmää, jolla voidaan pitää yllä historia tietoja yrityksen toiminnasta. Nämä historiatiedot pitävät sisällään muun muassa yrityksen ostetun tavaran, myydyin tavaran, sekä varastonhallintaan tarvittavat tiedot. Refox-järjestelmä toimittaa raportteja sisäiseen käyttöön, sekä mahdollistaa erilaisten arvojen seuraamista. Voidaan esimerkiksi tulostaa raportti tietyn tuotteen tai tuoteryhmän myynneistä halutulta ajanjaksolta. Järjestelmä pitää yllä myös tuotteiden saldotietoja, nämä ovat laskennallisia arvoja, jotka perustuvat yrityksen ostoihin ja myynteihin.

Ohjelmisto on myös helposti integroitavissa ulkoisiin järjestelmiin. Useita eri integraatioita on rakennettu käyttäjien tarpeiden mukaan. Yleisimpiä integraatioiden tarpeita on erilaisten hinnastojen hakeminen suoraan toimittajilta. Yleensä integraatiot rakennetaan ohjelmiston käyttäjäkohtaisesti, koska käyttäjiä on monilta eri aloilta. Näiden integraatioiden etuna on se, että saadaan verkkokauppaan suoraan tieto toimittajan varastosta. Näin voidaan verkkokaupassa näyttää tuote tilaustavarana, vaikka se olisikin omasta varastosta loppu. Integraatioita on myös rakennettu muihin järjestelmiin, näistä muutamia on Unifaun, Procounor ja Verifone. (Lillhonga 2022).

Osana Refox:a toimitetaan yleensä myös verkkokauppa. Verkkokauppa ja Refox ovatkin integroitu tiukasti yhteen. Verkkokaupan tuotteita ja ulkoasua hallitaan suoraan Refox:sta, verkkokaupalle ei siis ole erillistä hallintaa. Myös verkkokaupasta tehdyt tilaukset tulevat suoraan Refox:n. Verkkokauppoja voi olla yksi tai useita yhtä Refox:a kohden. Refox:sta hallitaan niin verkkokaupan visuaalista ilmettä, kuin itse sisältöä. Verkkokaupoissa esiintyvät alennukset, kampanjat sekä muut ominaisuudet ovat suoraan hallittavissa Refox:sta.

Ohjelmistoon on myös luotu kassa toiminto, joten tätä voidaan käyttää kivijalkaliikkeen kassana. Tähän tarvitsee kolmannen osapuolen toimittaman päätteen, mutta itse päätteen hallinta onnistuu Refox:sta. Pääte toimii myös kaikilla laitteilla, joilla Refox toimii. Pääte ei tarvitse fyysistä yhteyttä laitteeseen, jossa Refox:a käytetään, vaan pääte toimii internetin yli. Kassan riippumattomuus fyysisestä sijainnista luo paljon joustavuutta yrityksen liiketoiminnalle. Nämä ominaisuudet helpottavat käyttäjän toimintaa, kun hän voi toimia yhdessä sovelluksessa ja hallita sieltä koko yrityksen liiketoimintaa.

Prebeo-verkkokauppa on moduuli, joka integroidaan Refox-järjestelmään. Tämä ei siis toimi itsenäisesti vaan vaatii Refox:n toimiakseen. Kuten aikaisemmin jo mainittiin, kaikkea verkkokauppaan liittyvää hallitaan suoraan Refox:n käyttöliittymästä. Refox puolestaan voi toimia itsenäisesti, mikäli halutaan ohjelmistoa käyttää ilman verkkokauppaa. Verkkokaupan suurena etuna on sen tiukka integraatio toiminnanohjausjärjestelmän kanssa. Kaikki verkkokaupan myynnit siirtyvät suoraan Refox:n tilausputkeen. Tilaus pysyy siis koko ajan samassa järjestelmässä, eikä tarvitse siirtyä ohjelmistosta toiseen tai luoda erillisiä integraatioita. (Lillhonga 2022).

2.3.2 PHP

Ohjelmiston palvelimilla ajettavana ohjelmointikielenä käytetään PHP:tä. PHP on palvelimilla käytettävä ohjelmointikieli. Vuonna 2022 PHP on palvelimin applikaatioissa käytettävä ohjelmointikieli ja jopa 78,1 % verkkosivuista käyttää PHP:tä toiminnassaan. (W3Techs.com 2022). PHP on yleiskäyttöinen skriptauskieli, joka on suunniteltu käytettäväksi web-ohjelmoinnissa. PHP:tä voidaan upottaa osaksi HTML:ää. (PHP.net 2022). HTML tulee englannin kielen sanoista ”Hyper Text Markup Language” ja tämän avulla voidaan tarjota rakenne verkkosivuille. (Mozilla 2022). PHP on rekursiivinen akronyymi sanoista ”Hypertext Preprocessor”.

PHP on komentosarjakieli, eli koodi käsitellään rivi kerrallaan tiedostoa luettaessa. Tämä poikkeaa perinteisemmästä tavasta ajaa koodia, kuten esimerkiksi C-kielen koodi täytyy ennen suorittamista koota binäärikoodiksi. (Javapoint.com 2021.). PHP-koodi yleensä upotetaan HTML:n sekaan, jotta voidaan rakentaa dynaamisia verkkosivuja. PHP:n etuna on sen riippumattomuus käytettävästä käyttöjärjestelmästä. Näin voidaan rakentaa applikaatioita, jotka toimivat jokaisella käyttöjärjestelmällä. PHP-ohjelmointikieli on myös avointa lähdekoodia (eng. open source), tämä tarkoittaa sitä, että kuka tahansa voi tarkistaa kuinka ominaisuudet on rakennettu ja ehdottaa myös ratkaisuja mahdollisiin ongelmiin, joita voi koodista löytyä. (Kolade 2021).

PHP toimii hyvin eri tietokantojen kanssa, PHP voidaan yhdistää perinteiseen relationaaliseen tietokantaan, kuten MySQL. Yhteys voidaan kuitenkin luoda myös tietokantoihin, jotka eivät toimi kuten perinteiset tietokannat, kuten MongoDB tietokantaan. PHP:llä on myös aktiivinen yhteisö verkossa. Tämä tarkoittaa sitä, että jos virallisesta dokumentaatiosta ei löydy ohjeita ongelman ratkaisemiseen, löytyy todennäköisesti keskustelufoorumeilta joku, jolla on ollut sama ongelma tai henkilö, joka osaa ratkaista ongelman. Useat suosittu palvelut internetissä on rakennettu PHP-ohjelmointikieltä käyttäen. Esimerkiksi Facebook, Wikipedia ja Wordpress käyttävät PHP:tä. (Kolade 2021).

2.3.3 Laravel-ohjelmistokehys

Laravel-niminen ohjelmistokehys on PHP-ohjelmointikielelle rakennettu ohjelmistokehys. Tämä on pääosin palvelin puolen ohjelmointia varten käytettävä ohjelmistokehys, kuten PHP kielenäkin on palvelimille suunniteltu kieli. Tämä ei kuitenkaan tarkoita sitä, etteikö näistä olisi hyötyä myös ulkoasun rakentamiseen.

Laravel on avoimen lähdekoodin ohjelmistokehys. Laravel on suunniteltu käyttämään MVC-ohjelmistoarkkitehtuuria. MVC tulee englannin kielen sanoista ”Model View Controller”. Ohjelmistokehysten käyttäminen pakottaa verkkosivu-projektin koodin struktuurin hyvin skaalautuvaksi. Laravel on myös rakennettu säästämään kehittäjän aikaa, kun rakennetaan verkkosivu tyhjästä. Ohjelmistokehyksessä on otettu huomioon myös useimpien verkkohyökkäysten riskit, joten Laravel:lla rakennettu web-applikaatio on turvallinen. (Tutorialspoint.com 2021).

Laravel-ohjelmistokehysten isoja etuja on nopea ja yksinkertainen reititys moottori. Tarkoittaen tapaa luoda verkkosivuille uusia reittejä. Uuden reitin luominen ei vaadi, kuin URI:n ja ohjeet mitä tehdä, kun kyseistä reittiä kutsutaan selaimessa. (Laravel 2022a). URI eli Uniform Resource Identifier tarkoittaa merkkijonoa, joka on uniikki ja viittaa johonkin palvelimella tarjottavaan resurssiin. (Awati 2021). Laravel hallitsee myös sessioita ja välimuistia, jotta näitä ei tarvitse käyttäjän itse rakentaa. Sessiot tarkoittavat käytännössä yhden käyttäjän istuntoa verkkosivulla. Sessioon voidaan tallentaa käyttäjän valintoja tai muuta dataa. (Laravel 2022b). Esimerkiksi jos verkkosivu mahdollistaa taustan värin vaihtamisen, voidaan tämä valinta tallentaa sessioon. Välimuisti puolestaan nopeuttaa sivun toimintaa, koska jos usea eri käyttäjä hakee samaa resurssia verkkosivulta, ei tarvitse tätä joka kerta hakea logiikan mukaan uudestaan, vaan voidaan välimuistista palauttaa tarvittava resurssi. (Laravel 2022c).

3 SUUNNITTELU

Työkalua suunnitellessa on tärkeää tietää mitä yritys odottaa työkalulta, ja mitä on mahdollista toteuttaa, niin teknisestä näkökulmasta, kuin oman osaamisen näkökulmasta. Yrityksen kanssa tiiviillä kommunikaatiolla voidaan varmistaa, että työkalusta tulee tarpeellinen ja toivottu lisä ohjelmistoon. Suunnittelu vaiheessa pitää osata tehdä arvio työkalun laajuudesta ja sen toimitusajasta. Nämä ovat tärkeitä osia projektin etenemiselle, koska näiden perusteella yritys voi tehdä kustannusarvion. Lisäksi voidaan arvioida suunnitelman ja kustannusarvion pohjalta itse projektin toteuttamisen kannattavuus. Eli onko kustannukset hyväksyttävät suhteutettuna suunnitelman mukaisen työkalun tuomaan hyötyyn.

Suunnittelussa käytetään apuna laajaa kokemusta asiakkaiden tarpeesta, joka on syntynyt yrityksen vuosien mittaisesta yhteistyöstä asiakkaiden kanssa. Työkalua suunnitellessa on tärkeää suunnitella loppukäyttäjälle mahdollisimman intuitiivinen kokemus, kuitenkin luopumatta tehokäyttäjien tarvitsemista ominaisuuksista.

3.1 Aikataulu

Aikataulun suunnittelu on tärkeä pohja projektille. Ohjelmistoprojekteissa aikataulut ovat tärkeitä, jotta voidaan ennustaa kustannuksia. Aikataulun luominen on yksinkertainen abstraktilla tasolla. Ensin suunnitellaan työkalu ja sen ominaisuudet. Tämän jälkeen arvioidaan eri ominaisuuksien vaatimaa aikaa, ja näin voidaan helpommin tehdä päätöksiä eri ominaisuuksista. Kaikkien ominaisuuksien kannattavuus voidaan arvioida vertaamalla ominaisuuden tuottamaa hyötyä sen vaatimaan kehitysaikaan. Kun halutut ominaisuudet ja niiden laajuudet ovat tiedossa, voidaan koko projektin ajankäytöstä tehdä raaka-arvio. Tämä esitetään asiakkaalle, ja mikäli työkalun tuomat hyödyt ovat hyväksyttäviä sen vaatimaan aikaan on projekti hyvässä aloituspisteessä. Ajankäytön seuranta on myös tärkeää, jotta voidaan asiakkaalle kertoa tietoja projektin etenemisestä. Lisäksi on tärkeää tehdä sisäistä arviointia projektin aika-arvioista kesken työn. Näin ollen voidaan varmistua projektin valmistumisesta ajallaan, tai tarvittaessa voidaan neuvotella projektille lisäaikaa.

Projektia alettiin suunnittelemaan pitämällä aloituspalaveri yrityksen kanssa. Tässä palaverissa yritys antoi pohjavaatimukset projektille. Näiden vaatimusten pohjalta aloitettiin suunnittelemaan projektin toteutusta. Projektin suunnitteluun ja hallintaan käytettiin yrityksen sisäisesti kehittämää työnhallinta

järjestelmää, joka on osa Refox:a. Käydään seuraavaksi läpi yrityksen asettamia vaatimuksia työkalun toiminnalle. Työkalulla pitää pystyä muokkaamaan dataa käyttäen laskentataulukoita muokkausalus- tana. Kuten aikaisemmin jo mainittiin, rajattiin työkalun muokkaus kattamaan aluksi vain tuotteita, jotka ovat keskinäisin osa Refox:n käyttämää dataa.

Työkalun pitää pystyä tarkistamaan käyttäjän tekemät muutokset ennen kuin ne tallennetaan järjestel- mään. Lisäksi kaikista tallennuksista pitää jäädä merkintä. Näiden merkintöjen pitää voida myös palaut- taa tiedot ennalleen, mikäli käyttäjä on muokannut väärin tietoja ja haluaa palata tilaan ennen muok- kausta. Käyttöliittymän pitää olla intuitiivinen ja yksinkertainen. Käyttäjän pitää pystyä käyttämään työ- kalua ilman että tämä nostaa teknisen tuen kyselyitä tarpeettoman paljon. Lisäksi kaikki data, joka työ- kalulla tallennetaan pitää tarkistaa ennen tallentamista. Tarkistamalla data ennen tallentamista voidaan ennaltaehkäistä virheitä järjestelmässä. Tuotetiedoissa esiintyvät virheet voivat aiheuttaa vahinkoa käyt- täjän yrityksen liiketoiminnalle tai aiheuttaa odottamattomia virheitä muussa osassa toiminnanohjaus- järjestelmää. (Lillhonga palaveri 2021a).

Näiden vaatimusten avulla luotiin ensimmäinen suunnitelma ja aika-arvio suunnitelman työvaiheiden pohjalta. Suunnitelman valmistuttua pidettiin yrityksen kanssa toinen palaveri, jossa käytiin läpi suun- nitelman toteutuskelpoisuutta. Suunnitellessa heräsi kysymyksiä toteutukseen, joten tämä toinen pala- veri oli tarpeellinen molemmille osapuolille. Suunnitelman läpikäynnissä yritykseltä tuli tarkennuksia vaatimuksiin, sekä vastauksia kysymyksiin, joita oli herännyt suunnitelmaa tehdessä. Suunnitelmaa pa- rannettiin ja tarkennettiin tarpeellisilta osilta. Näin saatiin aikaiseksi suunnitelma, johon sekä yritys, että työkalun toteuttaja on tyytyväinen. Kun suunnitelmaan oli toteutettu yrityksen vaatimat tarkennukset, toimitettiin tämä vielä hyväksyttäväksi.

3.2 Laskentataulukko

Iso ongelma työkalun toiminnassa on se, kuinka yhdistetään useassa eri tietokantataulussa sijaitseva data yhteen laskentatauluktoon, ja kuinka se voidaan tehokkaasti tarkistaa ja tallentaa. Näytettävää dataa on kahta eri tyyppiä. Osassa tapauksista on toivottua näyttää tuotteeseen liittyvät tietokantataulun kaikki data osana laskentataulukkoa. Toisessa tapauksessa tahdotaan näyttää vain yhden kentän data taulu- kossa. Näistä hyvin esimerkkeinä on tuotteen eri kieliversiot ja varastosaldot. Kieliversioista tahdotaan näyttää joka kielen versiot, ja tässä tietokantataulussa sijaitsee myös tuotteen nimen lisäksi kuvaus ja mahdolliset lisäkuvaukset. Nämä kaikki tahdotaan näyttää laskentataulukossa, joten kaikista helpointa

on lisätä koko taulun data ryhmitellen ne kielen mukaan tuotteen tietojen perään. Varastosaldojen kohdalla tahdotaan näyttää vain tuotteen vapaa saldo, koska muut arvot liittyen tuotteen varastosaldoihin ovat laskennallisia ja perustuvat tähän arvoon.

Kun useasta eri tietokantataulusta liitetään dataa samaan laskentataulukkoon, syntyy ongelma. Kuinka eri laskentataulukoiden kentät osataan yhdistää koneellisesti oikeaan tietokantatauluun, sekä oikeaan kenttään tässä tietokantataulussa. Tämä ongelma ratkaistaan lisäämällä otsikot laskentataulukkoon, joihin voidaan kaikki tarvittava tieto merkitä. Laskentataulukko kuitenkin tahdotaan pitää mahdollisimman käyttäjäystävällisenä ja käyttäjälle kryptiset tietokantataulujen nimet ja kenttien nimet eivät kerro mitään. Tästä syystä laskentataulukkoon suunnitellaan kolme eri otsikkoa. Ylimmäksi otsikoksi tulee selkokielineen, käyttäjän ymmärtämä otsikko. Toiselle riville tulee kuvaus kentästä, eli selkokielineen nimen lisäksi on myös tarkempi kuvaus kentän sisältämästä datasta. Kolmannelle riville tulee työkalun käyttämä merkintä, johon tallennetaan kyseisen kentän nimi, tietokantataulu ja kuinka data kuuluu yhdistää tuotteeseen.

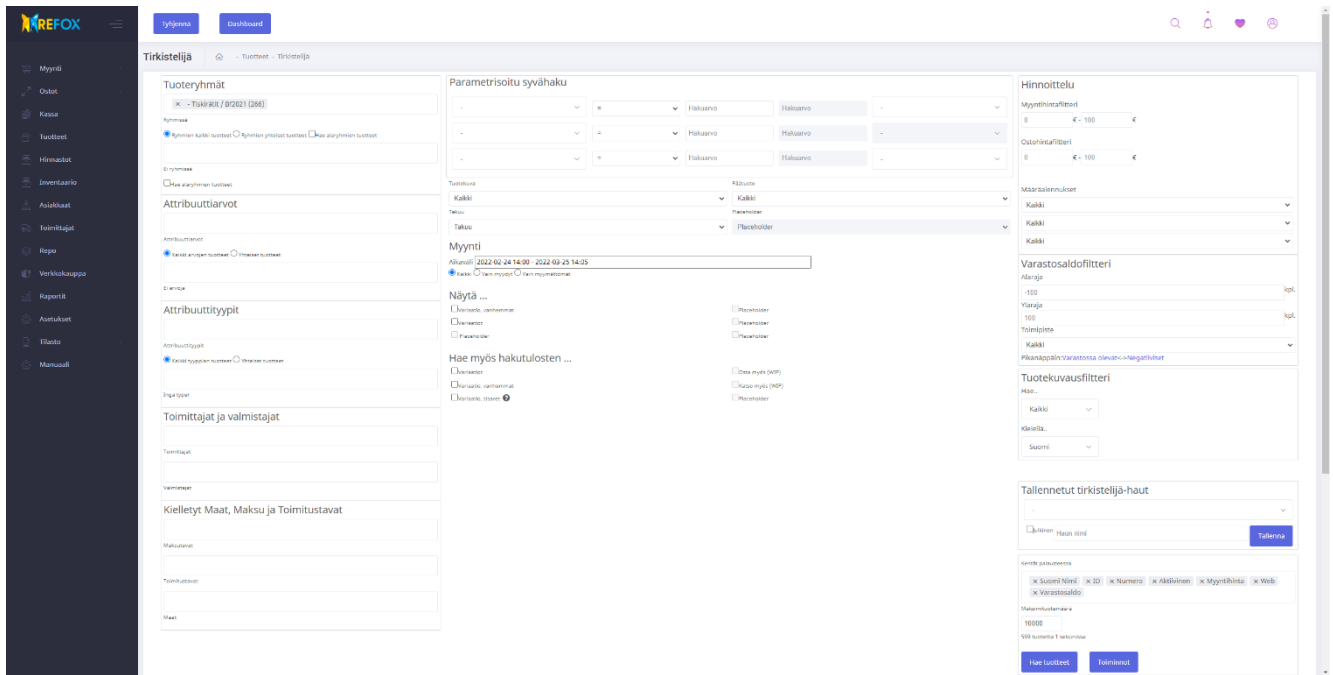
TAULUKKO 1. Esimerkki laskentataulukon rakenteesta.

id	Tuotenumero	Aktiivinen	Myyntihinta	Ostohinta	Vapaa saldo
Identifioiva juokseva numero	-	Määrittää onko tuote aktiivinen vai ei, hyväksyy arvoja 1 tai 0	-	-	Tuotteen vapaa saldo tiettyssä varastossa.
id	number	active	sales_price	purchase_price	stock:free_stock-1
1	A1	1	10	5	12
2	A2	0	0	0	0
3	A3	1	15	10	32

Taulukossa 1 voidaan nähdä osa esimerkki laskentataulukon rakenteesta ja datasta. Ensimmäisellä rivillä voimme havaita jo aiemmin mainitun selkokielineen otsikon. Tämä otsikko on käyttäjille suunnattu. Toi-

sella rivillä on laajempi kuvaus otsikosta, mikäli kenttä on sellainen, että se tarvitsee kuvauksen. Kolmannella rivillä on työkalun käyttämä merkintä, nämä yleensä ovat suoraan tietokantakentän nimiä, mutta esimerkissä voimme havaita ”Vapaa saldo” otsikon alla olevan kolmannen rivin otsikon. Tämä otsikko poikkeaa merkittävästi muista. Tässä voimme havaita rakenteen, joka avataan työkalussa seuraavalla logiikalla. Ensimmäisenä ennen ”:”-merkkiä on tietokantataulun nimi, joka tässä esimerkissä on ”stock”. Tämän jälkeen ”:”-merkin jälkeen on tietokantataulun kentän nimi.

Lisäksi mikäli relaatio on ”yksi useaan” (eng. has many). Voidaan lisätä merkkijonoon vielä ”-”-merkki ja tämän jälkeen merkitä numero. Tämä numero viittaa ”stock”-taulussa kenttään, joka mahdollistaa useiden relaatioiden olemassaolon. Kyseisessä esimerkissä ”stock”-taulussa on kenttä nimeltä ”storages_id”, johon tuo numero ”-”-merkin jälkeen viittaa. Tämä ”storages_id”-kenttä ei käy ilmi mitenkään laskentataulukosta, mutta työkalu osaa sisäisesti yhdistää esiintyvän numeron tuohon kenttään. Mikäli tuota numero ei olisi näkyvässä, tarkoittaisi se sitä, että relaatio on ”yksi yhteen” (eng. has one). Ja tällöin ei tarvittaisi erillistä tietoa yksilöimään dataa.



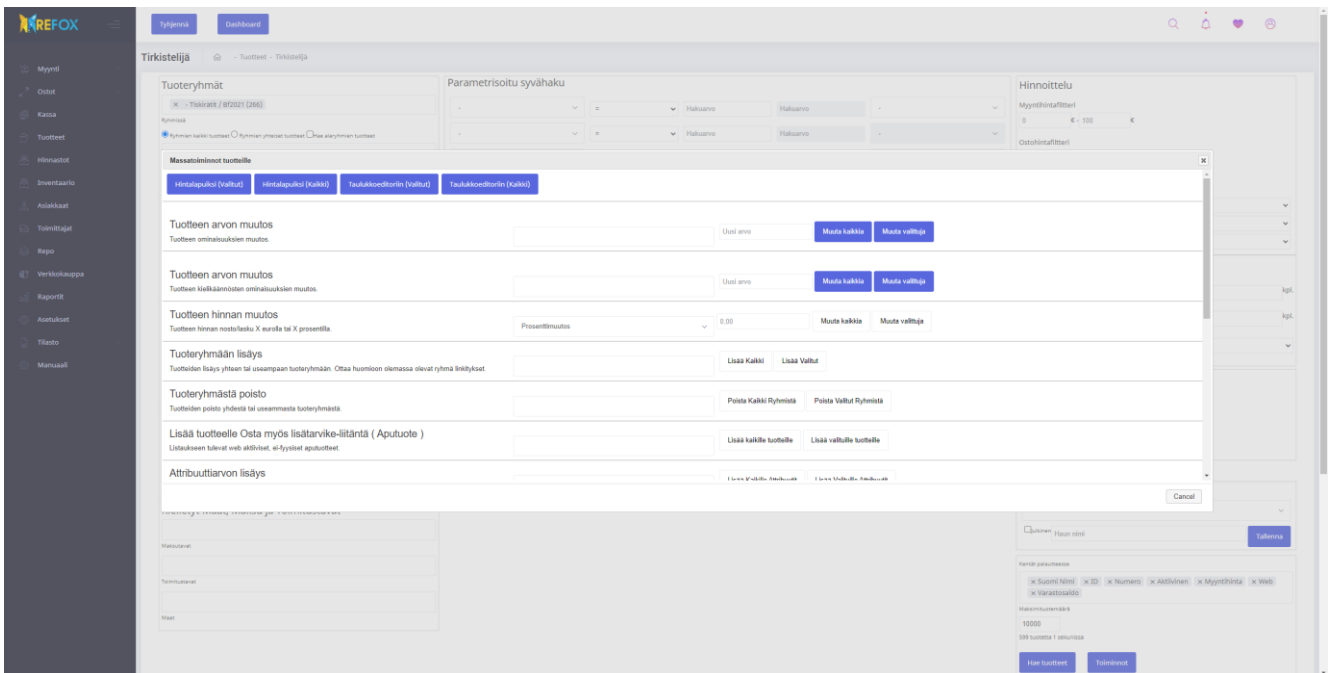
KUVA 3. Tirkistelijä-hakutyökalu.

Tämän ongelman ratkaisemiseen käytetään ohjelmistosta jo löytyvää laajaa tuotteiden hakutyökalua. Tämä hakutyökalu voidaan nähdä kuvassa 3. Tällä hakutyökalulla voidaan hakea listaus käyttäen erilaisia parametrejä, kuten esimerkiksi pystytään hakea vain tietyn tuoteryhmän tuotteet. Kuvassa 3 on Tirkistelijää käyttäen haettu saman tuoteryhmän tuotteet, kuin kuvassa 2. Tämä hakutyökalu tahdotaan integroida datanhallintatyökaluun, jotta voidaan laskentataulukoita luodessa käyttää valmiiksi rajattuja tuotelistauksia.

Saatuaan rajatut tuotteet laskentataulukkoon, on käyttäjän helpompaa tehdä tahtomansa muokkaukset. Lisäksi tämä on myös kevyempää työkalulle, koska ei tarvitse parsia tuhansien rivien mittaisia laskentataulukoita ja etsiä niistä tehtyjä muutoksia. Tehtyään muutokset laskentataulukkoon voidaan aloittaa työkalussa tarkastus ja tallennus prosessit.

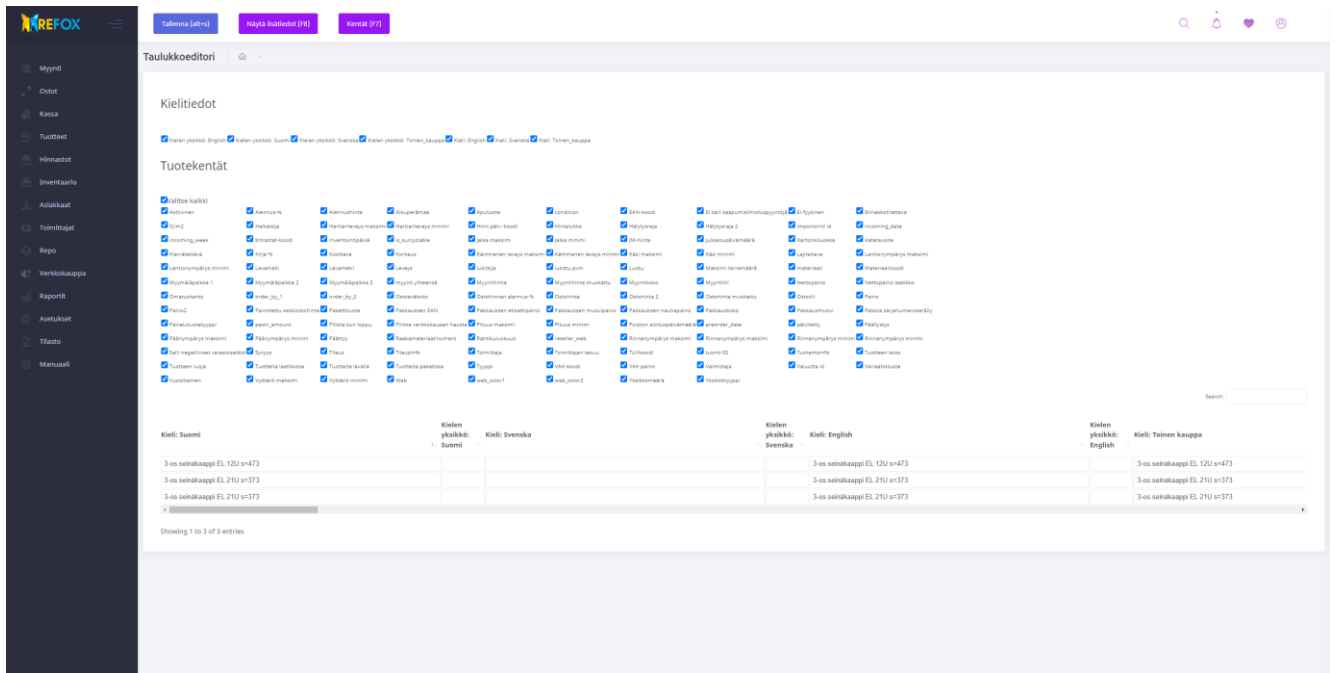
Ensin käyttäjä luovuttaa työkalulle muokatun laskentataulukon. Työkalu parsii laskentataulukossa olevan datan. Tämän jälkeen työkalu tarkistaa, että data on oikeassa muodossa. Lisäksi tässä vaiheessa työkalu tarkistaa mitä muutoksia käyttäjä on tehnyt tietoihin. Työkalu ei tallenna vielä tietoja, vaan palauttaa käyttäjälle näkymän, jossa ilmoitetaan mahdolliset virheet datassa, sekä muutokset joita laskentataulukon tallentaminen suorittaa. Käyttäjä voi tässä kohtaa tarkistaa, että tehdyt muutokset ovat haluttuja, sekä itse muutoksissa ei ole virheitä.

Tätä tapaa muokata on parannettu luomalla massamuokkaus ominaisuudet aikaisemmin mainittuun hakutyökaluun. Täällä voidaan muokata yleisimpiä arvoja, näitä havainnollistetaan kuvassa 5. Esimerkiksi voidaan hakea yhden tuoteryhmän kaikki tuotteet ja vaihtaa niiden kaikkien hinta arvoon x. Tällöin ei tarvitse avata jokaista tuotetta erikseen ja tehdä siihen muutoksia. Tämä on hyvä, mikäli muutokset tuotteisiin on yleisiä ja vaatii vain tietyn datan muuttamista selkeästi yhdestä toiseksi.



KUVA 5. Tirkistelijä-hakutyökalun massamuokkaus valinnat.

Kolmantena tapana muokata tuotteita on avata ohjelmistosta Taulukkoeditori, tämä voidaan nähdä kuvassa 6. Tämä matkii laskentataulukon ulkoasua ohjelman sisällä. Tähän voidaan myös avata hakutyökalua käyttäen tuotteita. Tämän toiminnan heikkoutena on sen mahdollisuus syöttää virheellistä dataa. Lisäksi taulukkonäkymässä ei myöskään ole mahdollista muokata kaikkea dataa, vaan vain yleisimpiä muokattavia kenttiä. Vaikka taulukkonäkymä matkiikin laskentataulukon ulkonäköä, ei se silti ole yhtä tuttu ja helppo käyttää, kuin käyttäjille tutut laskentataulukot. Kuvassa 6 on valittu kolme tuotetta taulukkoeditoriin, sekä suuri listaus eri valintoja itse taulukon yläpuolella on piilotettavissa. Tästä voidaan valita mitä kenttiä itse taulukossa tahdotaan näyttää.



KUVA 6. Taulukkonäkymä

3.3.2 Toimintatapa työkalun kanssa

Työkalun suurimpia vahvuuksia verrattuna edellisiin toimintatapoihin on mahdollisuus tehdä muutoksia todella laajasti järjestelmään. Tämä mahdollisuus muokata kaikkea dataa järjestelmässä on tehokäyttäjille oiva lisä. Työkalu laajuudestaan huolimatta on kuitenkin turvallinen käyttää, koska kaikki data tarkistetaan ennen tallentamista. Tästä syystä käyttäjien tekemät näppäily- tai ajatusvirheet voidaan saada kiinni ennen kuin ne päätyvät tuotanto ympäristöön.

Lisäksi työkalu pitää kirjaa kaikista sen tekemistä muutoksista. Mikäli virhe on päässyt tallentumaan järjestelmään, on helppo käydä palauttamassa arvot ennalleen. Näistä historiatiedoista voidaan myös tarkistaa mitä, kuka ja milloin on tehnyt.

3.4 Käyttöliittymä

Käyttöliittymän suunnittelu on vähintäänkin yhtä tärkeä, kuin itse taustalla pyörivän sovelluksen toteutus. Rakentamalla yksinkertainen ja intuitiivinen käyttöliittymä voidaan siirtää käyttäjiltä tulevaa painetta pois asiakaspalvelulta. Käyttöliittymän on myös tärkeä sopia visuaalisesti ohjelmistoon. Nämä

kaikki osapuolet käyttöliittymässä lisää käyttäjän itsevarmuutta käyttää työkalua. Työkalun ollessa käyttäjillä käytössä on tärkeää, että käyttäjä voi keskittyä tekemiinsä muutoksiin. Mikäli käyttäjä joutuisi epäilemään työkalun käyttöliittymässä esiintyviä elementtejä, veisi tämä turhaan huomiota pois muutoksista ja lisäisi virheiden syntymisen mahdollisuuksia. Käyttöliittymässä mukaillaan muita käyttöliittymiä mitä ohjelmistossa on jo rakennettu. Lisäksi pyritään pitämään käyttäjä mahdollisimman paljon samassa näkymässä, jottei käyttäjän tarvitse turhaan etsiä eri ominaisuuksia, vaan ominaisuudet löytyvät kaikki samasta näkymästä.

4 TOTEUTUS

Työkalun suunnitelman valmistuttua aloitettiin itse työkalun rakennus. Heti ensimmäisissä vaiheissa toteutuksen aloittamisesta huomattiin, että aika-arviot eri vaiheisiin olivat arvioitu hieman liian matalalle. Projekti eteni muuten suunnitelman mukaan. Puolessa välissä projektia kuitenkin yrityksen kanssa väli-palaverissa sovittiin yksi päivä lisää aikaa projektin toteutukselle, jotta olisi varmasti tarpeeksi aikaa saada projekti valmiiksi.

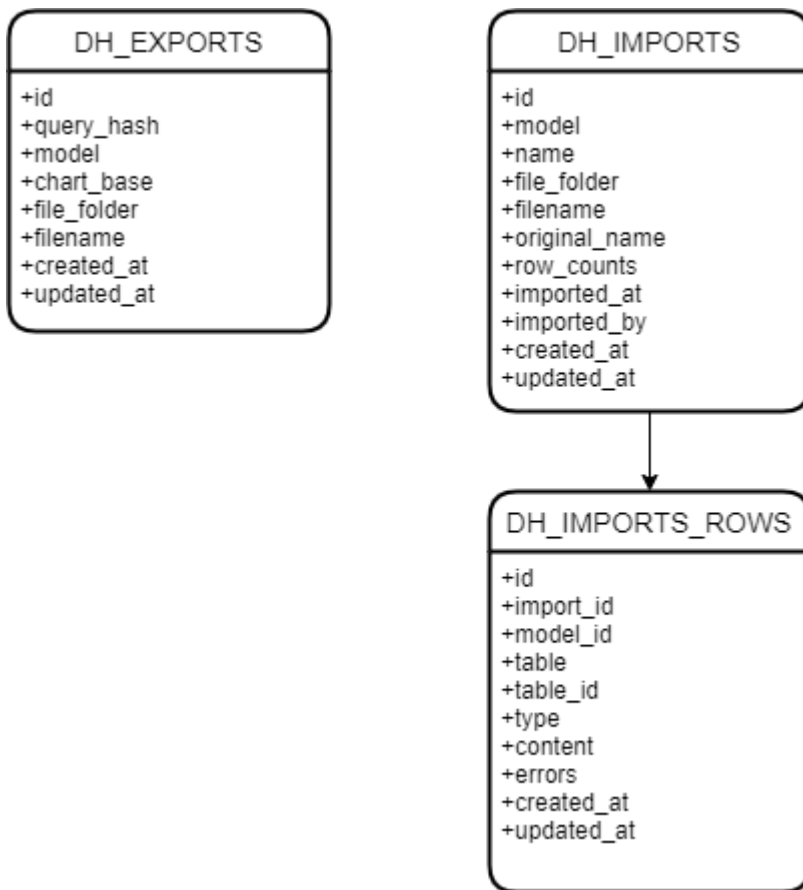
Vähän projektin puolenvälin jälkeen keksittiin parannusehdotuksia työkalulle, joita ei ollut suunnittelu-vaiheessa keksitty. Käytiin hieman aikaa näiden tarkemmalle suunnittelemiselle ja näiden ominaisuuksien aika-arvioiden tekemiselle. Esitin nämä parannusehdotukset yritykselle ja ne hyväksyttiin, joten projekti suunnitelmaan lisättiin kaksi vaihetta kesken projektin ja näiden lisäämä työmäärä myös kasvatti projektin pituutta kahdella päivällä.

Toteutuksen valmistuttua pidettiin yrityksen kanssa palaveri, jossa esiteltiin valmis tuote. Sekä käytiin läpi ongelmia, joita ei osattu suunnittelussa ottaa huomioon. Palaverin tavoitteena oli saada projekti tuotanto kuntoon. Projektiin valittiin muutamia parannuksia ja yrityksen puolelta tulleita vaatimuksia. Näille vaiheille luotiin aika-arviot ja sovittiin yrityksen kanssa kahdesta lisäpäivästä projektille. Taito luoda aika-arvioita oli selkeästi kehittynyt projektin aikana, koska viimeiset lisätyt parannukset saatiin ajallaan valmistumaan. Tämän jälkeen työkalun koodi julkaistiin ensin muille kehittäjille, jotta se voitaisiin valmistella julkaistavaksi käyttäjille. (Lillhonga palaveri 2021b).

Sisäisessä julkaisussa huomattiin työkalun aiheuttavan suuria ongelmia koko ohjelmiston toiminnalle. Tämä johtui eroavaisuuksista kehitysympäristöissä. Jouduttiin tekemään versiopäivityksiä paketeille, joita ohjelmisto käyttää. Näiden ongelmien ratkettua julkaistiin työkalu käyttäjille.

4.1 Tietokantarakenne

Tietokantarakenne työkalulle on hyvin yksinkertainen. Tämä koostuu kolmesta eri tietokantataulusta ”DH_EXPORTS”, ”DH_IMPORTS” ja ”DH_IMPORTS_ROWS”. Kuvassa 7 havainnollistetaan tietokantarakennetta. Työkalu on rakennettu tietokannan osalta riippumattomaksi muusta osasta tietokantaa.



KUVA 7. Tietokantarakenne.

Kuvassa 7 esitelty ”DH_EXPORTS”-tauluun tallennetaan tieto kaikesta datasta, joka otetaan ulos järjestelmästä. Työkalu generoidessaan laskentataulukkoa tallentaa kyseisen taulukon myös palvelimelle. Tämän jälkeen palvelin tarjoilee tiedoston käyttäjälle. Tästä syystä tauluun tallennetaan ”file_folder”- ja ”filename”-kenttiin generoidun tiedoston nimi ja sen tallennuspaikka. Lisäksi taulussa on mielenkiintoinen kenttä nimeltä ”query_hash”. Tähän kenttään tallennetaan Tirkistelijä-hakutyökalun generoima hakuparametri, jolla voidaan toistaa suoritettu haku. Tauluun tallennetaan näiden tietojen lisäksi myös ”model”-kenttään tieto siitä, minkä tyyppisestä objektista taulukko on generoitu. Tämä on mukana siksi, että työkalun toimintaa on tarkoitus tulevaisuudessa laajentaa tuotteiden ulkopuolelle. Lisäksi kenttä ”chart_base” määrittelee sen, tulostetaanko tyhjä laskentataulukko, vai taulukko täynnä dataa.

Kuvassa 7 voidaan myös nähdä taulu ”DH_IMPORTS”. Tähän tauluun tallennetaan tiedot jokaisesta tallennuksesta järjestelmään. Laskentataulukko tallennetaan palvelimelle myös tässä tapauksessa ja samalla tavalla tallennetaan tiedoston sijainti tietokantaan. Jokaiselle tiedostolle myös generoidaan uusi uniikki nimi palvelimelle tallennuksen yhteydessä, tästä syystä on myös kenttä ”original_name”, johon

tallennetaan tiedoston alkuperäinen nimi. Kenttään "row_counts" tallennetaan sarjallisoitu PHP-taulukko, joka sisältää lukumäärät eri tauluihin koostuvista muutoksista. Lisäksi tauluun tallennetaan myös tieto siitä, koska itse tallennus on suoritettu ja kenen toimesta. Nämä tiedot tallentuvat "imported_at"- ja "imported_by"-kenttiin. Kenttä "imported_by" viittaa "users"-tauluun, johon tallennetaan ohjelmiston käyttäjien tiedot. Tätä taulua ei piirroksessa kuitenkaan näy, koska sen toiminta ei ole muuten oleellista työkalun kannalta.

Näiden taulujen lisäksi kuvassa 7 voidaan nähdä myös taulu "DH_IMPORTS_ROWS". Tämä taulu sisältää tiedot jokaisesta yksittäisestä muutoksesta, joka on tehty tai aiotaan tehdä työkalun avulla. Tässä taulussa tärkeimpiä kenttiä on "import_id"-kenttä, joka viittaa "DH_IMPORTS"-tauluun. Näin ollen "DH_IMPORTS"- ja "DH_IMPORTS_ROWS"-taulujen relaatio on "yksi useaan" (eng. has many). Riville tallennetaan myös "model_id"-kenttään "DH_IMPORTS"-taulussa tallennetun "model"-kentän objektin tarkka id. Näin voidaan linkittää tämä muutos toisen tietokantataulun yksittäiseen riviin. Tapauksessa, jolloin meillä ei ole käytettävissä tietyn tyylistä objektia, vaan tallennetaan yksittäistä dataa johonkin tauluun, on käytössä kentät "table" ja "table_id". Näillä voidaan myös löytää rivi, johon tehtyjä muutoksia tässä kuvataan.

Tilanne, jossa meillä ei ole tietyn tyylistä objektia, vaan vain tietokantataulun nimi tulee vastaan, kun tallennetaan tai tarkistetaan erityisiä kenttiä, kuten "stock:free_stock-1"-kenttä, joka on kuvattu taulukossa 1. Kenttään "type" tallennetaan tieto siitä, kuvaako rivin muutokset luomistapahtumaa, vai päivitystapahtumaa. Puolestaan kenttään "content" tallennetaan tieto muutoksista, mitä data oli ennen ja mitä se oli jälkeen tapahtuman. Kenttään "errors" tallennetaan tieto virheistä, joita sattuisi tämän kentän muokkauksen yhteydessä. Mikäli "errors"-kentässä kuitenkin on jotain dataa, voidaan olettaa, että validointi on löytänyt virheen ja tallentaminen estetään käyttöliittymässä.

4.2 Datan hallinta

Datan hallinta yleisesti on yksi tärkeimpiä osapuolia ERP-ohjelmistoa. Lisäksi datan muokkaaminen suurissa määrin on aina toimenpide, johon sisältyy riskejä. Isoja määriä operaatioita tehdessä on suuri mahdollisuus käyttäjän virheisiin, lisäksi vaarana on myös ohjelmistossa esiintyvät tuntemattomat virheet, jotka voivat aiheuttaa odottamattomia seurauksia.

The screenshot shows the REFOX data management interface. On the left is a navigation menu with items like Myynti, Osto, Kassa, Tuotteet, Hinnastot, Tuotteenarvio, Aikakäsit, Toinnit, Ropo, Verkkokauppa, Raportit, Alueaset, Tilat, and Menut. The main area is titled 'Datan hallintatyökalu' and contains several sections:

- Yleiset**: Fields for 'Tuonnin nimi' and 'Tuodun excelin nimi'.
- Virheet**: A table with columns 'Tyyppi', 'Kohde', and 'Virhe'. It shows one error: 'Virhe' with the message 'Kenttä "numero" on vaadittu.'
- Tavoitellut muutokset**: A table with columns 'Luvut/päivitykset', 'Kohde', 'Kerätyt', 'Vanha arvo', and 'Uusi arvo'. It shows one change: 'Virhe' with 'Tuote' 'Myyntihinta' changing from '8,51' to '1'.
- Lataa excel**: A section for uploading files, with a 'Lataa' button.
- Tallenna excel**: A section for saving files, with a 'Tallenna' button.
- Edelliset tuonnit**: A table showing previous imports with columns 'Tuonnin nimi', 'Tiedoston nimi', 'Tarkistettu', 'Virheiden', 'Tuotto', 'Tuotje', and 'Tuonti aika'. It shows one import: 'products-1041280081.xlsx' with 'KYLÄ' and 'ET'.

KUVA 8. Työkalun virhenäkymä, muutosten kanssa.

Näitä yritetään työkalussa minimoida jakamalla datan tarkistus ja tallentaminen omiin vaiheisiin. Sekä näyttämällä käyttäjälle muutokset ja vaatimalla niiden hyväksymistä ennen tallennusta. Ohjelmiston sisäisiä virheitä ei voida kuitenkaan ennakoida muuta kuin testaamalla ja pitämällä logiikka mahdollisimman yksinkertaisena. Kuvassa 8 voidaan nähdä tilanne, jossa on tarkistettu laskentataulukko. Taulukossa on kuitenkin löytynyt muutosten lisäksi myös virheitä. Virheet otsikon alla voimme havaita, että työkalu on löytänyt yhden virheen ”numero”-kentästä. Tämä ei ollut kuitenkaan taulukossa esitelty ainoa muutos, vaan lisäksi myyntihintaa on muokattu.

4.2.1 Vastaanotto

Dataa voidaan työkaluun tuoda erityisessä muodossa olevalla laskentataulukolla. Helpoin tapa saada laskentataulukko oikeaan muotoon onkin ladata jo oikeassa muodossa oleva laskentataulukko työkalusta. Näin ei tarvitse käyttäjän itse huolehtia otsikoiden oikeasta muodosta tai muista ominaisuuksista, joita laskentataulukossa voisi käyttäjä muokata. Työkalu vaatii tarkan rakenteen syötettävälle laskentataulukolle, määrittelemällä laskentataulukon rakenne tarkkaan voidaan vähentää virheitä, sekä selkeyttää koodin logiikkaa.

	A	B	C	D	E	F	G	H	I	J
1	Id	Tuotenumero	Tyyppi	Aktiivinen	Järjestys	Ei-fyysinen	Myyntihinta	Myyntihinnan alennusprosentti	Ostohinta	Valuutta-numero
2										
3	id	number	type	active	order	immaterial	sales_price	sales_price_discount_percent	purchase_price	purchase_price_currency_id
4	1	A1	AB1	1	1	1	10	0	3	1
5	2	A2	AB2	1	2	0	15	0	6	1
6	3	A3	AB3	1	3	0	20	10	12	1
7										
8										
9										
10										

KUVA 9. Työkalun generoima laskentataulukko.

Kuvassa 9 nähdään työkalun generoima laskentataulukko, joka on avattu Microsoftin Excel-ohjelmistolla. Kuvassa myös esimerkki dataa testiympäristöstä havainnollistamassa laskentataulukon rakennetta. Mikäli laskentataulukosta tahdottaisiin muokata, jotain tiettyä osaa, voitaisiin oikean otsikon alta muokata haluamaansa kenttää, ja tämä jälkeen tallentaa muutokset Refox:iin.

4.2.2 Validointi

Datan validointi on keskeinen osa työkalua. Validointi toteutetaan hyväksikäyttäen Laravel-ohjelmistokehityksen omaa validointi työkalua. Tästä työkalusta luotiin oma versio, joka osaa toimia paremmin olemassa olevan ohjelmiston kanssa. Itse luotu validointityökalu osaa generoida tietokantataulun perusteella perussäännöt jokaiselle datakentälle. Näin voidaan varmistaa, että käyttäjä ei vahingossa voi syöttää dataa, jota olisi mahdotonta tallentaa.

Näiden automaattisten validointi sääntöjen lisäksi tarvitaan vielä sääntöjä, jotka ovat ohjelmiston toimintaan keskeisiä ja joita ei pystytä generoimaan. Nämä säännöt on pakko luoda käsin, mutta ohjelmistokehityksen tarjoamalla validointityökalulla on näiden sääntöjen luomiseen helpotetut tavat. Käytännössä jokaiselle kentälle, joka tarvitsee erikoisääntöjä, päädyttiin luomaan käsin tarkemmat säännöt.

Ongelmat ratkaistiin luomalla yksittäisille kentille käsin kirjoitetut omat funktiot, joilla data pystyttiin validoimaan ja tallentamaan. Mikäli kokonainen toinen taulu oli liitettyinä laskentataulukkoon, oli sen data helpompi validoida ja tallentaa, koska niitä voitiin käsitellä erillään muusta datasta. Tästä syystä ne eivät vaatineet erityistä funktiota tallentamista tai validointia varten.

4.2.3 Tallentaminen

Datan tallentaminen toimii samalla logiikalla, kuin validointi. Eli ensimmäisenä laskentataulukko luetaan ja sen sisältämä data ryhmitellään. Tämän jälkeen kaikki data käydään läpi ja validoinnin sijaan tallennetaan. Jokaisesta tallennusoperaatiosta otetaan talteen kaikki vanhat arvot, sekä muuttuneet arvot. Näin voidaan käyttäjälle näyttää tehdyt muutokset antaa myös käyttäjälle mahdollisuus palauttaa tekemiänsä muutoksia.

4.3 Virheidenhallinta

Mikäli datan validointi ei mene läpi, täytyy tästä ilmoittaa käyttäjälle ja pyrkiä neuvomaan tätä, kuinka korjata virheet, jotta hän pääsisi tekemään tahtomansa muutokset. Virheiden hallinta on myös tärkeää, koska monella ohjelmiston käyttäjällä on virheellistä dataa tietokannassa, joten voi olla mahdollista, että työkalun näyttämät virheet eivät ole peräisin käyttäjän itse tekemästä virheestä, vaan virheestä, joka on päässyt järjestelmään jonkin muun ohjelmiston ominaisuuden kautta. Korjaaminen kuitenkin jää käyttäjän vastuulle, joten pyritään neuvomaan käyttäjää tässä mahdollisimman hyvin.

Validointi sääntöjen perusteella ohjelmistokehityksen tarjoama validointityökalu generoi selko kielisiä virheilmoituksia, ja nämä voidaan suoraan näyttää käyttäjälle. Näin voidaan käyttäjälle ilmoittaa, että millä rivillä laskentataulukko on virhe, sekä missä kentässä virhe esiintyy. Virheet tallennetaan myös työkalun toimesta tietokantaan, joten niitä voidaan tarkastella myös myöhemmin. Mikäli kyseinen käyttäjä ei osaisi korjata virhettä tai itse virheilmoitus on virheellinen. Näin ollen käyttäjän esimies tai yrityksen asiakaspalvelu pääsee näkemään virhetilan ja laskentataulukon, joka aiheutti virheen ja se voidaan korjata.

Virheiden näyttämiseen käyttäjälle rakennettiin oma näkymä, joka ryhmittelee virheet, sekä ilmoittaa mitä virheitä laskentataulukosta löytyy. Validointi sääntöjen perusteella käyttäjälle myös ilmoitetaan, miksi annettu data on virheellistä. Esimerkiksi tuotenumero voi sisältää numeroita ja/tai kirjaimia. Tämän merkkijonon pituus on kuitenkin rajoitettu 50 merkkiin. Mikäli käyttäjä yrittäisi tallentaa 60 merkkiä pitkän merkkijonon, tulisi käyttäjälle virheilmoitus ”’Tuotenumero’ ei voi olla yli 50 merkkiä pitkä”. Näkymässä myös ryhmitellään virheet tuotekohtaisesti, jotta käyttäjä osaa korjata oikean rivin laskentataulukosta.

4.4 Vahinkojenhallinta

Työkalulla on mahdollista tehdä massiivisia muutoksia tietokantaan, joten sen luoma vaara virheille on todellinen. Mikäli virheellistä dataa pääsee tallentumaan työkalun kautta, on mahdollista, että osa järjestelmästä lakkaa toimimasta oikein. Näin se voi välillisesti aiheuttaa suuriakin tappioita käyttäjän yrityksen liiketoiminnalle.

Vahinkojen pääasiallinen estäminen tapahtuu validoimalla data tarkasti, kuten aiemmin on jo mainittu. Työkalun logiikkavirheitä pyritään estämään testauksella. Työkalun perustoimintaan onkin rakennettu testejä, joilla voidaan varmistaa validoinnin toimivuus. Validoinnissa käytettäviä sääntöjä täytyy ohjelmiston kehittyessä muokata ja parannella. Validointi sääntöjen tarkkuus heijastuu suoraan työkalun toimintavarmuuteen.

Tärkeänä osana vahinkojen estämistä on myös käyttäjältä hyväksynnän vaatiminen ennen tallentamista. Kun tarkistuksen jälkeen jo näytetään käyttäjälle muutokset, jotka aiotaan tehdä, sekä vaaditaan käyttäjää tallentamaan tiedot. Näin saadaan käyttäjä varmistamaan aikomansa muutokset ja suurimmat vahingot paljastuisivat jo ennen kuin mitään vahinkoa on sattunut.

Työkalu tallentaa kaikki tekemänsä muutokset, joten vahingon sattuessa on käyttäjän itse mahdollista käydä palauttamassa muutoksia, mikäli käyttäjä kuitenkin on epävarma ja ottaa asiakaspalveluun yhteyttä on myös mahdollista yrityksen käydä tarkistamassa virheiden tila ja korjata ne tallennetun tiedon perusteella. Historia tiedoilla ei ole takarajaa, joten käyttäjät voivat käydä tarkistamassa vaikka ensimmäisiä muutoksia, joita ovat tehneet. Näissä näkyvät arvot ennen muutosta, sekä muutoksen jälkeen, vaikka kyseisiä tietueita olisi tämänkin jälkeen muokattu.

4.5 Tapahtumien kirjaaminen

Kuten aikaisemmin on mainittu, työkalu tallentaa kaikki tapahtumat ylös, jotta mahdolliset virheet ovat jäljitettävissä. Tämä on tärkeää siksi että virheen sattuessa voidaan yrittää keksiä tapa välttää kyseinen virhe tulevaisuudessa ja näin ollen parantaa työkalun luotettavuutta. Lisäksi työkalun tallentamat muutokset voivat olla myös tärkeä osa käyttäjien omaa virheiden etsintää. Jos jokin arvo on muuttunut, työkalulla voidaan käydä historia tiedoista muutokset tarkistamassa. Näin voi käyttäjät tarkistaa kuinka kauan virhe on ollut järjestelmässä.

4.6 Integrointi

Integrointi on tärkeä osuus työkalua, koska se tulee osaksi olemassa olevaa järjestelmää. Tärkeimpänä osana integrointia on hakutyökalun integrointi. Oman haun toteuttaminen olisi lisännyt monta päivää kehitystyötä projektiin, lisäksi se olisi lisännyt teknologista velkaa ja aiheuttanut ihmetystä käyttäjillä. Integrointi muuhun järjestelmään toimii pääosin käyttämällä samaa tietokantaa muokkauksiin, joten eriliselle perinteiselle integraatiolle ei ole tarvetta.

4.7 Käyttöliittymä

Käyttöliittymän rakentamiseen otettiin vahvasti mallia muualta ohjelmistosta. Näin saatiin työkalu visuaalisesti näyttämään siltä, että se kuuluu järjestelmään. Lisäksi käyttöliittymässä otettiin huomioon myös mobiililaitteet, joita useat käyttäjät käyttävät. Käyttöliittymästä rakennettiin myös mahdollisimman intuitiivinen ja siihen lisättiin lyhyitä ohjeita ja viitteitä kirjalliseen ohjeeseen, mikäli käyttäjä jää jumiin työkalun käytössä.

The screenshot shows the NREFOX data management interface. On the left is a dark sidebar with navigation options like 'Myynti', 'Ostot', 'Kassa', etc. The main area is titled 'Datan hallintatyökalu' and contains three main sections:

- Lataa excel:** A section for downloading Excel files, with a 'Lataa' button and a 'Tallenna excel' section below it.
- Aputaulut:** A table of auxiliary tables with columns: Numero, Veroprosentti, IBAN, BIC, Näytetty laskulla, and Nimi. It shows a list of tables like 'Perustamismenot', 'Tutkimusmenot', etc.
- Edelliset tuonnit:** A table of previous imports with columns: Tuonnin nimi, Tiedoston nimi, Tarkistettu, Virheetön, Tuotu, Tuojä, Tuontiaika, and a 'Näytä virheet' button for each row.

KUVA 10. Datan hallintatyökalun perusnäkökulma.

Kuvassa 10 näkyy työkalun aloitusnäkökulma. Tässä näkyy vasemmassa yläkulmassa otsikon ”Lataa excel” alla valikko, josta voidaan ladata laskentataulukko käyttäjän muokattavaksi. Tämän oikealla puolella

näkyä aputaulut, joista löytyy tukevaa dataa eri tietojen muokkaamista varten. Kuvassa on ”Tilit” väli-lehti näkyvässä, josta voidaan tarkistaa eri tilien numerot. ”Tallenna excel” otsikon alla on mahdollisuus ladata laskentataulukko työkaluun tarkistettavaksi ja tallennettavaksi. Jokaiselle tallennukselle täytyy valita oikea tyyppi, jotta työkalu osaa käsitellä laskentataulukkoa oikein.

KUVA 11. Datan hallintatyökalun virhenäkymä.

Kuvassa 11 puolestaan voidaan nähdä tilanne, jossa on yritetty syöttää virheellistä dataa työkalun avulla. Sääntöjen mukaan jokaisella tuotteella kuuluu olla tuotenumero, ja tässä esimerkissä on yritetty syöttää laskentataulukko, josta on tuotenumero poistettu. Tästä syystä työkalu näyttää virheen ”Kenttä ’numero’ vaadittu”. Tämän rivin yläpuoleisella rivillä voimme nähdä linkin tuotekortille ja tätä kautta käyttäjä löytää oikean virhe rivin myös laskentataulukosta. Virheiden alapuolella näkyy myös ”Tavoitellut muutokset” otsikon alla taulukko, joka on esimerkissä tyhjä. Tämä johtuu siitä, ettei mitään muita muutoksia ole tehty laskentataulukkoon, kuin poistettu tuotenumero virheellisesti.

The screenshot displays the REFOX data management interface. The main content area is titled "Tarkistuksen yhteydessä havaitut muutokset" (Changes detected during the check). It contains several sections:

- Tallenna excel**: A button to save the current view as an Excel file.
- Tarkistuksen yhteydessä havaitut muutokset**: A table showing the results of the check.

Kohde	Löydettyjä rivejä excelistä	Olemassa olevia rivejä	Ei muuteta	Muokataan	Uusia
Tuote	1	1	0	1	0
Tuotteen saldotiedot	4	4	4	0	0
Tuotteen kieliversiöt	4	4	4	0	0
- Tarkennus havaituista muutoksista**: A table detailing the changes made to the product's discount rate.

Luotu/päivitetty	Kohde	Kontti	Vanha arvo	Uusi arvo
/product/product/343	Tuote	Alennushinta	0	2
- Lataa excel**: A button to download the data as an Excel file.
- Aputaulut**: A section for auxiliary tables, currently showing a list of IBANs and BICs for various products.

Numero	Veroprosentti	IBAN	BIC	Näytetty laskulla	Nimi
1001	0			0	Perustamismenot
1011	0			0	Turkummenot
1021	0			0	Tuotekäyttökulut
1031	0			0	Päivät
1081	0			0	Tekijänoikeudet
1085	0			0	Tavarainmaksu-oikeudet
1027	0			0	Lisensimaksut
1029	0			0	Taloustieteiden
1041	0			0	Atk-ohjelmien lisenssimaksut
1043	0			0	Puhelinlittymät
- Edelliset tuonnit**: A table showing the previous import records for the product.

Tuonnin nimi	Tiedoston nimi	Tarkistettu	Virheiden	Tuota	Tuota	Tuonti aika
	products-1641280081.xlsx	KYLLÄ	ET	ET		
	products-16412872811.xlsx	KYLLÄ	ET	ET		

Kuva 12. Datan hallintatyökalu onnistuneen tarkastuksen näkymä.

Kuvassa 12 näemme tilanteen, jossa kuvassa 11 näkyvät virheet on korjattu laskentataulukosta. Lisäksi on tehty muutos tuotteen alennushintaan. Tämä tieto näkyy otsikon ”Tarkennus havaituista muutoksista” otsikon alapuolelta löytyvästä taulukosta. Aikaisemmin tuotteella ei ollut asetettuna alennushintaa, eli se oli 0. Nyt työkalu aikoo tehdä muutoksen ja asettaa alennushinnaksi 2. Tämän taulukon yläpuolella näkyy otsikon ”Tarkistuksen yhteydessä havaitut muutokset” alla taulukko, jossa on numeroarvoina ilmoitettu osa-alueet, joita aiotaan muokata. Kyseisessä laskentataulukossa on yksi tuote, jolla on neljä eri varastopaikkaa, siksi saldotietoja ilmoitetaan neljä kappaletta. Tuotteella on myös neljä eri kieliversiota, täten ilmoitetaan kieliversioiksi myös neljä. Nämä tuotteisiin liittyvien osa-alueiden muutoksia seurataan siis erikseen.

The screenshot displays the REFOX data management interface. The top navigation bar includes the REFOX logo and user icons. A sidebar on the left lists various system functions like 'Myynti', 'Ostot', 'Kassa', etc. The main content area is divided into several sections:

- Yleiset**: A summary table showing changes in the 'Excel' category.

Kohde	Löydetyksi riviä excelistä	Olemassa olevia riviä	Ei muuteta	Muokataan	Uusia
Tuote	1	1	0	1	0
Tuotteen kalsitoidot	4	4	4	0	0
Tuotteen kielsiedot	4	4	4	0	0
- Toteutetut muutokset**: A table showing implemented changes, including a price reduction for product 'products/produk3543'. A yellow button labeled 'Palauta kentän "Alennushinta" muutokset' is visible.
- Edelliset tuonnit**: A table listing previous import records with columns for 'Tuonnin nimi', 'Tiedoston nimi', 'Tarkistettu', 'Virheiden', 'Tuotu', 'Tuojä', and 'Tuonti aika'. The table shows several records for 'products-16412872811.xlsx' and one for 'testit'.

KUVA 13. Datan hallintatyökalu, tuonti tallennettu näkymä.

Kuvassa 13 nähdään näkymä, jossa ilmoitetaan tallennuksen onnistuminen. Tämä näkymä palautetaan käyttäjälle, kun se on vahvistanut laskentataulukossa ilmoitetut muutokset. Tässä esimerkissä siis tuotteen alennushintaa on muokattu. Näkymä on hyvin samankaltainen, kuin edellinen näkymä. Erona tässä on ”Yleiset” otsikon alla olevat tiedot, josta voidaan tarkistaa laskentataulukon tallentaja ja tallennus hetki. Lisäksi tallentajan itse valitsema nimi tuonnille. ”Toteutetut muutokset” otsikon alla voidaan nähdä muutokset, jotka on toteutettu. Näiden muutosten vierellä on myös keltainen nappi ”Palauta kentän ’Alennushinta’ muutokset”. Tätä nappia painamalla voidaan palauttaa kyseisen tuotteen kentän arvo takaisin alkuperäiseen arvoon.

5 JOHTOPÄÄTÖKSET

Työkalu saatiin rakennettua onnistuneesti ja se täyttää yrityksen asettamat vaatimukset. Projekti ei valmistunut alkuperäisen aikataulun mukaan, koska kesken projektin huomattiin asioita, joita tahdottiin työkaluun lisätä. Nämä kuitenkin otettiin huomioon jo kesken projektin ja aikataulu neuvoteltiin uusiksi. Aikataulun arviointi oli jatkuvaa koko projektin ajan. Näin yritys pysyi ajan tasalla projektin etenemisestä.

Työkalun tarkistussäännöissä on jatkokehityksen tarvetta. Projektissa ei esiintynyt suuria ongelmia. Itse projekti oli vaatimustasoltaan sopiva vastaamaan tekijän osaamistasoa. Haasteita löytyi kuitenkin, kuten yleensäkin projekteissa. Näihin haasteisiin kuitenkin löytyi ratkaisu suunnitelmassa varatun ajan sisällä. Projektin teko vaiheessa ilmeni kuitenkin ongelma, joka olisi voitu välttää. Rakentaessa validointi ja tallennus logiikkaa, nämä rakennettiin alun perin erillisiksi osiksi. Lisäksi näiden tulokset tallennettiin eri tietokantatauluihin. Työkalusta oli tällä tavalla toteutettuna olemassa jo toimiva versio. Huomattiin vasta projektin loppupuolella, että näiden kahden logiikan erottaminen ei olisi järkevää. Koska logiikat toimivat erillään tarvittiin paljon ylimääräistä koodia pitämään käyttöliittymä samannäköisenä siirryttäessä validointivaiheesta tallentamisvaiheeseen.

Tämä ongelma saatiin ratkaistua yhdistämällä nämä logiikat. Logiikat voitiin yhdistää, koska ennen validointi päätöstä tai tallennusta täytyy laskentataulukon data parsia. Sekä täytyy tallentaa logiikan läpikäynnin yhteydessä syntyvät historia tiedot. Tämä aiheutti lisätyötä, mikä oli kuitenkin tarpeellista tehdä ennen projektin saattamista ensimmäiseen versioon. Logiikoiden yhdistäminen vaati työtä, koska ne oli kirjoitettu eri aikoina ja poikkesivat toisistaan jonkin verran. Lisäksi tallennettavat tiedot olivat eri tietokantatauluissa ja tämä vaati koko tietokantarakenteen uudelleen suunnittelua. Logiikat kuitenkin yhdistettiin ja tämän muutoksen avulla tuli koodista huomattavasti yksinkertaisempaa ja ylläpidettävämpää.

Validointi- ja tallennuslogiikoiden erottamisen ongelmat olisi voitu välttää, mikäli olisi kyseinen ominaisuus suunniteltu tarkemmin. Kokonaisuudessaan ohjelmistoprojektia suunnitellessa pitäisi löytää sopiva raja siinä, kuinka pitkälle suunnitteleminen kannattaa toteuttaa. Tässä tapauksessa projektin suunnittelu oli muilta osin sopivalla tasolla, mutta tätä yhtä ominaisuutta ei ollut osattu suunnitella tarpeeksi syvällisesti.

Aikataulun arviointi onnistui pääosin hyvin. Projektissa ilmeni kohtia, joissa aika-arvio oli alakanttiin ja vaadittiin lisää aikaa suorittamiselle. Mutta projektissa oli myös kohtia, joihin oli arvioitu ylimääräistä aikaa. Nämä eri kohdat tasoittivat hyvin toisiaan ja projekti valmistui suunnitelman mukaan.

Ohjelmisto projektiin kuuluu aina itse koodin kirjoittamisen lisäksi myös erilaisten dokumenttien kirjoittamista. Työkalu, joka rakennetaan lisäksi ohjelmaan, jolla on paljon käyttäjiä ja useita kehittäjiä, vaatii tarkan dokumentoinnin. Dokumentointia tarvitaan työkalun käyttäjille, sekä yrityksen sisäiseen dokumentaatioon lisättäväksi. Yrityksen sisäinen dokumentaatio on yhtä tärkeä, kuin loppukäyttäjälle toimitettava ohjekirja. Laajalla sisäisellä dokumentaatiolla voidaan nopeuttaa muiden kehittäjien osallistumista työkalun lisä ominaisuuksien kirjoittamiseen tai virheiden korjaamiseen. Sisäinen dokumentaatio kirjoitettiin yrityksen muiden dokumenttien jatkoiksi. Tämä dokumentaatio sisältää tarkempia kuvauksia koodin toiminnasta. Kuten eri tiedostojen kuvauksia sekä sijainteja. Tähän dokumenttiin on kirjattu ylös työkalun toiminta kehittäjiä tukevalla tavalla.

Työkalun käyttöliittymästä rakennettiin mahdollisimman intuitiivinen ja käyttöliittymään upotettiin ohjeita. Tämä ei kuitenkaan riitä näin laajassa projektissa. Käyttäjille on tärkeää, että ongelma tilanteessa on kirjoitettu ohje, johon he voivat tukeutua ennen asiakaspalveluun soittamista. Mikäli kirjoitettua ohjetta ei ole, voisi asiakaspalvelu ruuhkautua turhaan kysymyksillä, joihin on tiedossa helppo vastaus. Ohjeiden olemassaolo myös lisää käyttäjän itsevarmuutta käyttää työkalua, koska käyttäjä voi saada kirjallisen vahvistuksen tekevänsä oikein. Lisäksi ohjeet vähentävät virheiden syntyä sillä oletuksella, että käyttäjät lukevat ohjeet ennen kuin käyttävät työkalua.

Työkalu on ensimmäisessä versiossaan. Tämä tarkoittaa, että työkalu toimii ja on hyödyllinen, mutta käyttäjien palautteesta siihen tullaan tekemään jatkokehitystä. Sisäisiä ideoita jatkokehitykselle syntyi jo projektin teko vaiheessa, mutta näitä ei ole tarpeellista suorittaa, mikäli työkalu täyttää käyttäjien asettamat tarpeet.

6 LÄHTEET

Awati, R. 2021. Uniform Resource Identifier (URI). 2021. Saatavissa: <https://whatis.techtarget.com/definition/URI-Uniform-Resource-Identifier>. Viitattu: 25.03.2022.

Javatpoint.com. 2021. Compilation process in c. Saatavissa: <https://www.javatpoint.com/compilation-process-in-c>. Viitattu: 25.03.2022.

Kolade, C. 2021. What is PHP? The PHP Programming Language Meaning Explained. Saatavissa: <https://www.freecodecamp.org/news/what-is-php-the-php-programming-language-meaning-explained/>. Viitattu: 25.03.2022.

Laravel 2022a. Routing. Saatavissa: <https://laravel.com/docs/9.x/routing>. Viitattu: 25.03.2022.

Laravel 2022b. HTTP Session. Saatavissa: <https://laravel.com/docs/9.x/session>. Viitattu: 25.03.2022.

Laravel 2022c. Cache. Saatavissa: <https://laravel.com/docs/9.x/cache>. Viitattu: 25.03.2022.

Lillhonga palaveri 2021a. Pidetty 04.11.2021.
Osallistujat: Anton Saarenpää, Petter Lillhonga.

Lillhonga palaveri 2021b. Pidetty 21.12.2021.
Osallistujat: Anton Saarenpää, Petter Lillhonga.

Lillhonga. 2022. Refox. Saatavissa: <https://lillhonga.fi>. Viitattu 24.03.2022.

Mozilla. 2022. HTML basics. Saatavissa: https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/HTML_basics. Viitattu: 03.02.2022.

PHP.net. 2022. What is PHP?. Saatavissa: <https://www.php.net/manual/en/intro-whatis.php>. Viitattu: 03.02.2022.

Refox. 2022. Tarinamme. Saatavissa: <http://refox.fi>. Viitattu 24.03.2022.

Tutorialspoint.com. 2021. Laravel – Overview. Saatavissa: https://www.tutorialspoint.com/laravel/laravel_overview.htm. Viitattu: 25.03.2022.

Varusteleka. 2022. Varustelekan esittely. Saatavissa: <https://www.varusteleka.fi/fi/page/yritysesittely/19791>. Viitattu 25.03.2022.

W3Techs.com. 2022. Usage statistics of PHP for websites. Saatavissa: <https://w3techs.com/technologies/details/pl-php>. Viitattu. 03.02.2022.