



Karelia-ammattikorkeakoulu  
Tietojenkäsittelyn koulutus

# Modulaaristen käyttäjä- ja moduulikonfiguraatioiden luonti Odoo-toiminnanohjausjärjestelmässä

Arttu Tapio

Opinnäytetyö, huhtikuu 2022

[www.karelia.fi](http://www.karelia.fi)



**Karelia**  
AMMATTIKORKEAKOULU

**OPINNÄYTETYÖ**  
**Huhtikuu 2022**  
**Tietojenkäsittelyn koulutus**

Tikkarinne 9  
80200 JOENSUU  
+358 13 260 600 (vaihde)

**Tekijä**  
Arttu Tapio

**Nimeke**  
Modulaaristen käyttäjä- ja moduulikonfiguraatioiden luonti Odoo-toiminnanohjausjärjestelmässä

**Toimeksiantaja**  
Collapick Company Oy

Opinnäytetyön aiheena on luoda käyttäjä- ja moduulikonfiguraatioita useisiin Odoo-toiminnanohjausjärjestelmässä käytettäviin moduuleihin. Työn tarkoituksena oli toteuttaa modulaarisesti toimivat käyttäjä- ja moduulikonfiguraatiot Collapick Company Oy:lle. Niiden avulla toimeksiantaja pystyy hallinnoimaan käyttöoikeuksia ja asetuksia paremmin asiakkaille tarjoamissaan Odoo-ympäristöissä.

Opinnäytetyö oli toiminnallinen. Työn toteutus koostui kolmesta vaiheesta. Ensimmäinen vaihe oli suunnitteluvaihe, jossa tutustuttiin tarkemmin Odoo-toiminnanohjausjärjestelmään ja sen rajoituksiin sekä mahdollisuuksiin. Toinen vaihe oli kehitysvaihe, jossa tapahtui käyttäjä- ja moduulikonfiguraatioiden kehitys. Kolmas vaihe oli viimeistelyvaihe, jossa tuotetulle kokonaisuudelle tehtiin käyttöönotto toimeksiantajan Odoo-ympäristöihin. Työn tuotokselle tehtiin kattavat testaukset ennen työn käyttöönottoa. Testaus suoritettiin sekä moduulikohtaisesta että eri moduulien yhdistelmillä.

Opinnäytetyön tuloksena valmistui yksi uusi moduuli sekä kattavat muutokset ja lisäykset viiteen moduuliin. Toteutetut konfiguraatiot mahdollistavat toimeksiantajan nykyisten ja tulevien moduulien joustavan käytön. Odoo-toiminnanohjausjärjestelmän käyttöoikeuskonfiguraatiot ovat tärkeässä roolissa koko järjestelmän toiminnan kannalta ja niiden toteutuksessa on oltava huolellinen toimintaympäristön modulaarisuudesta johtuen.

**Kieli**  
suomi

Sivuja 40  
Liitteet 3  
Liitesivumäärä 3

**Asiasanat**  
Odoo, toiminnanohjausjärjestelmät, modulaarisuus, konfigurointi



**THESIS**  
**April 2022**  
**Degree Programme in Business Information Technology**

Tikkarinne 9  
80200 JOENSUU  
FINLAND  
+ 358 13 260 600 (switchboard)

Author  
Arttu Tapio

Title  
Creation of Modular User and Module Configurations in Odoo ERP

Commissioned by  
Collapick Company Oy

The topic of the thesis is to create user and module configurations to several modules used in Odoo enterprise resource planning system. The thesis was aimed at developing modular user and module configurations for Collapick Company Oy. They enable the client to better manage the access rights and settings in the Odoo environments they provide to their customers.

The thesis was practice-based. The implementation of the thesis consisted of three phases. The first phase was the design phase, in which the Odoo enterprise resource planning system and its limitations and possibilities were studied in detail. The second phase was the development phase, in which the development of the user and module configurations took place. The third phase was the finishing phase, in which the produced whole was deployed to the client's Odoo environments. Extensive testing was performed on the implementation of the thesis before its deployment. Testing was performed both module-specific and with combinations of different modules.

As a result of the thesis, one new module and comprehensive modifications and additions to five modules were produced. The implemented configurations enable flexible use of the client's current and future modules. The access rights configurations of the Odoo enterprise resource planning system act in an important role in the operation of the entire system, and care must be taken in their implementation due to the modularity of the operating environment.

Language  
Finnish

Pages 40  
Appendices 3  
Pages of Appendices 3

Keywords  
Odoo, enterprise resource planning, modularity, configuration

## Sisältö

|       |   |    |
|-------|---|----|
| 1     | Johdanto .....  | 6  |
| 2     | Lähtökohdat ja tavoitteet.....                              | 7  |
| 3     | Odoon ja sen toiminnanohjausjärjestelmä .....               | 8  |
| 3.1   | Odoon versiot ja ohjelmistoversiot .....                    | 9  |
| 3.2   | Modulaarisuus .....   | 10 |
| 3.3   | Arkkitehtuuri.....  | 12 |
| 4     | Helppokäyttöliittymät.....                                  | 13 |
| 4.1   | Ponniste-helppokäyttöliittymät .....                        | 14 |
| 4.2   | Collapick Tempo -helppokäyttöliittymät.....                 | 15 |
| 5     | Opinnäytetyössä käytetyt menetelmät ja aikataulu .....      | 17 |
| 5.1   | Lähestymistapa ja aikataulu.....                            | 17 |
| 5.2   | Käytetyt menetelmät ja työkalut .....                       | 18 |
| 5.2.1 | Odoon-toiminnanohjausjärjestelmä .....                      | 19 |
| 5.2.2 | Collapickin moduulit ja helppokäyttöliittymät.....          | 20 |
| 5.2.3 | Docker .....  | 20 |
| 5.2.4 | Amazon Web Services -pilvipalvelut ja versionhallinta ..... | 21 |
| 5.2.5 | Visual Studio Code ja laajennukset .....                    | 22 |
| 6     | Tuotoksen suunnittelu ja toteutus .....                     | 22 |
| 6.1   | Suunnitteluvaihe .....                                      | 23 |
| 6.2   | Kehitysvaihe .....  | 26 |
| 6.3   | Viimeistelyvaihe .....                                      | 32 |
| 7     | Lopullinen tuotos.....                                      | 33 |
| 7.1   | Collapick Base -moduuli .....                               | 33 |
| 7.2   | Viisi muuta Collapick-moduulia .....                        | 34 |
| 8     | Pohdinta.....   | 36 |
| 8.1   | Toteutus ja tulokset.....                                   | 36 |
| 8.2   | Oma oppiminen .....   | 37 |
| 8.3   | Jatkokehitysmahdollisuudet .....                            | 37 |
|       | Lähteet.....  | 39 |

### Liitteet

|         |   |
|---------|---|
| Liite 1 | Monitasoinen arkkitehtuuri visualisoituna           |
| Liite 2 | Odoon Community- ja Enterprise-version vertailu     |
| Liite 3 | Collapick Base -moduulin kansiorakenne ja tiedostot |

## Termit

|                   |  |
|-------------------|--|
| API               | Lyhenne sanoista Application Programming Interface eli edistynyt ohjelmointiliittymä, ohjelmointirajapinta, jota käytetään ohjelmoinnissa eri osien välisessä keskustelussa. |
| Backend           | Sovelluksen taustataso, jossa käyttäjien pyynnöt ja muut prosessit yleensä suoritetaan. Toimii usein palvelimella.   |
| CSS               | Tyylikieli, jota käytetään frontendissä HTML5:n yhteydessä määrittelemään, kuinka elementit näytetään.   |
| CSV               | Lyhenne sanoista Comma Separated Values, eli suomeksi pilkuilla erotellut arvot. Tiedostomuoto, jossa tekstiä erotellaan pilkuilla.  |
| ERP               | Lyhenne sanoista Enterprise Resource Planning, eli toiminnanohjausjärjestelmä.   |
| Frontend          | Sovelluksen esitystaso, jonka kanssa käyttäjät vuorovaikuttavat ja jossa heille tietoa esitetään. Käyttäjän näkökulmasta frontend on usein sama kuin käyttöliittymä.         |
| HTML5             | Merkintäkieli frontendille, jolla sen rakennetta voidaan määritellä. Lyhenne sanoista Hypertext Markup Language 5.   |
| JavaScript        | Ohjelmointikieli, tyypillisesti käytetty sekä front- ja backendissä toimintojen suorittamiseen.  |
| JSON              | Lyhenne sanoista JavaScript Object Notation, on tiedostomuoto jäsenllylle tiedolle.  |
| Käyttöliittymä    | Sovelluksen osa, jonka kautta käyttäjä sovellusta käyttää.   |
| Merkintäkieli     | Kieli, jonka avulla merkitään, miten tietoa näytetään.   |
| MVC-arkkitehtuuri | Ohjelmistoarkkitehtuuri, joka erottaa käyttöliittymän sovellustiedosta. Lyhenne MVC tulee sanoista Model-View-Controller eli suomeksi Malli-Näkymä-Käsittelijä.              |

|                    |   |
|--------------------|---|
| ORM                | Lyhenne sanoista Object Relational Mapping, eli objek-tisuhdekartoitus. Ohjelmointitekniikka, jolla oliopohjai-set ohjelmointikielet muuntavat tietoa.                                |
| Python             | Yleiskäyttöinen ohjelmointikieli.   |
| React              | React tai ReactJS on JavaScript pohjainen ohjelmointi-kieli frontendin käyttöliittymille.   |
| Relaatiotietokanta | Tietokanta, joka perustuu taulujen välisiin relaatioihin eli suhteisiin.  |
| RPC                | Lyhenne sanoista Remote Procedure Call eli etäprose-duurikutsu, on pyyntö-vastaus protokolla, jonka avulla ohjelma voi siirtää tietoa tai toteuttaa toimintoja toi-ssessa ohjelmassa. |
| SaaS               | Lyhenne sanoista Software as a Service, eli palvelun-tarjoaja ylläpitää ohjelmistoa asiakkaalle palveluna.  |
| SQL                | Lyhenne sanoista Structured Query Language, ohjel-mointikieli, jota käytetään yleensä tietokannoissa.   |
| Tyylikieli         | Kieli, jonka avulla tiedon esittämisen tyyliä ilmaistaan. Käytetään usein merkintäkielien yhteydessä.   |
| XML                | Lyhenne sanoista Extensible Markup Language, mer-kintäkieli sekä tiedostomuoto.   |

# 1 Johdanto

Opinnäytetyön aiheena on modulaaristen käyttäjä- ja moduulikonfiguraatioiden luonti Odoo-toiminnanohjausjärjestelmässä, johon on liitetty mobiiliyhteensopivia helppokäyttöliittymiä. Opinnäytetyön tietoperustassa perehdytään Odoo-toiminnanohjausjärjestelmään, sen modulaariseen ympäristöön ja siihen liitettäviin helppokäyttöliittymiin. Toiminnallisen opinnäytetyön lopputuloksena on yksi toiminnanohjausjärjestelmässä käytettävä moduuli sekä muissa moduuleissa käytettävää koodia, joilla konfiguraatioita saadaan muodostettua. Lisäksi opinnäytetyön tavoitteena on muodostaa toimintatavat, joilla samat konfiguraatiot saadaan lisättyä tulevaisuudessa mahdollisiin uusiin moduuleihin.

Opinnäytetyön toimeksiantajana toimii suomalainen ohjelmistoyritys, Collapick Company Oy, jolla on aiempaa kokemusta tuotannonohjauksesta sekä sitä varten luoduista helppokäyttöliittymistä. Yritys laajensi toimintaansa näistä järjestelmistä opinnäytetyössä aiheena toimivaan järjestelmään, yhdistäen siihen yrityksen aiemmin kehittämät helppokäyttöliittymät. Tämä muodosti tarpeen opinnäytetyössä tuotettaville konfiguraatioille ja ratkaisuille, joita on tarkoitus pystyä hyödyntämään opinnäytetyön valmistuttua sekä tulevaisuudessa niiden pohjalta konfiguraatioita laajentamalla. Erityistä huomiota opinnäytetyössä kiinnitettiin käyttöoikeuskonfiguraatioihin sekä toteutuksen yleiskäyttöisyyteen.

Opinnäytetyössä kerrotaan Odoo-toiminnanohjausjärjestelmästä, sen arkkitehtuurista, siihen liitettävistä helppokäyttöliittymistä sekä siitä, mitä rajoitteita ja vaatimuksia nämä opinnäytetyölle asettavat. Opinnäytetyön keskiössä olevasta Odoo-toiminnanohjausjärjestelmästä ei löydy montaa aiempaa ja ajankohtaista opinnäytetyötä, joten opinnäytetyö tuo alan toimijoille sekä opiskelijoille uutta tietoa kyseisestä järjestelmästä sekä sen toimintaympäristöstä.

## 2 Lähtökohdat ja tavoitteet

Opinnäytetyön toimeksiantajana toimii suomalainen Collapick Company Oy, tästä lähtien Collapick, joka on vuonna 2011 Joensuussa perustettu ohjelmistoyritys (Collapick Company Oy 2022a). Collapick on yksityisomistuksessa oleva osakeyhtiö, jonka pääkonttori sijaitsee Joensuussa ja sivukonttori Tampereella (Collapick Company Oy 2022b). Collapickin tärkeimpiä tuotteita ovat Collapick Tempo, Odoo-toiminnanohjausjärjestelmän toteutukset sekä verkkokauppojen toteutukset (Collapick Company Oy 2022a). Collapickilla työskentelee tällä hetkellä 11 työntekijää. Yrityksen toimitusjohtajana toimii Petteri Torssonen ja sen liikevaihto vuonna 2020 oli 506 tuhatta euroa. (Suomen Asiakastieto Oy 2022.)

Tarve opinnäytetyönä toteutettavalle koodille ilmeni, kun toimeksiantajana toimiva Collapick laajensi toimintaansa toiminnanohjausjärjestelmiin ja sen tuottamia helppokäyttöliittymiä alettiin liittää kiinni Odoo-toiminnanohjausjärjestelmään (Collapick Company Oy 2022a). Tästä muodostui tarve yleisille ja moduulikohtaisille konfiguraatioille sekä toimintatavoille vastaavien konfiguraatioiden muodostamiseksi uusiin moduuleihin tulevaisuudessa.

Haasteena opinnäytetyössä ovat Odoo-toiminnanohjausjärjestelmän modulaarinen ympäristö sekä sen kaksi eri versiota, Community ja Enterprise (Odoo S.A. 2022a). Ympäristöstä johtuen myös Collapickin toteuttamat moduulit ja helppokäyttöliittymät on jouduttu rakentamaan modulaarisesti, mikä tarkoittaa, että moduulit on konfiguroitava toimimaan irrallisina paloina sekä näistä paloista muodostuvina kokonaisuuksina. Kehitettyjen moduulien on toimittava jokaisella mahdollisella yhdistelmällä molemmissa Odoo-toiminnanohjausjärjestelmän versioissa, joten moduulien määrän kasvaessa potentiaalisten yhdistelmien määrä kasvaa eksponentiaalisesti. Samalla myös ongelmien ja virheiden mahdollisuudet kasvavat, joten toteutusta on testattava riittävästi ennen käyttöönottoa.

Opinnäytetyön tavoitteena on saada toteutettua yksi irrallinen moduuli, työnimeltään Collapick Base, joka toimii aina pohjana muille Collapickin moduuleille.



Tämän moduulin on tarkoitus sisältää käyttöoikeusryhmät Collapickin moduulien käyttöön, joilla määritellään käyttäjien oikeuksia helppokäyttöliittymissä sekä Odoo-toiminnanohjausjärjestelmän näkymissä (Odoo S.A. 2022b). Lisäksi tavoitteena on toteuttaa vastaavia konfigurointeja sekä käyttöoikeusryhmiä moduulikohtaisesti Collapickin viiteen suurimpaan moduuliin, jotka ovat työnimittäin seuraavat: Collapick Sale (eli Myynti), Collapick Purchase (eli Osto), Collapick MRP (eli Tuotanto), Collapick Stock (eli Varasto) ja Collapick Project (eli Projekti).

### **3 Odoo ja sen toiminnanohjausjärjestelmä**

Opinnäytetyön keskiössä oleva Odoo-toiminnanohjausjärjestelmä, tästä lähtien Odoo, on avoimen lähdekoodin ohjelmisto, jonka 30 ydinmoduulin kehityksestä ja lähdekoodin hallinnoinnista vastaa belgialainen yritys Odoo S.A. Odoolla on myös aktiivinen yhteisö, joka kehittää ilmaisia ja maksullisia moduuleja kaikille liiketoiminnan osa-alueille, jotka vastaavat yritysten kehittyviin tarpeisiin. (Odoo S.A. 2022c; Ganesh ym. 2016.) Lisäksi Odoon kehityksen ja yhteisön tukena toimii Odoo Community Association, OCA, joka tuottaa Odoohon ilmaisia moduuleja, joiden tarkoitus on laajentaa sekä parantaa Odoon toimintoja ja käytettävyyttä. Odoo Community Association on Odoo S.A.:sta irrallinen, voittoa tavoittelematon organisaatio, jonka kautta kehittäjät voivat yhteistyössä muiden kanssa panostaa Odoon kattavaan moduulitarjontaan. (Odoo Community Association 2022a.)

Odoon ensimmäisen version kehitti vuonna 2005 belgialainen Fabien Pinckaers, joka myös perusti Odoo S.A. -yrityksen sen silloisella nimellä OpenERP S.A. (Odoo S. A. 2022c; roxannevarza 2010). Pinckaers toimii nykyisin Odoo S.A.:n toimitusjohtajana (Odoo S.A. 2022c). Ohjelmiston ensimmäinen versio toimi nimellä TinyERP ja vuonna 2008 ohjelmiston nimeksi vaihdettiin OpenERP sekä yrityksen nimeksi OpenERP S.A., kunnes lopulta vuonna 2014 nimet vaihtuivat nykyiseen muotoonsa Odoo ja Odoo S.A. (Wikipedia 2021a; Ganesh ym. 2016). Alun perin Odoo keskittyi vahvasti pelkästään

toiminnanohjausjärjestelmän osioihin, mutta myöhemmin siihen on sisällytetty paljon muitakin liiketoimintaan liittyviä osioita, kuten verkkosivuston luonti tai verkkokurssien järjestäminen. Tästä johtuen sekä yrityksen että ohjelmiston nimi vaihdettiin pois OpenERP:stä, päätyen nykyiseen nimeen Odoo. (Ganesh ym. 2016.)

Odoo S.A. tarjoaa Odoota suoraan asiakkailleen joko SaaS-mallisena pilvipalveluna tai paikan päälle asennettuna (Odoo S.A. 2022a; Odoo S.A. 2022d). Tämän lisäksi Odoo S.A.:lla on eri tasoisia kumppaneita, jotka toimittavat Odoo-järjestelmiä omille asiakkailleen useilla eri malleilla. Odoo S.A. ja sen kumppanit myös jakavat tietoa ohjelmistosta sekä mahdollisista asiakkaista alueillaan. (Odoo S.A. 2022e.)

Odoo S.A. on vuosien saatossa kasvanut pienestä belgialaisyhtiöstä isoksi kansainväliseksi toimijaksi ERP-markkinoilla. Sillä on nykyään 10 eri toimipistettä ja yli 3500 kumppania ympäri maailmaa. Käyttäjää Odoolle on jo yli 7 miljoonaa ja yksittäisten asiakkaiden käyttäjämäärät vaihtelevat pienten yritysten 1 käyttäjän ja isoimpien asiakkaiden yli 300 tuhannen käyttäjän välillä. (Odoo S.A. 2022c.)

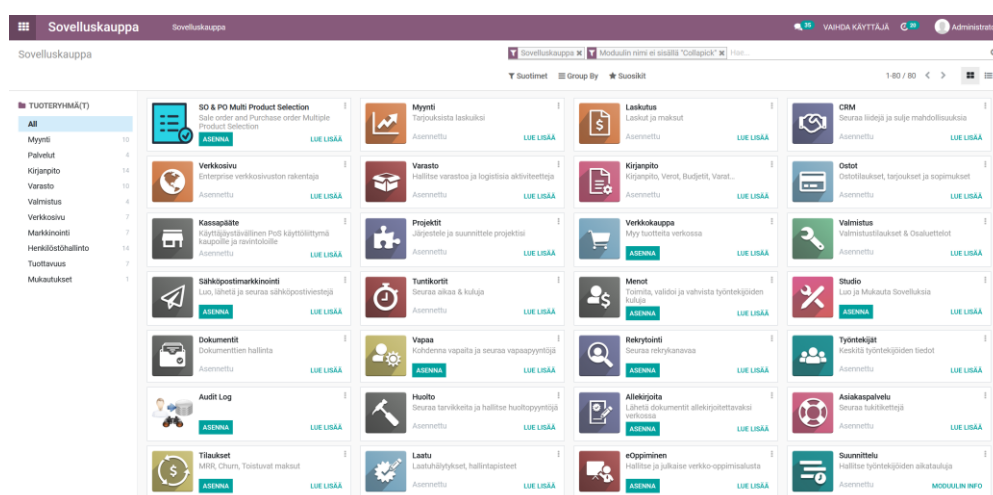
### **3.1 Odoon versiot ja ohjelmistoversiot**

Odoosta on tarjolla kaksi erilaista versiota: ilmainen Odoo Community eli yhteisöversio ja maksullinen Odoo Enterprise eli yritysversio. Molemmat versiot käyttävät samaa ydinkoodia, mutta Odoo Enterprise tuo mukanaan useita laajennettuja toiminnallisuuksia, mobiilisovelluksen, toiminnallisen tuen sekä ohjelmistoversion päivitys mahdollisuuden ja hostauspalvelun. (Odoo S.A. 2022f.) Tämä tarkoittaa käytännössä sitä, että Odoo Communityn ottava asiakas joutuu joko pystyttämään ohjelmiston itselleen paikan päälle tai etsimään Odoo-kumppanin, joka tarjoaa ohjelmiston pystytystä ja ylläpitoa palvelimella. Ero hinnassa on kuitenkin huomattava, sillä Odoo Enterprisestä asiakas joutuu palvelimen ylläpito- ja kustannuksien lisäksi maksamaan moduuli- ja käyttäjämäärä kohtaisia kuukausimaksuja, jotka voivat helposti nousta satoihin euroihin kuussa (Odoo S.A. 2022d).

Kahden erilaisen version lisäksi Odoosta on saatavilla useita eri ohjelmistoversioita. Vanhemmat ohjelmistoversiot ovat nimeltään sellaisia kuin OpenERP 8.0, kun taas uudemmat, ohjelmistoversiosta 9.0 lähtien, ovat nimeltään sellaisia kuin esimerkiksi Odoo 14.0. (Odoo S.A. 2022g; Ganesh ym. 2016.) Odoo S.A. pyrkii nykyään julkaisemaan uuden ison ohjelmistoversion kerran vuodessa, yleensä syys-lokakuun aikaan. Näissä päivityksissä tuodaan sisään uusia toiminnallisuuksia sekä korjauksia olemassa oleviin toiminnallisuuksiin. Uusin Odoosta julkaistu ohjelmistoversio on 15.0, joka julkaistiin lokakuussa vuonna 2021. (Odoo S.A. 2022g.)

## 3.2 Modulaarisuus

Odoon rakenne oli sen alkuaikoina muodostettu tiiviiksi paketiksi, joka sisälsi aina tietyn määrän toiminnallisuuksia. Ohjelmistoversion 9.0 aikaan Odoo S.A. kuitenkin luopui Odoossa tästä rakenteesta ja siirtyi käyttämään sen nykyistä modulaarista rakennetta. (Ganesh ym. 2016.) Käytännössä tämä tarkoittaa sitä, että Odoon asennuksen alussa ympäristöön ei mene kuin minimimäärä moduuleja sisään ja ympäristön haltija tai hallinnoija pystyy tämän jälkeen määrittämään mitkä moduulit siihen asennetaan (Odoo S.A. 2022a). Näitä moduuleja pystytään vielä jälkikäteen lisäämään ja poistamaan Odoon sisäisen Sovelluskaupan (kuva 1) kautta.



Kuva 1. Kuvakaappaus Odoo Enterprise:n Sovelluskaupasta käyttöliittymässä (Kuva: Tapio 2022).

Modulaarisuuden kautta Odoota pystytään kehittämään ja käyttämään hyvinkin kohdistetusti sekä joustavasti. Tämä tarkoittaa, että lähdekoodi on purettu pienempiin kokonaisuuksiin ja sitä on helpompi käsitellä sekä päivittää. (Ganesh ym. 2016.) Tämä ei kuitenkaan tarkoita, että moduuleja pystyisi asentamaan miten haluaa, sillä monet moduulit vaativat toimiakseen yhden tai useamman muun moduulin asennuksen (kuva 2). Kaikille Odoon moduuleille on kuitenkin yhteistä, että ne tarvitsevat base-moduulin, eli Odoon ydintoimintojen, asennuksen.

#### Riippuvuudet

| Nimi     | Tila      |
|----------|-----------|
| product  | Asennettu |
| barcodes | Asennettu |
| digest   | Asennettu |

Kuva 2. Kuvakaappaus Odoon Varasto-moduulin riippuvuuksista käyttöliittymässä (Kuva: Tapio 2022).

Modulaarisuuden myötä kaikkia yksittäisen osa-alueen toiminnallisuuksia ei tarvitse sitoa yhteen osaan, tai tässä tapauksessa moduuliin, vaan ne voidaan levittää pienempien, laajentavien moduulien alle. Nämä laajentavat moduulit voivat lisätä Odoohon uusia toiminnallisuksia, näkymiä tai raportteja, tai ne voivat muokata jo olemassa olevia asioita (Odoo S.A. 2022a). Esimerkiksi Odoo Community Association tuottaa pääasiallisesti moduuleja, jotka muokkaavat jo olemassa olevia asioita sen sijaan, että ne lisäisivät Odoohon uusia kokonaisuuksia (Odoo Community Association 2022b). Useimmat Odoohon tehdyt moduulit ovat laajentavia moduuleja.

Kaikki Odoon toiminnallisuus ydintoimintojen jälkeen, ja jopa osa ydintoiminoista, tuodaan sisään moduulien kautta. Tämän takia on tärkeää, että moduulien sisältö kerrotaan jo käynnistysvaiheessa Odoolle. Tämä tapahtuu moduulien sisäisten manifest.py- ja init.py-tiedostojen avulla. Manifest-tiedosto tuo sisään kaikki näkymät, moduulin yleistiedot ja riippuvuudet sekä käyttöoikeustiedot. Init-tiedosto puolestaan tuo sisään kaikki Pythonilla luodut tietomallit ja niiden toiminnallisuudet. (Odoo S.A. 2022a; Odoo S.A. 2022h.)

Odoon moduulit jakautuvat karkeasti kahteen eri tyyppiin, joka ilmoitetaan moduulin manifest-tiedostossa: sovellukset ja extrat (Odo S.A. 2022h). Moduuleja, jotka tuovat Odoohon isoja kokonaisuuksia, kutsutaan usein Sovelluksiksi. Ne näkyvät yleensä käyttäjälle ja tuovat Odoohon suurempia toiminnallisia kokonaisuuksia, kuten esimerkiksi myynnin ja siihen liittyvät toiminnot Myynti-moduulin kautta. Nämä moduulit sisältävät usein paljon uusia tietomalleja ja ne tuovat mukanaan lisäyksiä kaikille arkkitehtuurin eri tasoille. Puolestaan pienempiä moduuleita, jotka laajentavat olemassa olevia toiminnallisuksia tai tietomalleja eivätkä välttämättä tuo lisäyksiä kaikille arkkitehtuurin eri tasoille, kutsutaan usein extroiksi. (Odo S.A. 2022a.)

### **3.3 Arkkitehtuuri**

Odo on arkkitehtuuriltaan hyvin samankaltainen kuin muutkin nykyaikaiset verkkopohjaiset ohjelmistot, sillä sen arkkitehtuuri on monitasoinen (kts. liite 1). Odoon arkkitehtuurissa on kolme eri tasoa, joissa ohjelmisto toimii: esitys-, logiikka- ja datataso. (Odo S.A. 2022a.) Aiemmin Odoon arkkitehtuuri pohjautui MVC-arkkitehtuuriin, mutta muutos arkkitehtuuriin toteutettiin samalla, kun modulaarisuus otettiin käyttöön (Ganesh ym. 2016).

Odoon esitystaso koostuu käyttöliittymästä, johon logiikka- ja datatasojen toiminnot sekä tiedot käännetään käyttäjäystävällisempiin muotoihin. Tarkoituksena tällä on esittää tiedot käyttäjälle selkeästi ja ymmärrettävästi. Suurin osa esitystason käyttöliittymistä pohjautuu HTML5:n, CSS:n, JavaScriptin, XML:n sekä Qwebin avulla muodostettuihin näkymiin, joissa käyttäjä pystyy vuorovaikuttamaan ohjelman kanssa. (Odo S.A. 2022a.) Odoon esitystaso tekee aina tiedonhakupyyynnöt logiikkatasolle, ei suoraan datatasolle (Odo S.A. 2022a). Suurin osa Odoon esitystason näkymistä koostuu toistuvien peruskomponenttien lisäksi lista- tai lomakenäkymistä (Odo S.A. 2022i). Näiden näkymien lisäksi Odoossa on saatavilla monia toiminnanohjausjärjestelmälle tyypillisiä näkymiä, kuten kalenteri-, gantt-, kaavio-, kanban- ja pivot-näkymä (Odo S.A. 2022j).

Logiikkataso on Odoossa vastuussa siitä, että käyttäjän tai järjestelmän aloittamat toiminnot tapahtuvat oikein ja että tieto liikkuu datatasosta esitystasolle. Käytännössä logiikkataso toimii välikätenä kahden muun tason välillä ja pitää huolen, että järjestelmän väärinkäyttö minimoidaan. Odoon logiikkataso koostuu Pythonilla sekä JavaScriptillä toteutetuista toiminnallisuuksista ja Pythonilla toteutetuista malleista. (Odo S.A. 2022a.) Logiikkatason olennaisena osana toimii ORM-taso. Tämä tarkoittaa, että suurinta osaa tarvittavasta SQL-koodista ei tarvitse kirjoittaa käsin, vaan ORM-taso tuo mukanaan yleisiä metodeja, joilla kyselyt datatasolle voidaan hoitaa. Lisäksi ORM-taso sisältää useita tietoturvasuutta lisääviä autentikaatio- ja turvallisuusmekanismeja. (Odo S.A. 2022k; Odo S.A. 2022l.)

Odoon datataso on sidottu PostgreSQL-relaatiotietokantaan, johon Odoon käytämien mallien tarvittavat tiedot tallennetaan. Datatason sisältöön vaikuttavat pääasiassa Odoon modulaariseen kokonaisuuteen valitut palat ja niiden kautta lisääntyvät tietomallit. Datatasoon pääsy tapahtuu aina logiikkatason läpi, eikä datataso keskustele koskaan suoraan esitystason kanssa. (Odo S.A. 2022a.)

## 4 Helppokäyttöliittymät

Opinnäytetyössä isossa osassa toimivat myös Collapickin kehittämät helppokäyttöliittymät, jotka liitetään osaksi Odoon kokonaisuutta. Ison toiminnanohjausjärjestelmän käyttöliittymä sisältää hyvin paljon tietoa, eikä kaikki se tieto ole aina tarpeellista, joten Collapick on vuosien saatossa pyrkinyt kehittämään mobiiliresponsiivisia helppokäyttöliittymiä, joissa näkyvillä on vain tarpeellinen tieto. Näiden helppokäyttöliittymien avulla työntekijän on nopeampaa ja helpompaa käyttää järjestelmää kuin alkuperäisen käyttöliittymän kautta.

Usein helppokäyttöisyys mielletään niin, että jokin on helppokäyttöinen, kun sen käyttäminen on vaivatonta, nopeaa, helposti opittavissa ja virheettius on alhainen. Helppokäyttöisyyttä tavoitellessa pitää pystyä ymmärtämään käyttäjän tarpeita sekä odotettuja toimintatapoja. (Wikipedia 2021b.) Kun tämä

helppokäyttöisyyden tavoittelu tuodaan käyttöliittymien suunnitteluun mukaan, on tärkeää toimia vuorovaikutuksessa loppukäyttäjien kanssa. Saadun palautteen pohjalta käyttöliittymien kehittäminen helppokäyttöisemmäksi on aikaa vievä, jatkuva prosessi.

#### 4.1 Ponniste-helppokäyttöliittymät

Vuonna 2015 Collapick laajensi toimintaansa ja alkoi kehittämään Ponniste-tuotetta, jonka oli tarkoitus olla mobiiliresponsiivinen ja integroitavissa teollisuuden järjestelmiin. Tarkoituksena oli toteuttaa helppokäyttöinen ja visuaalinen tuote, jolla työntekijän on helppo kirjata talteen tuotannon tapahtumia. (Collapick Company Oy 2022a.)

Ponniste-tuotteen helppokäyttöliittymät (kuva 3) rakennettiin React-koodilla, jonka avulla mobiiliresponsiivisuuden toteuttaminen oli helppoa. Ponniste-tuote rakennettiin toimimaan tietokoneella selaimessa ja mobiililaitteilla oman, erillisen sovelluksensa kautta.



Kuva 3. Mockup-kuva Collapickin Ponniste-tuotteen käyttöliittymistä selaimessa ja mobiililaitteessa (Kuva: Collapick Company Oy 2022).

Lopputuloksena Ponniste-tuotteesta tuli tuotannonohjausjärjestelmä, jonka integroinnin olemassa oleviin järjestelmiin Collapick toteutti useille eri aloilla toimiville asiakkailleen (Collapick Company Oy 2022c).

Ponniste-tuotteen kehityksen myötä Collapickilla on valmiiksi laaja osaaminen helppokäyttöliittymistä sekä niiden toteutuksesta. Lisäksi Collapickilla on tallessa Ponniste helppokäyttöliittymien komponenttikirjastot, joiden avulla uusien helppokäyttöliittymien kehittämisen aloittaminen on nopeaa ja helppoa.

## **4.2 Collapick Tempo -helppokäyttöliittymät**

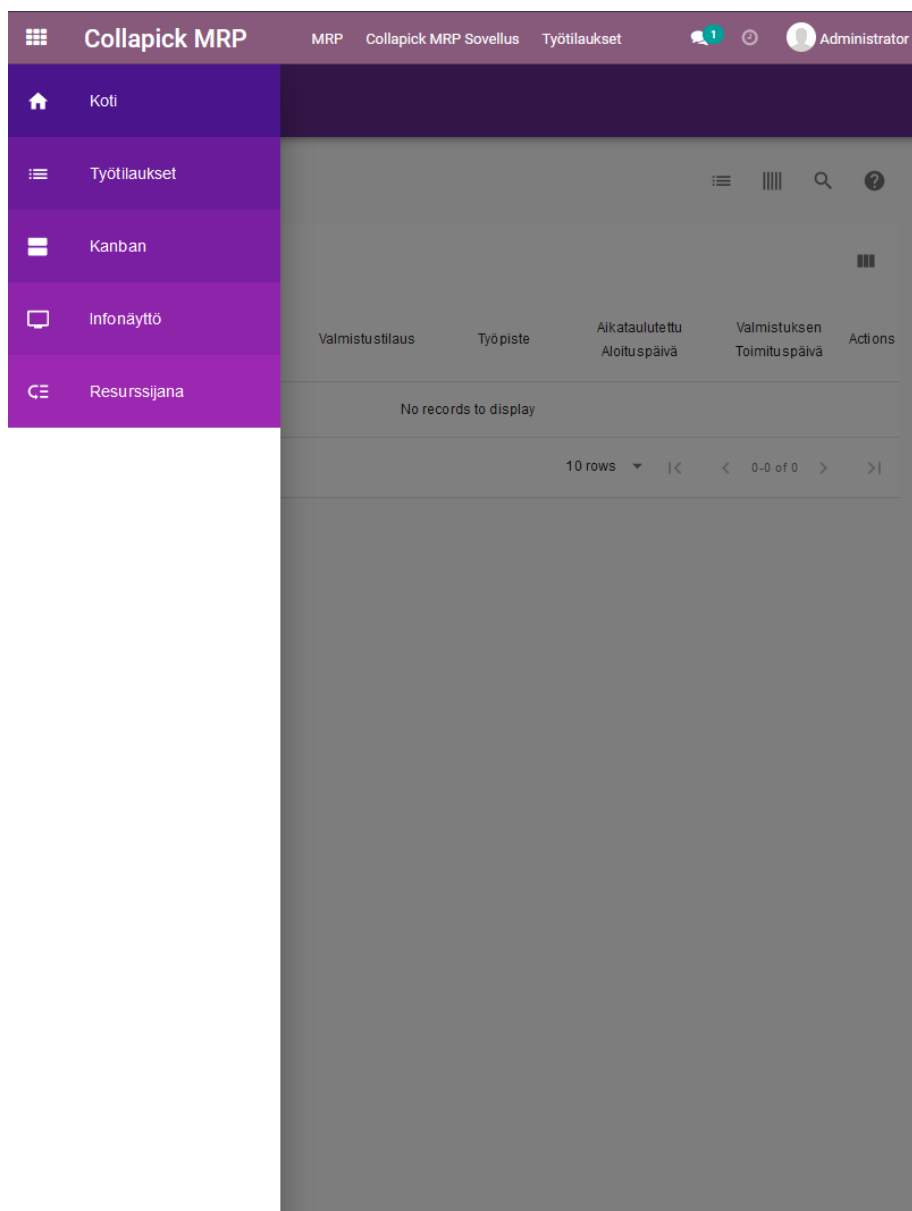
Vuonna 2020 Collapick teki strategisen muutoksen siirtyä Odoon kumppaniksi, minkä myötä Ponniste-tuotteen kehityksen fokus siirtyi Odoohon. Tarkoituksena oli muodostaa Ponniste-helppokäyttöliittymien pohjalta uudet helppokäyttöliittymät, joita pystyttäisiin hyödyntämään Odoossa. Odoohon siirtyminen tarkoitti myös laajempaa kokonaisuutta, kun tuotannonohjauksesta siirryttiin toiminnanohjaukseen. Collapickin aiempi kokemus teollisuudesta auttoi nopeasti ymmärtämään mitä helppokäyttöliittymiltä Odoossa vaadittaisiin. (Collapick Company Oy 2022a.)

Collapick Tempo on huomattavasti laajempi kokonaisuus kuin aiempi Ponniste-tuote, joten se sisältää paljon muutakin kuin vain helppokäyttöliittymät. Collapick Tempo laajentaa useimpia Odoon Sovellus-moduuleja, kuten Myyntiä, Ostoa, Valmistusta, Varastoa ja Projektia. Opinnäytetyön toteutuksen aikaan helppokäyttöliittymät on toteutettu edellä mainituista kolmelle viimeiselle: Valmistukselle, Varastolle ja Projektille. (Collapick Company Oy 2022d.)

Modulaarisesta toteutuksesta johtuen Collapickin toteuttamat helppokäyttöliittymät ylittävät Odoon vastaavien moduulien rajoja, minkä takia Collapickin moduuleille täytyy muodostaa niiden omat käyttöoikeusryhmät. Näin Odoon puolella pääsyä valikoihin ja tietomalleihin saadaan rajattua tai vapautettua tarkoituksen mukaan. (Odo S.A. 2022b.)



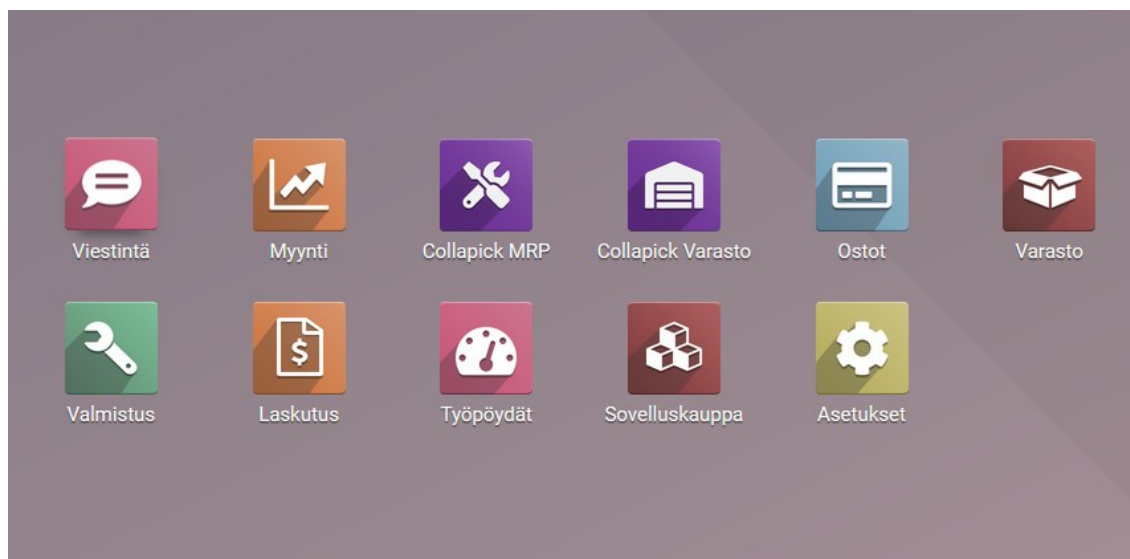
Collapick Tempon helppokäyttöliittymät on toteutettu Reactilla ja niissä on hyödynnetty aiempia Ponniste-tuotteen React-komponentteja. Helppokäyttöliittymien mobiiliresponsiivisuus on säilytetty ja niiden värimaailmaa on päivitetty vastaamaan Collapickin logon violettiä (kuva 4).



Kuva 4. Kuvakaappaus Collapick MRP:n helppokäyttöliittymistä Odoon Enterprise:ssä, skaalautuneena iPadiin (Kuva: Tapio 2022).

Helppokäyttöliittymät on muodostettu erillisinä kokonaisuuksina omiin moduuleihinsa, jotta ne toimivat modulaarisesti ja näin ollen niillä on omat valikkonsa. Moduulien valikot on upotettu Odoon käyttöliittymään (kuva 5), mutta

tavoitteena on koostaa erillisten valikkojen sisältö yhden valikon alle opinnäytetyön aikana (Odoon S.A. 2022m).



Kuva 5. Kuvakaappaus Odoon Enterprisen valikkonäkymästä. Näkyvissä Collapick MRP ja Collapick Stock -moduulien valikot (Kuva: Tapio 2022).

Collapick Tempon helpokäyttöliittymät keskustelevat Odoon kanssa sen ORM API:n kautta, hyödyntäen XML-RPC- ja JSON-RPC-rajapintoja. Näiden etäkäytösten kautta toimintojen suoritus-, tiedonhaku- sekä päivityspyynnöt tapahtuvat turvallisesti rajapinnan kautta (Odoon S.A. 2022i; Odoon S.A. 2022n).

## 5 Opinnäytetyössä käytetyt menetelmät ja aikataulu

### 5.1 Lähestymistapa ja aikataulu

Opinnäytetyö on toiminnallinen työ, joka tehtiin toimeksiantona Collapick Company Oy:lle. Opinnäytetyössä toteutettavan moduulin ja konfiguraatioiden tarve ilmeni, kun tekijä oli suorittamassa työharjoitteluaan toimeksiantajalla keväällä 2021. Opinnäytetyöprojektin suunnittelu aloitettiin yhdessä toimeksiantajan ja tekijän toimesta vuoden 2021 loppukeväästä. Työn aihe valittiin työn tekijän aiemman kokemuksen sekä kiinnostuksen pohjalta, huomioiden

toimeksiantajan senhetkiset tarpeet ja mahdollisuudet. Opinnäytetyön tekeminen aloitettiin vuoden 2021 elokuussa, kun työn tekijä oli päättänyt työharjoittelunsa suorittamisen ja siirtynyt toimeksiantajalle töihin. Opinnäytetyön toteutus tapahtui pääasiallisesti työn tekijän omalla ajalla, työajan ulkopuolella.

Opinnäytetyöprosessi alkoi perehtymisellä käytettävien järjestelmien tekniseen dokumentaatioon, taustakirjallisuuteen ja lähdekoodiin. Lisäksi prosessin alussa muodostettiin yhteinen kuva työn tekijän ja toimeksiantajan välillä niistä toimeksiantajan moduuleista, joihin työn toteutus rajattiin. Konkreettisen työn toteutus alkoi lokakuussa 2021 ja toteutuksessa seurattiin ketteriä kehitysmenetelmiä. Kehitystyön testaus tapahtui kehitysvaiheen aikana tekijän toimesta ja kehitystyön loppuvaiheessa testauksessa olivat mukana toimeksiantajan työntekijät. Kehitystyön aikana palautetta toteutuksen etenemisestä ja suunnasta saatiin toimeksiantajalta.

Opinnäytetyön aikataulu oli vahvasti sidoksissa toimeksiantajan omaan kehitystyöhön ja asiakasprojekteihin. Toimeksiantajan moduulien ja helppokäyttöliittymien kehitys oli jatkuvaa koko opinnäytetyöprosessin ajan, mikä asetti vaatimuksia toteutuksen kehittämiseksi, testaukselle ja aikataululle. Lisäksi toimeksiantajan asiakasprojekteille, joissa opinnäytetyön toiminnallista osuutta on tarkoitus hyödyntää, on määritelty suunniteltuja käyttöönottopäivämääriä alkuvuodelle 2022. Tavoiteaikataulu opinnäytetyön toiminnallisen osuuden valmistumiselle oli tästä johtuen asetettu vuoden 2021 loppuun mennessä. Työn kirjallista osuutta valmisteltiin koko prosessin ajan, mutta sen viimeistely jäi vuoden 2022 ensimmäiselle neljännekselle.

## **5.2 Käytetyt menetelmät ja työkalut**

Opinnäytetyössä käytetyt menetelmät ja työkalut määrittyivät sekä Odoon arkkitehtuurin vaatimien lähtökohtien että toimeksiantajan käytettyjen kehitystyökalujen pohjalta. Kehitystyötä varten käytössä oli toimeksiantajalta saatu Mac OSX -käyttöjärjestelmällä toimiva kannettava tietokone.

Odoon arkkitehtuuri määrittä, että kehitettävä moduuli tuli toteuttaa käyttäen Python- ja XML-ohjelmointikieliä. Kehitystyötä ja testausta varten tarvittiin myös Odoon-ympäristö, mikä tarkoitti sellaisen asentamista paikallisesti kehitystyöhön käytettävälle koneelle hyödyntäen Docker-alustaa.

Koska opinnäytetyön toteutus oli vahvasti sidonnainen muuhun toimeksiantajan kehitystyöhön Odoon suhteen, oli selvää, että käytetyt työkalut ja menetelmät olisivat samoja myös opinnäytetyössä yhdenmukaisuuden säilyttämiseksi. Toimeksiantajan suosituksesta versionhallinnassa hyödynnettiin Amazonin AWS-pilvipalveluita ja kehitystyössä VSC-editoria sekä sen tiettyjä laajennuksia.

### **5.2.1 Odoo-toiminnanohjausjärjestelmä**

Opinnäytetyön keskeisessä osassa olevasta Odoo-toiminnanohjausjärjestelmästä on toteutusta varten käytössä ohjelmistoversio 14.0, joka on julkaistu vuoden 2020 lokakuussa (Odoo S.A. 2022g). Toteutuksen kehittämistä varten käytössä on Docker-image, joka sisältää Odoon lähdekoodit ja tarvittavat asetukset toimivan kontin luomiseksi. Nämä asetukset ovat toimeksiantajan toimittamat ja niiden muokkaukset toteutetaan toimeksiantajan ohjeiden mukaisesti. Mukana olevat lähdekoodit sisältävät Odoon ydinkoodit sekä Community- ja Enterprise-version laajentavien moduulien koodit.

Toteutuksen on tarkoitus toimia sekä Community- että Enterprise-versiossa, joiden sisältö on hieman erilainen (kts. liite 2). Näin ollen kehittämistä ja testausta varten käytössä ovat erilliset kontit, joista vain toisessa ovat mukana Enterprise-version lähdekoodit. Molempien ympäristöjen tietokantana on PostgreSQL-relaatiotietokanta, joka rakentuu myös Docker imagen kontissa. Tietokannan käsittelyn apuvälineenä käytössä on DBeaver Community, joka on ilmainen universaali tietokantatyökalu (DBeaver 2022).

### 5.2.2 Collapickin moduulit ja helppokäyttöliittymät

Toimeksiantajan kehitystyö Odoon parissa on jatkuvaa ja saatavilla olevia Odoo-moduuleita on suurehko määrä. Työn laajuuden rajaamiseksi näistä moduuleista valittiin työssä käytettäväksi viisi suurinta, joihin helppokäyttöliittymien yhdistämistä on tehty. Valituiksi moduuleiksi valikoituivat seuraavat:

- Collapick MRP, joka liittyy Odoon Valmistus-moduuliin
- Collapick Stock, joka liittyy Odoon Varasto-moduuliin
- Collapick Project, joka liittyy Odoon Projekti- ja Työntekijät-moduuleihin
- Collapick Sale, joka liittyy Odoon Myynti-moduuliin
- Collapick Purchase, joka liittyy Odoon Osto-moduuliin

Valinta perustui helppokäyttöliittymien lisäksi moduuleihin liittyvien osa-alueiden kokoon ja arvioituun käyttömäärään. Opinnäytetyöprosessin aikana näistä moduuleista kolme sisältävät omat helppokäyttöliittymänsä ja kahdella on yhteyksiä näiden kolmen helppokäyttöliittymiin. Collapick MRP, Collapick Stock ja Collapick Project -moduulit sisältävät omat helppokäyttöliittymänsä ja Collapick Sale sekä Collapick Purchase tuovat näihin helppokäyttöliittymiin erilaisia lisä-kenttiä.

Opinnäytetyössä käytetään versionhallinnan development- eli kehityshaarasta löytyviä versioita näille viidelle moduulille. Työn kehityksessä käytettävien versioiden päivityksestä kehityskoneelle keskustellaan toimeksiantajan kanssa ja näihin viiteen moduuliin tulevat muutokset pidetään tallessa vain paikallisella koneella, kunnes ne viedään lopputestaukseen. Tässä vaiheessa ne lisätään kehityshaaraan, jotta mahdollisten ongelmien ilmetessä vaikutukset jäävät vain kehittäjien ympäristöihin. Erillisen haaran käyttöä mietittiin myös, mutta kehityshaaran jatkuvan päivittymisen vuoksi tämä ajatus haudattiin lopulta.

### 5.2.3 Docker

Opinnäytetyössä ja toimeksiantajan toiminnassa Odoo-ympäristöjen toiminnasta vastaa Docker-alusta, jolla ympäristön kontit pyörivät. Docker-alusta

pyörittää kontteja, jotka rakentuvat Docker-imagen kautta (Docker 2022). Opinnäytetyön tapauksessa kokonaisuuden muodostavat neljä erillistä konttia, joilla on kullakin omat tehtävänsä:

- Odoon-kontti, jossa Odoon-toiminnanohjausjärjestelmän pääasiallinen toiminta tapahtuu.
- Psql-kontti, jossa PostgreSQL-relaatiotietokanta pyörii.
- Cron-kontti, joka suorittaa ajoitettuja tehtäviä Odoon ja konttien suhteen.
- Nginx-kontti, joka toimii kokonaisuuden proxy-palvelimena.

Docker-imagessa käytettävät asetukset ovat toimeksiantajan toteuttamat ja niiden muokkauksesta työn toteutusta varten on sovittu toimeksiantajan kanssa. Nämä asetukset sisältävät myös virheviestien ja -tilanteiden käsittelyä kontteissa, joka on kehitystyön kannalta tärkeää. Docker-alustan kautta saatavia tietoja pystytään hyödyntämään sekä kehitystyössä että testauksessa.

#### **5.2.4 Amazon Web Services -pilvipalvelut ja versionhallinta**

Toimeksiantajan versionhallinta tapahtuu Amazonin Amazon Web Services-pilvipalvelun kautta, joka tarjoaa yrityksille kehitystyökaluja ja IT-infrastruktuuria pilvessä (Amazon Web Services 2022). Työn tekijä saa tunnukset tähän pilvipalveluun toimeksiantajalta.

Versionhallinta tapahtuu AWS-palvelun CodeCommit-osiossa, jossa kaikkien toimeksiantajan Odoon-moduulien koodia säilytetään ja johon opinnäytetyössä kehitettävän moduulin koodit myös viedään säilytystä varten. Toimeksiantajan moduuleilla on jokaisella oma repositorionsa ja näillä on kaikilla vastaavasti vähintään kaksi haaraa, master- eli päähaara ja development- eli kehityshaara.

Oleellisena osana versionhallintaa ovat myös CodePipeline-osion kautta tapahtuva päähaaran automaattinen rakennus. Nämä rakennetut moduulit pystytään viemään komentorivin kautta palvelinympäristöön helpommin. Opinnäytetyössä syntyvälle moduulille tulee päähaaraan myös vastaava automaattinen rakennus.

### 5.2.5 Visual Studio Code ja laajennukset

Opinnäytetyön varsinainen tuotos tapahtuu täysin koodilla, joka kirjoitetaan editorin kautta. Toimeksiantajan suosituksesta työssä käytetään Visual Studio Code -editoria, johon lisätään laajennuksia helpottamaan työskentelyä.

Laajennukset kattavat muihin työkaluihin ja ohjelmointikieliin liittyviä toimintoja sekä helpotuksia. Seuraavat laajennukset asennettiin editoriin auttamaan kehitystyötä:

- Auto Rename Tag, joka korjaa HTML- ja XML-kielten tageja.
- Bracket Pair Colorizer 2, joka värjää koodin sulkuja ja helpottaa sen lukemista.
- Docker, joka sallii Docker-alustan käyttöä ja lokien lukemista editorin kautta.
- Odo Code Snippets, joka on kehitetty nimenomaan Odoon lähdekoodien helpompaan lukemiseen.
- Prettier, joka formatoi koodia luettavampaan muotoon.
- Pylance, joka sisältää helpotuksia Python-kielen käyttöön.
- Rainbow CSV, joka värjää CSV-tiedostojen sisältöä, tehden niistä huomattavasti helpompia lukea.

Työssä käydään läpi suuria määriä lähdekoodia, joten koodin luettavuus on tärkeä asia läpi koko prosessin. Erityisesti CSV-tiedostojen luettavuus on tärkeää, koska käyttöoikeuksien määrittelyt on tehty näihin tiedostoihin.

## 6 Tuotoksen suunnittelu ja toteutus

Työn toiminnallisen osuuden toteutus alkoi vuoden 2021 elokuussa ja työn tekijä oli alustavasti varannut sille noin 180 tuntia käytettäviä työtunteja. Toteutus jakautui kolmeen selvästi erilaiseen vaiheeseen, joille oli kullekin varattu alustavasti tietyn verran työtunteja.

Ensimmäinen vaihe oli suunnitteluvaihe, jossa tarkoituksena oli tiedon kartuttaminen ja toteutuksen kokonaisuuden tarkempi suunnittelu tämän tiedon pohjalta. Tälle vaiheelle oli alustavasti laskettu noin 50 tuntia työtunteja.

Toinen vaihe oli kehitysvaihe, eli varsinaisen tuotoksen toteutus ja testaus. Tälle vaiheelle oli varattu alustavasti eniten tunteja, noin 110 työtunnin verran. Tämän vaiheen testaus oli tarkoitus suorittaa puhtaasti työn tekijän toimesta.

Kolmas ja viimeinen vaihe oli viimeistelyvaihe, jossa tehtiin tuotoksen käyttöönotto, laajempi testaus ja viimeistely. Tälle vaiheelle oli alustavasti varattu vähiten työtunteja, vain noin 20 tunnin verran. Toteutuksen kolmannessa vaiheessa oli lisäksi tarkoitus käyttää testauksessa mukana muita henkilöitä toimeksiantajan kautta.

## **6.1 Suunnitteluvaihe**

Toteutuksen alkuperäinen suunnitelma oli tuottaa yksi Odoon-moduuli, nimeltään Collapick Base, joka toimisi alustana, johon muita toimeksiantajan moduuleja liitettäisiin modulaarisesti sekä lisätä joitain pieniä konfiguraatioita näihin muihin moduuleihin. Tämä epämääräinen ja hyvin löyhä idea oli muodostettu yhdessä toimeksiantajan kanssa, eikä se vielä tässä vaiheessa sisältänyt tarkempia linjauksia toteutuksen suunnasta, pelkästään sen sisällöstä. Tämän takia ensimmäisenä oli perehdyttävä Odoon omiin dokumentaatioihin sekä lähdekoodiin moduulien rakenteista ja siitä, minkälaisia haasteita tai rajoituksia konfiguraatiot eri moduulien välillä aiheuttaisivat.

Odoosta löytyi vain harvoja tieteellisiä lähteitä ja niistäkin suurimman osan ajantasaisuus oli kyseenalainen. Tästä johtuen parhaaksi ja luotettavimmaksi lähteeksi tiedon keruulle muodostui Odoon oma tekninen dokumentaatio, joka oli saatavilla Odoon verkkosivuilla. Tämä dokumentaatio oli hyvin laaja ja versiosta 13.0 lähtien Odoon on toteuttanut dokumentaation versiokohtaisesti. Näin oli helppo löytää toteutuksessa käytetyn version 14.0 omat dokumentaatiot, joiden



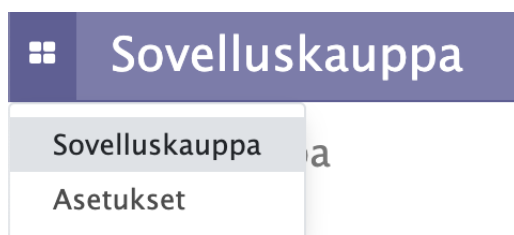
kautta Odoon tarkempia teknisiä rajoituksia pystyttiin alkaa ottamaan huomioon toteutuksen suunnittelussa.

Kahden viikon perehtymisen jälkeen oli selvää, että toteutuksessa tuotettavan Odoon-moduulin tulee olla sisällöltään kevyt, koska sille ei haluta mitään riippuvaisuuksia muihin Odoon osiin kuin Odoon omaan base-moduuliin (kuva 6). Tämä moduuli sisältää Odoon ydintoiminnot, joten se on aina asennettuna, ja toteutettavan moduulin oli tarkoitus pystyä olemaan järjestelmässä asennettuna ilman, että Odoossa on muita moduuleja asennettuna.

```
1  # -*- coding: utf-8 -*-
2  {
3      'name': "Collapick Base",
4
5      'summary': """
6          Collapick Base""",
7
8      'description': """
9          Collapick Base addon that makes the base for other Collapick addons.
10         """,
11
12      'author': "Collapick",
13      'website': "http://www.collapick.com",
14
15
16      'category': 'Collapick',
17      'version': '0.3',
18
19      # any module necessary for this one to work correctly
20      'depends': ['base',
21                ],
22
```

Kuva 6. Kuvakaappaus Collapick Base -moduulin kehitysvaiheen manifest-tiedostosta (Kuva: Tapio 2022).

Koska moduulin riippuvuus haluttiin rajata hyvin tiukasti, tarkoitti se myös, että siihen voi laittaa laajennuksia vain niihin luokkiin ja näkymiin, mitä Odoon base-moduuli mukanaan järjestelmään tuo. Tämä rajasi sisällön muutamiin tietomalleihin, kuten käyttäjiin, ja Asetukset- sekä Sovelluskauppa-valikoiden näkymiin (kuva 7).



Kuva 7. Kuvakaappaus Odoon Community-version valikoista, kun vain base-moduuli on asennettuna (Kuva: Tapio 2022).

Alkuperäisen suunnitelman mukaan moduulin avulla rajattaisiin käyttäjille näytettäviä osioita asennetuista moduuleista huolimatta, huomioiden kuitenkin tarvittavat käyttöoikeudet helppokäyttöliittymien puolella. Esimerkiksi hallissa työtään tekevän hitsaajan ei haluttu näkevän Odoon Valmistus-moduulin valikkoa, vaan pelkästään Collapick MRP -moduulin valikko, jonka kautta töiden valmistamista merkitään. Tämä käytännössä tarkoitti, että käyttäjille piti luoda uusia käyttäjäryhmiä, joille ei anneta pääsyä kuin muutamisiin näkymiin, mutta annetaan kuitenkin täydet oikeudet tietomalleihin. Jotta nämä käyttäjäryhmät toimisivat modulaarisesti, oli ne luotava jo Collapick Base -moduulissa. Näin niihin pystytään lisäämään käyttöoikeuksia muiden moduulien kautta, silloin kun nämä muut moduulit ovat asennettuna.

Alkuperäisessä suunnitelmassa oli myös tarkoitus muodostaa jonkinlainen yleisvalikko Collapickin helppokäyttöliittymille ja asetuksille. Koska Asetukset-näkymä on osa Odoon base-moduulia (kuva 7) ja uuden yleisvalikon luominen ei vaadi kuin base-moduulin, oli tämä osio helpompi suunnitella. Näin selvisi, että Collapick Base pystyi siis luomaan oman valikkonsa sekä laajentamaan Asetukset-näkymää ilman, että se joutui ottamaan muita riippuvuuksia käyttöön.

Seuraavan noin kahden viikon aikana selvitystyö keskittyi tarkemmin viiden muun Collapick-moduulin tarpeiden selvittämiseen. Tämä tarkoitti perehtymistä Odoon muiden moduulien sisältöihin ja siihen, miten nämä vastaavuudet heijastuvat Collapickin moduuleissa. Samalla selkeä ero Odoon Community- ja Enterprise-versioiden erinäisten moduulien välillä selkeni, joka auttoi rajaamaan käytettäviä Odoon tietomalleja. Tänä aikana toteutuksen rajoitukset selkenivät paremmin ja näistä rajoituksista käytiin aika ajoin keskustelua työn tekijän ja

toimeksiantajan välillä, että työssä saadaan muodostettua kaikki alkuperäisen suunnitelman mukaiset konfiguraatiot.

Vajaan kuukauden selvitystyön jälkeen suunnitelmalle oli muodostettu tarkemmat rajaukset ja nämä käytiin tarkemmin läpi yhdessä toimeksiantajan kanssa. Päätöksenä oli olla käyttämättä mitään Enterprise-version moduuleissa tuotavia tietomalleja mukana moduuleiden konfiguraatioissa, jotta lopullisen tuotoksen käyttäminen on mahdollista sekä Community- että Enterprise-versioissa. Pienen linjojen hiomisen jälkeen toteutus oli valmis siirtymään kehitysvaiheeseen, jossa alkoi varsinaisen koodin tuottaminen.

Suunnitteluvaiheeseen kulunut aika oli lopulta aika lähelle sitä, mitä työn tekijä oli itselleen kartoittanut ennen työn toteutuksen aloittamista. Aikaa tähän vaiheeseen kului yhteensä noin 55 työtunnin verran. Odoon laajojen dokumenttien ja lähdekoodien lukemiseen varattu aika oli riittävä ainakin tämän työn laajuuden huomioon ottaen.

## **6.2 Kehitysvaihe**

Vuoden 2021 syyskuun loppupuolella työn toteutus siirtyi kehitysvaiheeseen, joka alkoi Collapick Base -moduulin rakentamisella. Tarkoituksena oli ensin saada kyseinen moduuli valmiiksi ja sen jälkeen tehdä tarvittavia muutoksia muihin viiteen moduuliin.

Ensimmäisessä versiossa Collapick Base -moduuliin tulivat sisällöksi vain tarvittavat tiedostot, sen oman valikon koodit (kuva 8) ja kaksi uutta käyttäjäryhmää (kuva 9). Nämä käyttäjäryhmät eivät vielä tässä vaiheessa sisältäneet mitään tietomallien käyttöoikeuksia, ainoastaan rajasivat pääsyä moduulin omaan valikkoon. Moduulissa luotujen käyttäjäryhmien alustaviksi nimiksi tulivat Collapick User ja Collapick Manager.

```

<!-- Top menu item -->
<!-- Other Addons should have their React App under this -->
<menuItem id="collapick_base.menu_root"
  name="Collapick Base"
  web_icon="collapick_base,static/description/icon.png"
  sequence="2"
  groups="collapick_base.group_collapick_user"/>

```

Kuva 8. Collapick Base -moduulin valikon luonti XML-koodin kautta. Käyttöoikeutta rajattu käyttäjäryhmällä groups-kohdassa (Kuva: Tapio 2022).

```

<!-- Module Category for Collapick -->
<record model="ir.module.category" id="module_category_collapick">
  <field name="name">Collapick</field>
  <field name="description">Collapick related ventures</field>
  <field name="sequence">19</field>
</record>

<!-- Regular User for Collapick Modules -->
<record id="group_collapick_user" model="res.groups">
  <field name="name">Collapick User</field>
  <field name="category_id" ref="collapick_base.module_category_collapick"/>
</record>

<!-- Admin User for Collapick Modules -->
<record id="group_collapick_manager" model="res.groups">
  <field name="name">Collapick Manager</field>
  <field name="category_id" ref="collapick_base.module_category_collapick"/>
  <field name="implied_ids" eval="[(4, ref('group_collapick_user'))]"/>
  <field name="users" eval="[(4, ref('base.user_root')), (4, ref('base.user_admin'))]"/>
</record>

```

Kuva 9. Collapick Base -moduulin alustavien käyttäjäryhmien luonti XML-koodin kautta (Kuva: Tapio 2022).

Tämän jälkeen konseptin toimintaa pystyi testaamaan tekemällä kaksi samantyyppistä käyttäjää, joista toiselle oli annettu Collapick User -käyttäjäryhmä. Testauksessa käyttäjäryhmien käyttö toimi odotetusti eli valikon näkyvyys oli rajoitettu vain käyttäjälle, jolla oli käyttäjäryhmä Collapick User. Laajempaa kehitystä pystyi alkaa toteuttamaan näihin ryhmiin, kun konseptin toiminta oli varmistettu.

Seuraavana oli vuorossa lisätä Collapick Base -moduuliin alavalikko Asetukset-valikon näkymään sekä tehdä sinne pohja, johon muut moduulit pystyvät lisäämään omat asetusvalikkonsa. Tämän lisääminen oli toteutuksena hyvin helppo ja nopea tehdä, mutta testaukseen kului yllättävän paljon aikaa. Tämä johtui Odoon omien asetusvalikkojen modulaarisuudesta, jonka takia lisäselvitystyötä piti tehdä, että Collapick Base -moduulin asetusvalikko meni kohtaan, jossa ei se ei liikkuisi liikaa. Lisäksi Collapick Base päävalikon alle tuli alavalikko, jonka kautta Collapick Manager -ryhmän omaava käyttäjä pystyy siirtymään suoraan Collapickin asetusvalikkoon (kuva 10).



Kuva 10. Kuvakaappaus Collapick Base -moduulin tyhjästä päävalikosta, jonne on lisätty vain asetusvalikko (Kuva: Tapio 2022).

Seuraavana moduuliin piti lisätä käyttäjän sisäänkirjautumisen uudelleenohjaukseen logiikka, sillä se oli tietomallien käyttöoikeustasoja lukuun ottamatta viimeinen moduulista puuttuva asia. Aluksi tätä uudelleenohjautumisen logiikkaa tuli ohjelmoitua omaan funktioonsa, mutta testailun ja lähdekoodien tutkimisen jälkeen oli parempi muuttaa Odoon olemassa olevaa logiikkaa. Tämä siksi, että Odoon omassa logiikassa oli jo huomioitu tietoturvaa, eikä sitä näin ollen tarvitsi testata alusta asti.

Odoon oma uudelleenohjauksen logiikka perustuu siihen, että käyttäjälle avataan kirjautumisessa jokin tietty action, eli valikon avaava toiminto. Odoon omassa koodissa tämä toiminto on kuitenkin aina oletuksena Viestintä-valikon toiminto, ellei kehittäjätila ole käytössä. Tämä toiminto oli helppo rakentaa toimivaksi tuon kehittäjätilan toiminnon pohjalta, joka lähdekoodeista löytyi. Lopulta käyttäjän taakse laitettiin kenttä, johon käyttäjä voi valita toiminnon, joka kirjautumisessa uudelleenohjaukselle annetaan. Jos tämän kentän jättää tyhjäksi, niin käyttäjä menee oletusreittiä pitkin, eli päättyy Viestintä-valikkoon.

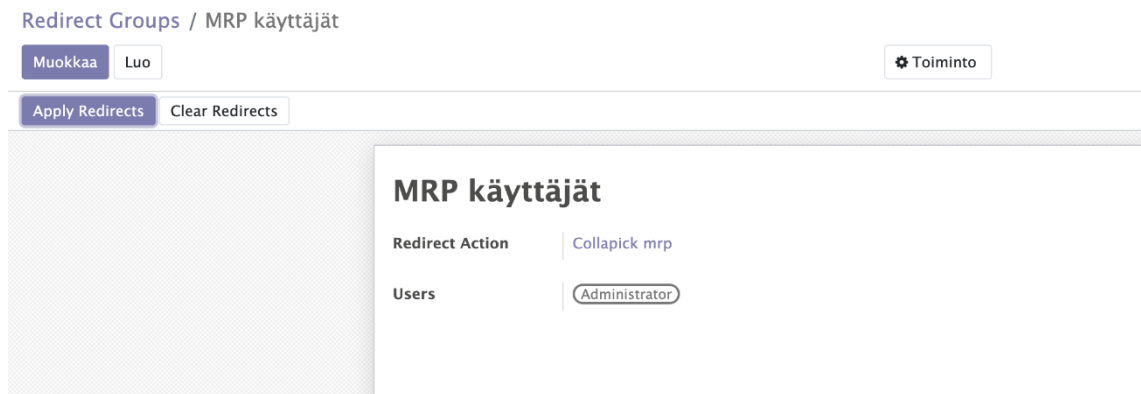
Kun uudelleenohjauksen logiikka oli testattu toimivaksi, oli aihetta katsoa tilannetta yhdessä toimeksiantajan kanssa, joka havaitsi heti lisätarpeen. Tämä oli tarve luoda uudelleenohjauksen ryhmiä, joihin voitaisiin lisätä käyttäjiä ja näin helpommin hallita sitä kautta, että minne kukin käyttäjä ohjautuu kirjautumisessa (kuva 11).

### Collapick Customization

|                 |               |
|-----------------|---------------|
| Redirect Action | Collapick mrp |
| Redirect Group  | MRP käyttäjät |

Kuva 11. Kuvakaappaus käyttäjän lomakkeelle lisätyistä kentistä, joilla uudelleenohjauksen toimintoa ja ryhmää voidaan hallita käyttäjän takaa (Kuva: Tapio 2022).

Tämän toteutus oli helppo tehdä, mutta toteutuksen testaus vaati hieman enemmän aikaa. Collapick Base -päävalikon alle lisättiin Collapick Manager -oikeudet vaativa valikko, jonka alta näitä ryhmiä pystyi hallinnoimaan. Lisäksi ryhmiin piti tehdä logiikka, joka vaihtaa valituille käyttäjille ryhmälle valitun toiminnon tai tyhjentää toiminnon kaikilta käyttäjiltä (kuva 12).



Kuva 12. Kuvakaappaus esimerkkiryhmästä, jolla useiden käyttäjien toimintoa voidaan hallita ryhmän kautta (Kuva: Tapio 2022).

Kirjautumisen uudelleenohjauksen logiikan toimivuuden testaamisen jälkeen kokonaisuutta käytiin toimeksiantajan kanssa läpi ja Collapick Base -moduulin sisällön todettiin olevan tässä vaiheessa riittävällä tasolla. Seuraavaksi oli siirryttävä viiden muun Collapick-moduulin pariin. Näiden viiden moduulin kohdalla tuli kuitenkin tässä vaiheessa vastaan pieni, mutta kriittinen huomioitava asia, eli se, että ne olivat kaikki edelleen aktiivisessa kehityksessä. Asiasta keskusteltiin yhdessä toimeksiantajan kanssa ja käytännöksi sovittiin, että työn tekijä ottaa kehitystyötä varten sen hetkiset versiot näistä viidestä moduulista, mutta ei päivitä käytettäviä versioitaan ennen viimeistelyvaihetta.

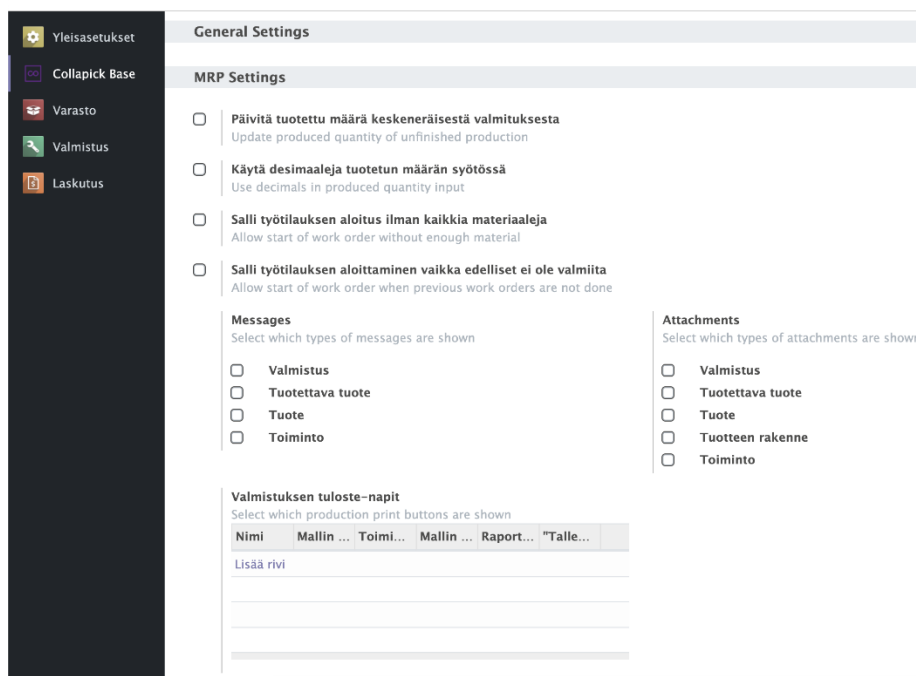
Tässä vaiheessa viiteen moduuliin tehtiin vain muutoksia niiden olemassa olevien valikkojen sijainteihin, näiden valikkojen pääsyoikeuksiin sekä luotiin asetusvalikot Collapick Base -moduulissa luodun asetusvalikon sisälle. Olemassa olevien valikkojen siirtäminen ja niiden pääsyoikeuksien rajaaminen ei ollut iso työ, mutta tätä tehdessä ilmeni, että käyttäjäryhmät Collapick User ja Collapick Manager ovat riittämättömät. Toimeksiantajan kanssa käydyn keskustelun jälkeen tuli päätös, että kaikkiin viiteen moduuliin tehdään omat User ja Manager käyttäjäryhmät. Näiden moduulien ja myös Collapick Base -moduulin olemassa

olevat käyttäjäryhmät nimettiin niin, että jokaisen moduulin käyttäjäryhmät kantavat nimeä mallilla moduulin nimi + User tai Manager (kuva 13).



Kuva 13. Kuvakaappaus käyttäjän ryhmistä ympäristössä, jossa asennettuna ovat Collapick Base ja Collapick MRP (Kuva: Tapio 2022).

Viiden moduulin eri valikkojen pääsyoikeuksien rajaaminen vei oman aikansa, mutta toteutus ei ollut erityisen vaikea. Kun kaikki näiden moduulien valikot oli siirretty tarpeen mukaan ja niiden pääsyoikeudet oli asetettu, oli vuorossa asetusvalikkojen teko (kuva 14). Tämäkään ei ollut vaikea tehdä, koska ensimmäisen moduulin jälkeen loppuihin neljään pystyi hyödyntämään ensimmäisessä tehtyä koodia. Toteutuksen testaaminen vei enemmän aikaa kuin itse toteutus, sillä modulaarisuuden toimivuutta piti testata eri yhdistelmillä näiden viiden moduulin suhteen.



Kuva 14. Kuvakaappaus Collapick MRP asetusvalikosta Collapick Base -moduulin asetusvalikon sisällä (Kuva: Tapio 2022).

Viimeisenä kehitysvaiheen tehtävänä oli antaa kaikille aiemmin luoduille käyttäjäryhmille oikeuksia eri tietomalleihin. Suunnitteluvaiheessa oli rajattu, että nämä annetaan myös modulaarisesti ja niiden täytyy vastata tilannetta mikä jokaisen moduulin asennuksen jälkeen on varmasti järjestelmässä olemassa. Tämä tarkoitti, että Collapick Base -moduulissa piti antaa molemmille käyttäjäryhmille halutut oikeudet sen perusteella, mitä Odoon base-moduulissa on käyttäjäryhmille annettu. Puolestaan esimerkiksi Collapick MRP -moduulissa sekä Collapick Base, että Collapick MRP -moduulien käyttäjäryhmille pystyttiin antamaan vain niitä tietomallien käyttöoikeuksia, jotka tulevat järjestelmään asennetuksi Collapick MRP -moduulin manifest-tiedostossa listattujen riippuvuuksien perusteella (kuva 15).

```
# any module necessary for this one to work correctly
'depends': [
    'base',
    'mrp',
    'collapick_base',
],
```

Kuva 15. Kuvakaappaus Collapick MRP -moduulin manifest-tiedostosta, joka näyttää moduulin riippuvuudet (Kuva: Tapio 2022).

Käyttöoikeuksien toteuttaminen eri käyttäjäryhmille kaikkien kuuden moduulin suhteen oli kehitysvaiheen suurin työ. Prosessi vaati erittäin paljon aikaa ja testausta, koska kaikkien käyttöoikeuksien toimivuutta oli testattava sekä moduuli-kohtaisesti, että moduulien eri yhdistelmillä. Testaus toteutettiin manuaalisesti, koska moduuleissa mukana olevat helppokäyttöliittymät olivat testauksen tärkein kohde ja niiden testausta ei pystynyt automatisoimaan ilman, että testauksen toteutukseen olisi kulunut mahdollisesti enemmän aikaa kuin manuaaliseen testaukseen kului. Vaikkakin manuaalinen testaus kulutti paljon aikaa, se myös auttoi löytämään muutamia ongelmakohtia, jotka pystyttiin testauksen jälkeen korjaamaan. Tämä kehitysvaiheen viimeinen tehtävä seurasi hyvin pitkälti kaavaa, jossa ensin tehtiin toteutusta, jonka jälkeen testattiin, ja sitten sama toistettiin taas alusta.

Kun käyttöoikeudet oli saatu kuntoon, käytiin tilannetta läpi yhdessä toimeksiantajan kanssa. Tässä vaiheessa todettiin kehitysvaiheessa toteutuneen tuotoksen sisältö kattavaksi ja oltiin valmiita siirtymään viimeiseen vaiheeseen, jossa



nämä kehitysvaiheen tuotokset yhdistettäisiin viiden moduulin osalta niiden uusimpaan kehityshaaraan versionhallinnassa.

Kehitysvaiheeseen kulunut aika oli toteutuksen alussa arvioitua suurempi. Kehitysvaiheeseen kului aikaa noin 150 työtuntia, mikä oli noin 40 työtuntia enemmän kuin vaiheelle oli suunniteltu. Tämä johtui pääosin käyttöoikeuksien testaamisen ja toteutuksen prosessista, johon kului paljon aikaa.

### **6.3 Viimeistelyvaihe**

Viimeisenä toteutuksen vaiheena oli viimeistelyvaihe, jonka aikataulu oli pakko sovittaa pienelle aikavälille. Tämä johtui pääasiassa siitä, että viiden moduulin kehitysvaiheen muutokset ja lisäykset piti sovittaa niiden kehityshaaroihin, minkä lisäksi mahdolliseen moduuleissa lisääntyneeseen koodiin piti tehdä myös muutoksia. Lisäksi tämä piti toteuttaa nopeasti, että normaali kehitystyö viiden moduulin osalta voi jatkua mahdollisimman pian.

Kehitysvaiheen toteutusten vieni kehityshaaraan oli kuitenkin lopulta hyvin suoraviivainen prosessi, sillä moduulien kehitysvaiheen ja uusimman kehityshaaran sisällöt vastasivat toisiaan suurimmaksi osaksi. Lisämuutokset, joita tarvitsi tehdä, olivat riittävän pieniä, ettei niiden toteutus vienyt liikaa aikaa. Näiden kehitysvaiheen toteutusten vieminen kehityshaaraan vei lopulta alle kaksi kalenteripäivää, jonka jälkeen tapahtui siirtyminen laajempaan testaamiseen yhdessä toimeksiantajan henkilöstön kanssa.

Laajempaan testaukseen osallistuivat toimeksiantajan puolelta kaikki ohjelmoijat, joiden avulla testauksessa saatiin katettua useat eri käyttöjärjestelmät sekä verkkoselaimet. Testauksessa löytyi muutamia kehitysvaiheessa läpi päässeitä puutteita ja näiden puutteiden korjaamiseen kului noin yksi päivä, jonka jälkeen lopputuotoksen käyttöönottoa pystyttiin alkaa suunnittelemaan.

Viimeistelyvaiheeseen kului työn tekijällä aikaa suunnittelun verran, eli noin 20 työtuntia. Muiden toimeksiantajan puolelta testauksessa mukana olleiden

henkilöiden yhteenlaskettu ajankäyttö jäi alle 10 työtuntiin. Viimeistelyvaihe toteutui muihin vaiheisiin verrattuna hyvin nopeassa tahdissa toimeksiantajan tarpeista johtuen.

## **7 Lopullinen tuotos**

Opinnäytetyön toiminnallisen osion lopullisena tuotoksena valmistui yksi Odoomoduuli, jonka nimeksi jäi Collapick Base. Tämän moduulin lisäksi viiteen toimeksiantajan olemassa olevaan moduuliin tehtiin riippuvuus Collapick Base -moduuliin sekä muutoksia useisiin niissä jo aiemmin valmiina olleisiin osiin, kuten valikoihin ja tietomallien käyttöoikeuksiin. Kokonaisuudessa toteutettiin useita käyttäjäryhmiä ja niistä keskeisimpänä on Collapick Base User -käyttäjäryhmä, joka toimii helppokäyttöliittymien käyttämisessä vaadittuna käyttäjäryhmänä.

Alkuperäisessä suunnitelmassa oli myös tavoitteena tehdä erillinen dokumentaatio, jonka avulla toimeksiantaja pystyisi tulevaisuudessa helpommin tekemään vastaavia toteutuksia muihin sen tuottamiin moduuleihin. Lopulta tätä dokumentaatiota ei kuitenkaan toteutettu, sillä työssä toteutetun koodin ja sen kommentoinnin pohjalta pystyy saamaan selkeän kuvan siitä, miten vastaava toteutus pitää tehdä. Näin työn tekijän ja toimeksiantajan päätöksestä erillistä dokumentaatiota ei päädytty tekemään.

### **7.1 Collapick Base -moduuli**

Lopullisen tuotoksen keskeisessä osassa oleva Collapick Base -moduuli ei sisällä montaa käyttöliittymässä näkyvissä olevaa osaa, mutta se muodostaa pohjan, jolle muiden moduulien tarpeita pystytään rakentamaan. Moduulin tiedostomäärä on pieni (liite 3), eikä siihen ole tarkoitus jatkossa tulla kuin tarpeiden vaatimia pakollisia lisäyksiä.

Lopullinen versio Collapick Base -moduulista kantaa versionumeroa 1.0 ja se sisältää seuraavat asiat:

- Päävalikko Collapick Base
- Asetusvalikko Collapick Base
- Kirjautumisen uudelleenohjauksen logiikka
- Uudelleenohjautumisen toiminnon valinta käyttäjä-tietomalliin ja lomakenkymään
- Uudelleenohjautumisen toiminnon ryhmät ja näkymät niiden hallintaa varten
- Moduulikategorian Collapick luonti
- Käyttäjryhmien Collapick Base User ja Collapick Base Manager luonti
- Käyttäjryhmien käyttöoikeuksien määrittely.

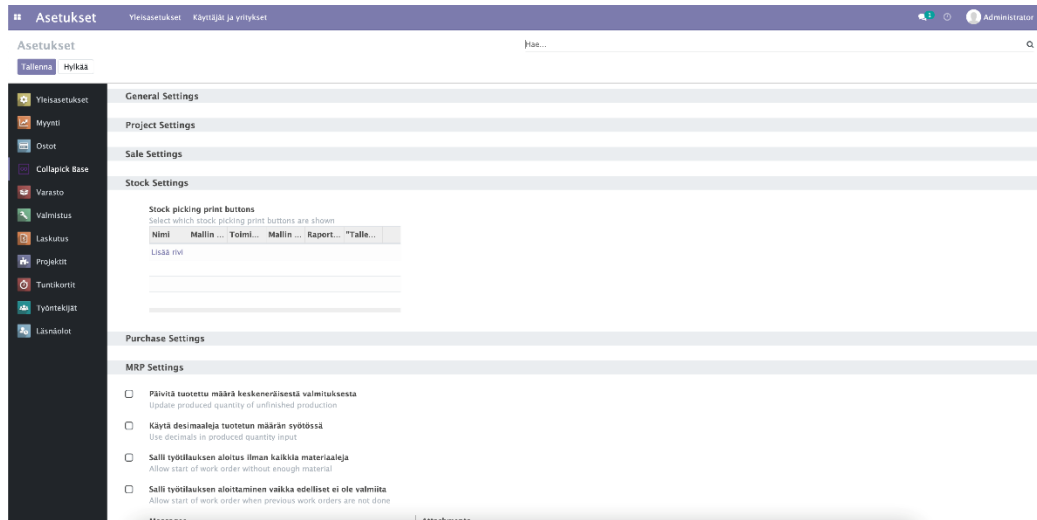
## **7.2 Viisi muuta Collapick-moduulia**

Lopullisen tuotoksen osana olleet viisi muuta moduulia, Collapick MRP, Collapick Stock, Collapick Project, Collapick Purchase ja Collapick Sale, eivät saaneet mitään muuta uutta käyttöliittymässä näkyvää sisältöä kuin Collapick Base -asetusvalikon sisälle laitettut moduulikohtaiset asetusvalikot. Viiden irrallisen moduulin toiminta muuttui kuitenkin olennaisesti, kun niille lisättiin riippuvuus Collapick Base -moduuliin ja niiden olemassa olevia osia muokattiin niin, että moduuleista muodostuva kokonaisuus näyttää yhtenäisemmältä aiempaan verrattuna.

Kaikkiin näistä viidestä moduulista toteutettiin moduulikohtaisesti seuraavat asiat:

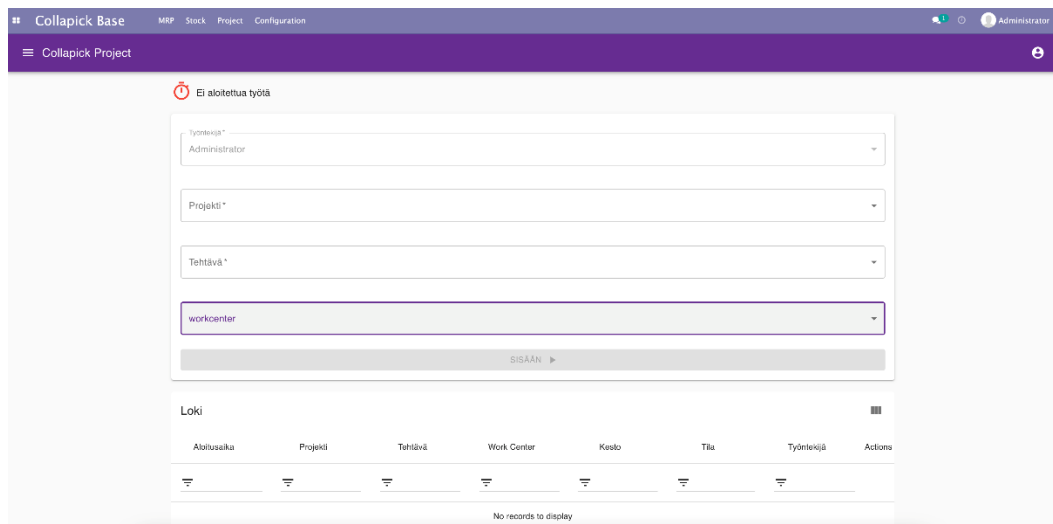
- Moduulikohtaisten käyttäjryhmien luonti
- Collapick Base -moduulin ja moduulin omien käyttäjryhmien käyttöoikeuksien määrittely
- Kaikkien valikkojen vaadittujen pääsyoikeuksien määrittely
- Joidenkin toimintojen, tietomallien ja kenttien vaadittujen käyttöoikeuksien määrittely

- Moduulikohtaisen asetusvalikon luonti Collapick Base -moduulin asetusvalikon sisälle (kuva 16).



Kuva 16. Kuvakaappaus Collapick Base -moduulin asetusvalikosta, jonka sisällä ovat muiden moduulien asetusvalinnat (Kuva: Tapio 2022).

Lisäksi Collapick MRP, Collapick Stock ja Collapick Project moduuleissa siirrettiin helppokäyttöliittymien valikot Collapick Base -valikon alle (kuva 17). Collapick Purchase ja Collapick Sale -moduulit eivät sisällä helppokäyttöliittymiä, joten niissä tälle ei ollut tarvetta.



Kuva 17. Kuvakaappaus Collapick Base -valikon alle laitetuista MRP, Stock ja Project -helppokäyttöliittymistä (Kuva: Tapio 2022).

## 8 Pohdinta

### 8.1 Toteutus ja tulokset

Odoon käyttöoikeudet ovat kriittinen osa tietoturvallista käyttöä ja ne ovat tarpeelliset, että järjestelmä toimii oikein. Modulaarisuuden ja riippuvuuksien takia näissä käyttöoikeuksissa ilmenee paljon päällekkäisyyksiä, jotka on otettava huomioon ja käsiteltävä oikein. Käyttöoikeusryhmät eivät vaikuta pelkästään tiedonhakuun ja -päivitykseen tai toimintojen suorittamiseen, vaan niillä on myös suora vaikutus käyttöliittymässä näkyvään sisältöön. Opinnäytetyössä oli tärkeää olla huolellinen, että käyttöoikeusryhmiä käytetään oikein myös näkymien rajauksen suhteen, jotta valikot tai kentät eivät näy niille, joille niiden ei ole tarkoituskaan näkyä. Vastaavasti oli myös pidettävä huolta, että mikään valikko tai kenttä ei jää tahattomasti täysin piiloon käyttöliittymässä.

Opinnäytetyön tuloksena valmistunut moduuli ja muiden moduulien muutokset ovat tärkeitä Collapickin helppokäyttöliittymien toiminnan kannalta. Niiden kautta muodostuva rakenne helppokäyttöliittymien ja moduulien konfiguroinneille, erityisesti käyttöoikeuksille, toimivat hyvänä pohjana tulevalle kehitystyölle. Valmistuneen moduulin ja muiden moduulien muutosten osalta opinnäytetyössä saavutettiin alussa asetetut tavoitteet.

Alkuperäisessä tavoitteessa oli suunniteltu erillisen dokumentoinnin luontia tulevan kehitystyön helpottamiseksi. Tätä ei kuitenkaan toteutettu vaan toteutuksen aikana muodostunut prosessi jäi toimeksiantajan käyttöön hiljaisena tietona sekä kommenttiriveinä moduulien tiedostoissa. Toteutus erilliselle dokumentoinnille jätettiin toteuttamatta yhteispäätöksenä toimeksiantajan kanssa. Sen tuottama hyödyn määrä ei olisi ollut merkittävästi suurempi kuin tilanteessa, johon tieto prosessista toimeksiantajalla jätettiin.

## 8.2 Oma oppiminen

Opinnäytetyötä tehdessäni opin paljon Odoon-toiminnanohjausjärjestelmästä sekä useista ohjelmointikielistä. Työn aloitusvaiheessa oma osaamiseni Python ja XML-kielten suhteen oli melko olematonta, mutta työn edetessä oma osaamiseni kasvoi merkittäväällä vauhdilla. Myös oma ymmärrykseni modulaarisesta arkkitehtuurista oli olematonta työn alussa, mutta selvitystyön ja käytännön tekemisen jälkeen tunnen ymmärtäväni arkkitehtuuria hyvin. Lisäksi oma osaamiseni ihan perustason käytännön asioista ohjelmistoyrityksen toiminnassa kasvoi tasaiseen tahtiin koko opinnäytetyö prosessin ajan.

Työn aikana jouduin useaan kertaan keskustelemaan toteutuksesta ja sen rajoitteista muille henkilöille, joista kaikilla ei ollut kattavaa ymmärrystä rajoitteista. Sain hyvää harjoitusta siitä, miten selitän asiat muille ihmisille sekä miten perustelen valintojani toteutuksen suhteen. Viimeistelyvaiheessa erityisesti pääsin koordinoimaan isompaa määrää henkilöitä ja ohjaamaan testausprosessia. Tämä oli itselleni uusi kokemus ja aika nopeasti huomasin nauttivani vastavasta roolista.

Opinnäytetyön jälkeen koen myös ymmärtäväni paremmin ohjelmistokehitystä ja sen vaatimuksia. Lisäksi ymmärrän ja osaan käyttää versionhallintaa paremmin. Osaamiseni myös projektin vaiheiden suunnittelusta ja arvioinnista kehittyi ja nautin siitä, kun sain toteuttaa suunnittelua ja arviointia omatoimisesti.

## 8.3 Jatkokehitysmahdollisuudet

Odoon toiminnanohjausjärjestelmän valtavan koon takia on mahdotonta kuvitella, että opinnäytetyössä olisi pystynyt tekemään kaiken kattavan ratkaisun. On kuitenkin hyvä ottaa huomioon mahdollisuudet ja tarpeet jatkokehitykselle, jota toimeksiantaja tulee todennäköisesti tulevaisuudessa opinnäytetyönä syntyvälle koodillekin tekemään. Samasta syystä toteutuksen on oltava suhteellisen helposti toistettavissa tulevaisuudessa.

Koska toteutus toimii modulaarisesti ja toteutusvaiheessa koodiin on lisätty kommentteja sekä levitetty hiljaista tietoa toimeksiantajan suuntaan, on vastavien toteutusten tekeminen uusiin moduuleihin melko suoraviivainen prosessi. Tämä tarkoittaa, että toimeksiantaja pystyy haluamallaan aikataululla muuttamaan muita moduulejaan osaksi modulaarista kokonaisuutta myös ilman opin- näytetyön tekijän osallistumista.

Prosessi itsessään on myös sen verran yksinkertainen, että mahdollisen erillisen dokumentoinnin toteuttaminen on tehtävissä tulevaisuudessa, mikäli toimeksiantaja näin haluaa. Kuka tahansa toimeksiantajan teknisestä henkilöstöstä pystyy nopeasti ymmärtämään ja luomaan noin yhden sivun mittaisen ohjeen prosessin askeleista.

Olin tietoinen jo ennen toiminnallisen osuuden valmistumista, että jatkokehitys-ideoita oli pyöritetty toimeksiantajan suunnalla ja minuakin oli näiden suhteen konsultoitu. Näyttää siis melko varmalta, että oppinäytetyössä valmistunut tuotos alkaa nopeasti kehittymään ja laajentumaan.

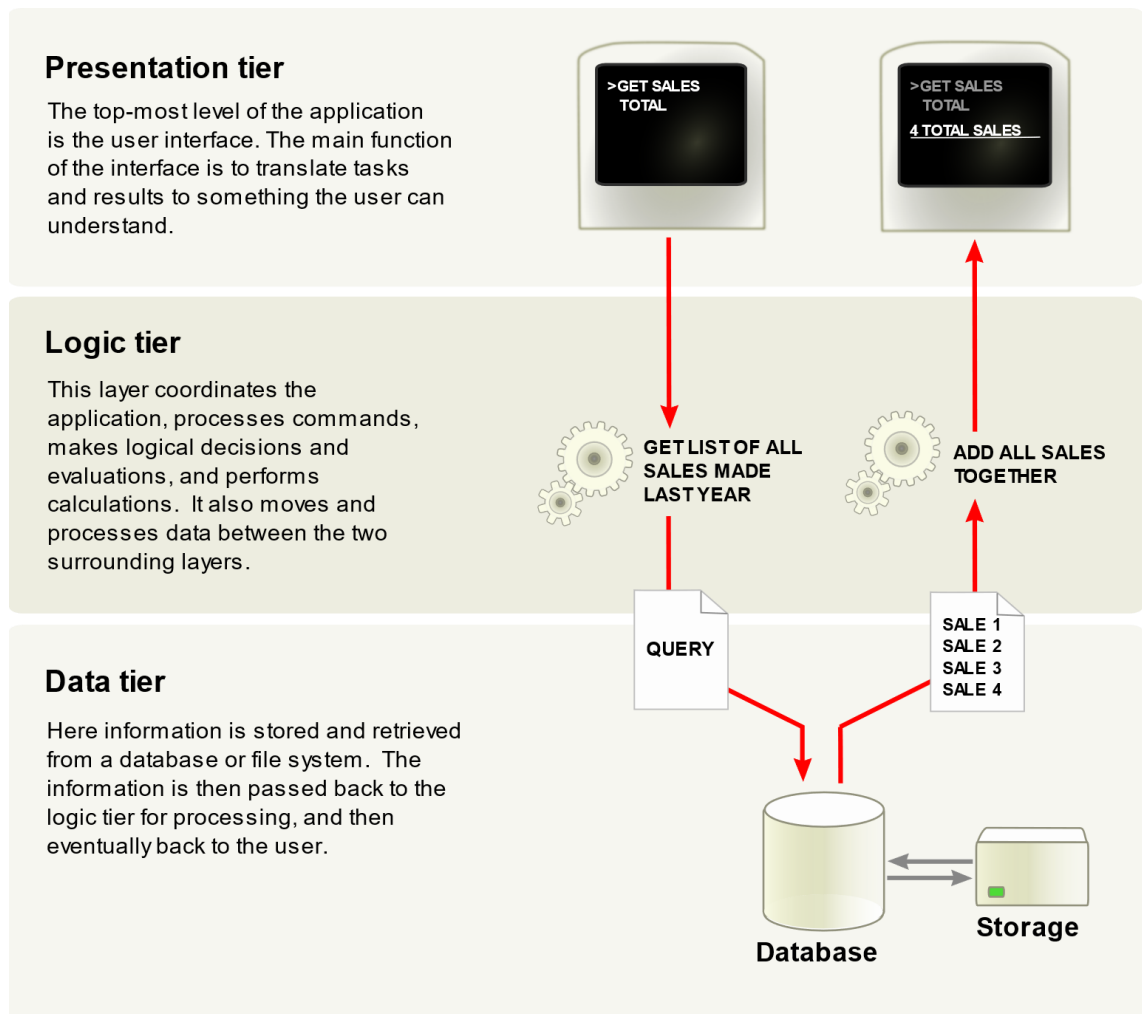
## Lähteet

- Amazon Web Services. 2022. About AWS. Amazon Web Services. <https://aws.amazon.com/about-aws/>. 20.02.2022.
- Collapick Company Oy. 2022a. Collapickin tarina. Collapick Company Oy. <https://collapick.com/yritys/>. 13.02.2022.
- Collapick Company Oy. 2022b. Yhteystiedot. Collapick Company Oy. <https://collapick.com/yhteystiedot/>. 13.02.2022.
- Collapick Company Oy. 2022c. Tutustu töihimme Teollisuus-asiakkaat. Collapick Company Oy. <https://collapick.com/asiakkaat/>. 14.02.2022.
- Collapick Company Oy. 2022d. Collapick Tempo – tuntuvia parannuksia tuotantoon. Collapick Company Oy. <https://collapick.com/collapick-tempo/>. 14.02.2022.
- DBeaver. 2022. About. DBeaver. <https://dbeaver.io/about/>. 20.02.2022.
- Docker. 2022. Use containers to Build, Share and Run your applications. Docker. <https://www.docker.com/resources/what-container>. 20.02.2022.
- Ganesh, A., Sunitha, C., Shanil, K. N. & Midhundas, A. M. 2016. Open-ERP/Odoo – An Open Source Concept to ERP Solution. ResearchGate GmbH. [https://www.researchgate.net/publication/306301008\\_OpenERPOdoo\\_-\\_An\\_Open\\_Source\\_Concept\\_to\\_ERP\\_Solution](https://www.researchgate.net/publication/306301008_OpenERPOdoo_-_An_Open_Source_Concept_to_ERP_Solution). 13.02.2022.
- Odoo Community Association. 2022a. Welcome to the OCA. Odoo Community Association. <https://odoo-community.org/>. 13.02.2022.
- Odoo Community Association. 2022b. Odoo Community Association GitHub. GitHub Inc. <https://github.com/OCA>. 13.02.2022.
- Odoo S.A. 2022a. Chapter 1: Architecture Overview. Odoo S.A. [https://www.odoo.com/documentation/14.0/developer/howtos/rdtraining/01\\_architecture.html](https://www.odoo.com/documentation/14.0/developer/howtos/rdtraining/01_architecture.html). 13.02.2022.
- Odoo S.A. 2022b. Security in Odoo. Odoo S.A. <https://www.odoo.com/documentation/14.0/developer/reference/addons/security.html>. 14.02.2022.
- Odoo S.A. 2022c. About Us. Odoo S.A. <https://www.odoo.com/page/about-us>. 13.02.2022.
- Odoo S.A. 2022d. Odoo Pricing. Odoo S.A. <https://www.odoo.com/pricing>. 13.02.2022.
- Odoo S.A. 2022e. Partner Program. Odoo S.A. <https://www.odoo.com/become-a-partner>. 13.02.2022.
- Odoo S.A. 2022f. Compare Odoo Editions. Odoo S.A. <https://www.odoo.com/page/editions>. 13.02.2022.
- Odoo S.A. 2022g. Release Notes. Odoo S.A. <https://www.odoo.com/page/release-notes>. 13.02.2022.
- Odoo S.A. 2022h. Module Manifests. Odoo S.A. <https://www.odoo.com/documentation/14.0/developer/reference/addons/module.html>. 14.02.2022.
- Odoo S.A. 2022i. Chapter 7: Basic Views. Odoo S.A. [https://www.odoo.com/documentation/14.0/developer/howtos/rdtraining/07\\_basicviews.html](https://www.odoo.com/documentation/14.0/developer/howtos/rdtraining/07_basicviews.html). 13.02.2022.



- Odoo S.A. 2022j. View Types. Odoo S.A. <https://www.odoo.com/documentation/14.0/developer/reference/addons/views.html#view-types>. 13.02.2022.
- Odoo S.A. 2022k. Chapter 4: Models And Basic Fields. Odoo S.A. [https://www.odoo.com/documentation/14.0/developer/howtos/rdtraining/04\\_basicmodel.html](https://www.odoo.com/documentation/14.0/developer/howtos/rdtraining/04_basicmodel.html). 13.02.2022.
- Odoo S.A. 2022l. ORM API. Odoo S.A. <https://www.odoo.com/documentation/14.0/developer/reference/addons/orm.html#>. 13.02.2022.
- Odoo S.A. 2022m. Menus. Odoo S.A. [https://www.odoo.com/documentation/14.0/developer/howtos/rdtraining/06\\_firstui.html#menus](https://www.odoo.com/documentation/14.0/developer/howtos/rdtraining/06_firstui.html#menus). 14.02.2022.
- Odoo S.A. 2022n. WebServices. Odoo S.A. <https://www.odoo.com/documentation/14.0/developer/howtos/backend.html?highlight=rpc#web-services>. 14.02.2022.
- roxannevarza. 2010. OpenERP gets €3 million for US expansion. Yahoo. <https://techcrunch.com/2010/02/17/openerp-gets-e3-million-for-us-expansion/>. 13.02.2022.
- Suomen Asiakastieto Oy. 2022. Collapick Company Oy. Suomen Asiakastieto Oy. <https://www.asiakastieto.fi/yritykset/fi/collapick-company-oy/24150935/yleiskuva>. 13.02.2022.
- Wikipedia. 2021a. Odoo. Wikimedia Foundation Inc. <https://en.wikipedia.org/wiki/Odoo>. 13.02.2022.
- Wikipedia. 2021b. Käytettävyys. Wikimedia Foundation Inc. <https://fi.wikipedia.org/wiki/Käytettävyys>. 14.02.2022.

Liite 1. Monitasoinen arkkitehtuuri visualisoituna.



Wikipedia 2021a; Odoo S.A. 2022a.

Liite 2. Odoo Community- ja Enterprise-version vertailu.

| Odoo 15  | Community | Enterprise |
|--|-----------|------------|
| <b>General</b>   |           |            |
| Unlimited Functional support   | ✗         | ✓          |
| Version upgrades   | ✗         | ✓          |
| Training   | ✗         | ✓          |
| <b>User interface</b>  |           |            |
| Desktop + Web browser  | ✓         | ✓          |
| Mobile + Android and iOS   | ✗         | ✓          |
| <b>Finance</b>   |           |            |
| Accounting   | ✗         | ✓          |
| L Comprehensive Accounting<br><small>General Ledger, Asset-Liability, Accounts Payable, Accounts Receivable, Bank, Budget, Cash, CRM, Credit Control, Customer Invoicing, Expense, Fixed Assets, Inventory, Payroll, Procurement, Reporting, ...</small> | ✗         | ✓          |
| Expenses   | ✓         | ✓          |
| Payroll  | ✗         | ✓          |
| L Expense Digitalization (OCR)   | ✗         | ✓          |
| L Reimbursement in Payroll   | ✗         | ✓          |
| Documents  | ✗         | ✓          |
| L Spreadsheet  | ✗         | ✓          |
| Sign   | ✗         | ✓          |
| <b>Sales</b>   |           |            |
| CRM  | ✓         | ✓          |
| Sales  | ✓         | ✓          |
| Point of Sale  | ✓         | ✓          |
| L Loyalty Programs and Gift Cards  | ✗         | ✓          |
| Subscriptions  | ✗         | ✓          |
| Rental   | ✗         | ✓          |
| Amazon Connector   | ✗         | ✓          |
| <b>Webshops</b>  |           |            |
| Website Builder  | ✓         | ✓          |
| eCommerce  | ✓         | ✓          |
| Blog   | ✓         | ✓          |
| Forum  | ✓         | ✓          |
| Live Chat  | ✓         | ✓          |
| eLearning  | ✓         | ✓          |
| <b>Inventory &amp; MRP</b>   |           |            |
| Inventory  | ✓         | ✓          |
| L Barcode  | ✗         | ✓          |
| Manufacturing (MRP)  | ✓         | ✓          |
| L (Mechanical, Control Panel, Scheduling)  | ✗         | ✓          |
| PLM  | ✗         | ✓          |
| Purchase   | ✓         | ✓          |
| Maintenance  | ✓         | ✓          |
| Quality  | ✗         | ✓          |
| <b>Human Resources</b>   |           |            |
| Employee   | ✓         | ✓          |
| L Departmental Dashboard   | ✗         | ✓          |
| Recruitment  | ✓         | ✓          |
| Time Off   | ✓         | ✓          |
| Benefits   | ✗         | ✓          |
| Appraisals   | ✗         | ✓          |
| Fleet  | ✓         | ✓          |
| <b>Marketing</b>   |           |            |
| Social Marketing   | ✗         | ✓          |
| Email Marketing  | ✓         | ✓          |
| L Mail Marketing   | ✗         | ✓          |
| SMS Marketing  | ✓         | ✓          |
| Events   | ✓         | ✓          |
| Marketing Automation   | ✗         | ✓          |
| Surveys  | ✓         | ✓          |
| <b>Services</b>  |           |            |
| Project  | ✓         | ✓          |
| Timesheet  | ✓         | ✓          |
| L CRM (Lead, Email, Website, LinkedIn, Facebook)   | ✗         | ✓          |
| L Response Timepoint (Mobile App)  | ✗         | ✓          |
| Field Service  | ✗         | ✓          |
| Helpdesk   | ✗         | ✓          |
| Planning   | ✗         | ✓          |
| Appointments   | ✗         | ✓          |
| <b>Productivity</b>  |           |            |
| Discuss  | ✓         | ✓          |
| Approvals  | ✗         | ✓          |
| Notes  | ✓         | ✓          |
| Calendar   | ✓         | ✓          |
| Digital Phone (VoIP) integration   | ✗         | ✓          |
| IoT  | ✗         | ✓          |
| <b>Customization</b>   |           |            |
| Studio   | ✗         | ✓          |

Liite 3. Collapick Base -moduulin kansiorakenne ja tiedostot.

