

# ConExting-keskusteluseinän toteuttaminen Odoo-tekniologialla



Ammattikorkeakoulututkinnon opinnäytetyö

Tietojenkäsittelyn koulutusohjelma

Hämeenlinna, kevät 2018

Valtteri Lattu

Tietojenkäsittelyn koulutusohjelma  
Visamäki

---

<b>Tekijä</b>	Valtteri Lattu	<b>Vuosi</b> 2018
<b>Työn nimi</b>	ConExting-keskusteluseinän toteuttaminen Odoo-tekniologi- alla	
<b>Työn ohjaaja/t</b>	Lasse Seppänen	

---

## TIIVISTELMÄ

Opinnäytetyössä tutkittiin avoimeen lähdekoodiin pohjautuvaa Odoo-toiminnanohjausjärjestelmää (Enterprise Resource Planning, ERP), sekä sitä, miten Odoota hyödyntäen toteutettiin ConExting-keskusteluseinä. Opinnäytetyön tavoitteena oli antaa lukijoille selkeä käsitys Odoon toiminoista, käytettävyydestä ja siitä miten ConExting-keskusteluseinä ohjelmoitiin ja toteutettiin.

Työn toimeksiantajana toimi Tawasta OS Technologies Ltd, joka kuuluu Mindpolis Group yritysryhmään. Nykyinen ConExting-keskusteluseinä on toteutettu Yii-ohjelmistokehitysalustalla, jonka ylläpidettävyys sekä jatkokehitys on todettu haasteelliseksi. Uusi keskusteluseinä päädyttiin toteuttamaan Odoo-tekniologialla, jota työn toimeksiantaja käyttää pohjaratkaisuna useissa verkkopalveluissa ERP-toteutuksien lisäksi.

Opinnäytetyölle asetetut tavoitteet täytettiin ja ConExting-keskusteluseinästä toteutettiin opinnäytetyön aikana tuotteen pilottiversio uudella tekniologialla. Valmis verkkopalvelu julkaistaan loppukesästä 2018. Verkkopalvelussa toteutettiin kaksi erilaista moduulia, jotka toimeksiantajan on mahdollista hyödyntää tulevaisuudessa muissa projekteissa, asiakkaan tarpeen mukaan.

**Avainsanat** Odoo, ohjelmointi, toiminnanohjausjärjestelmä, ConExting-keskusteluseinä, Python

**Sivut** 27 sivua, joista liitteitä 0 sivua

Degree Programme in Business Information Technology  
Visamäki, Hämeenlinna

---

<b>Author</b>	Valtteri Lattu	<b>Year</b> 2018
<b>Subject</b>	Implementing the ConExting discussion service with Odoo Technology	
<b>Supervisors</b>	Lasse Seppänen	

---

ABSTRACT

In this thesis the idea was orientate to Odoo Enterprise Resource Planning (ERP) which is based an open source code and also how ConExting discussion board is implemented. The aim of the thesis was to give readers a clear understanding of Odoo's functionality, usability and how the ConExting discussion board was programmed and implemented.

The work was commissioned by Tawasta OS Technologies Ltd, which is member of the Mindpolis Group of companies. The current ConExting discussion platform has been implemented on the Yii software development platform, where maintainability and sequential development has been identified very challenging. The new discussion board came to be implemented with Odoo technology, which is commissioned by the client because they use Odoo for a number of web services in addition to ERP implementations.

The all goals what have been set for the Thesis were fulfilled and the ConExting discussion board was implemented a pilot version with the new technology. The finished web service will be published in the late summer 2018. Two different modules were implemented in the web service that the repository will be able to use in future projects in other projects, in accordance with the customer's need.

**Keywords** Odoo, programming, Enterprise Resource Planning, ConExting –discussion service, Python

**Pages** 27 pages including appendices 0 pages

# SISÄLLYS

1	JOHDANTO.....	1
2	AIHEEN ESITTELY.....	2
2.1	Toimeksiantaja .....	2
2.2	Työn tausta.....	2
3	ODOO.....	4
3.1	Toiminnanohjausjärjestelmästä yleistä .....	4
3.2	Sisällönhallintajärjestelmä .....	5
3.3	Odoo-teknologia.....	5
3.3.1	Historia .....	6
3.3.2	Toiminnallisuus.....	7
3.3.3	Git-versionhallinta .....	7
3.3.4	Avoin lähdekoodi.....	8
3.3.5	Modulaarisuus .....	8
3.3.6	Laajennettavuus .....	9
3.3.7	Räätälöitävyys.....	10
3.3.8	Lokalisointi.....	10
3.3.9	Käyttöoikeudet .....	11
4	CONEXTING-KESKUSTELUSEINÄN TOTEUTTAMINEN .....	12
4.1	Määrittely.....	12
4.2	Odoo-järjestelmän asennus .....	13
4.3	Keskusteluseinä-moduulin luonti.....	14
4.3.1	Moduulin rakenne .....	15
4.3.2	Käyttäjänäkymät.....	16
4.3.3	Discuss-moduulin hyödyntäminen .....	17
4.3.4	Website twitter moduulin hyödyntäminen.....	19
4.3.5	Tuetut stream palvelut .....	20
4.3.6	ConExting-keskusteluseinän ulkoasu .....	21
5	KÄYTTÖÖNOTTO.....	23
6	YHTEENVETO .....	24
6.1	Tulokset.....	24
6.2	Haasteet .....	24
6.3	Jatkokehitysmahdollisuudet .....	25
7	LÄHTEET.....	26

## SANASTO

<b>ERP</b>	Toiminnanohjausjärjestelmä (Enterprise Resource Planning)
<b>CRM</b>	Asiakkuudenhallinta (Customer relationship management)
<b>Odoo</b>	Avoimeen lähdekoodiin perustuva yritysohjelmisto
<b>CMS</b>	Sisällönhallintajärjestelmä (Content management system)
<b>Python</b>	Ohjelmointikieli, jossa on yksinkertainen syntaksi ja korkean tason tietorakenteet
<b>XML</b>	Merkintäkielen standardi, jota käytetään tiedonvälitykseen sekä tallentamiseen
<b>HTML</b>	Standardoitu kuvauskieli (Hypertext Markup Language), jota eritoten käytetään verkkosivustojen luomisessa
<b>JSON</b>	Avoimen standardin tiedostomuoto, jota käytetään tiedonvälityksessä.
<b>LESS</b>	Tyyli-tiedoston laji, joka mahdollistaa erilaisia muuttujia, luokkia sekä toimintoja täydentääkseen CSS-merkintäkieltä
<b>Frontend</b>	Tarkoitetaan koodia, jota ajetaan verkkoselaimessa, jonka käyttäjät näkevät käyttäessään sivustoa
<b>Backend</b>	Taustalla oleva järjestelmä, jossa hallinnoidaan sivuston asetuksia
<b>Bootstrap</b>	Avoimen lähdekoodin työkalu HTML-, CSS- ja JS kehitystyön kehittämiseen

## 1 JOHDANTO

Tämän opinnäytetyön tarkoituksena on suunnitella sekä toteuttaa ConExting-keskusteluseinä Odoon-tekniikalla. Odoon on avoimen lähdekoodin liiketoimintaohjelmisto, joka sisältää sekä toiminnanohjaus- että sisällönhallintajärjestelmän. Toiminnanohjausjärjestelmällä yritys voi hallinnoida kaikkia prosessejaan keskitetysti, kun taas sisällönhallintajärjestelmä mahdollistaa verkkosivujen ja verkkokaupan hallinnan. Opinnäytetyössä tullaan perehtymään Odoon eri teknisiin ratkaisuihin ja erityisesti Odoon tarjoamaan discuss-lisäosaan, jota tullaan hyötykäyttämään ConExting-keskusteluseinän toteuttamisessa.

Opinnäytetyöhön on koottu tietoa Odoon-järjestelmän toiminnallisuuksista ja ominaisuuksista. Tutkielmassa kerrotaan, miten Odoon ohjelmoidaan ja mitä ohjelmoinnissa tulee ottaa huomioon. Ensimmäisissä kappaleissa kerrotaan, millainen Odoon on järjestelmänä ja miten se eroaa muista toiminnanohjausjärjestelmistä. Lisäksi käsitellään Odoon modulaarisuuteen, räätälöintiin sekä avoimen lähdekoodin vaikutukseen Odoon kehittämisessä.

Opinnäytetyön keskiosassa sekä loppuosassa tutustutaan tarkemmin ConExting-keskusteluseinän toteuttamiseen. Keskiosa keskittyy keskusteluseinän toteuttamiseen vaadittaviin ominaisuuksiin ja elementteihin, ja loppuosassa kerrotaan tuloksista ja mahdollisista jatkokehityksistä.

Opinnäytetyö tulee vastaamaan alla oleviin kysymyksiin:

- Mikä on Odoon?
- Mikä on toiminnanohjausjärjestelmä?
- Miten Odoon ohjelmoidaan?
- Kuinka Odoon luodaan uusi moduuli ja miten sitä kehitetään?
- Mikä on ConExting-keskusteluseinä?

## 2 AIHEEN ESITTELY

Tässä luvussa kerrotaan ConExting-keskusteluseinän toteutuksesta, toimeksiantajasta sekä työn taustatiedoista. Luvussa kerrotaan, miksi ConExting-keskusteluseinä tullaan toteuttamaan Odoo-tekniologialla, projektin aikana käytettävistä moduuleista ja järjestelmistä sekä osittain myös ohjelmointikielestä. Osiossa käydään läpi myös lyhyesti nykyisen ConExting-verkkopalvelun toteutustapa.

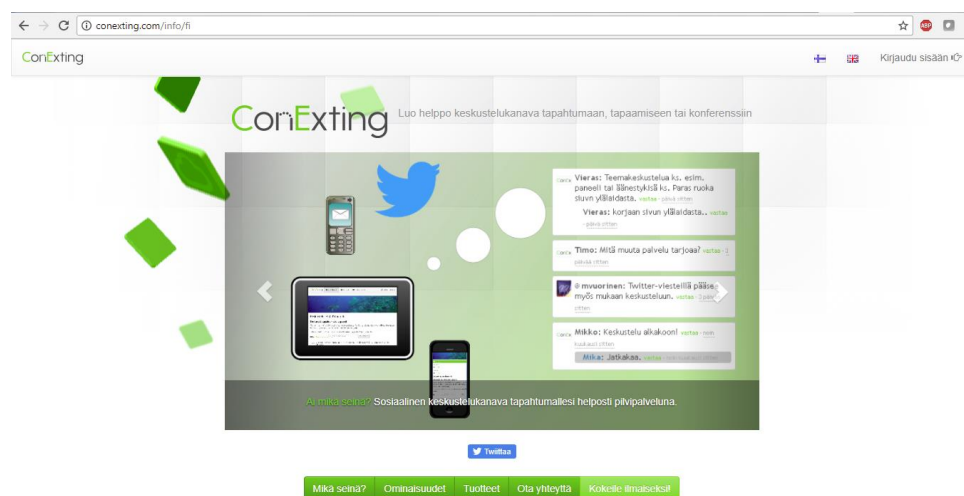
### 2.1 Toimeksiantaja

Toimeksiantajana toimii Oy Tawasta OS Technologies Ltd. (myöhemmin Tawasta), joka kuuluu Mindpolis Group yritysyhmään. Mindpolis Group yritysyhmä työllistää noin 35 henkilöä ja sillä on toimipisteet Tampereella, Helsingissä, Turussa ja Hämeenlinnassa.

Tawasta on 2004 perustettu avoimen lähdekoodin palveluita tuottava asiantuntijayritys. Yrityksen pääpalveluihin kuuluvat Joomla!, Magento commerce, Odoo -järjestelmillä toteutetut verkkopalvelut sekä palveluluiden tuki- ja ylläpitopalvelut SAAS-palveluna.

### 2.2 Työn tausta

Vuoden 2017 loppupuolella toimeksiantaja antoi projektin, jossa on tarkoitus toteuttaa ConExting-verkkopalvelu toimeksiantajan tuotevalikoimaan kuuluvalla Odoo-tekniologialla. ConExting-verkkopalvelu on alun perin toteutettu 2010–2012 Yii-ohjelmistokehitysalustalla, ja sen ylläpidettävyys sekä jatkokehitys on todettu erittäin haasteelliseksi. Osa verkkopalvelun ominaisuuksista on myös havaittu tarpeettomiksi (Kuva 1).



Kuva 1. Nykyinen ConExting-palvelu.

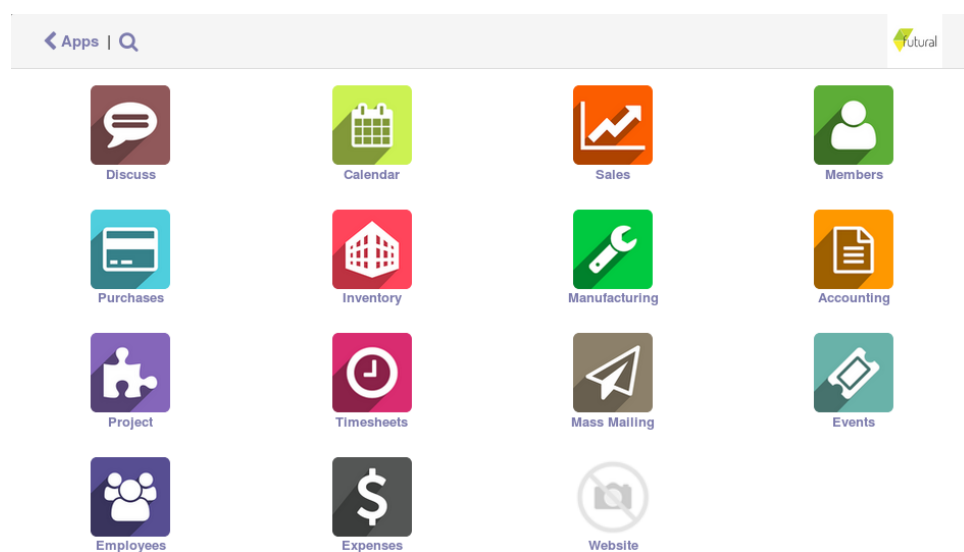
Työharjoittelun aikana opinnäytetyön toteuttaja pääsi tutustumaan pinta-puolisesti Odooseen ja sen mahdollisuuksiin, modulaarisuuteen ja muokattavuuteen. Ennen palvelun toteutusta toimeksiantajan ohjelmistoasi-antuntijat hyväksyivät tutkimusprosessissa ehdotetun toteutustavan. To-teutuksessa tullaan hyödyntämään isolta osin Odoon Discuss-lisäosaa, jonka ympärille seinän ominaisuudet tullaan rakentamaan sekä toteutta-maan myös julkiseen käyttöliittymään.

ConExting-verkkopalvelun toteuttaminen tulee olemaan erittäin haasteel-linen, koska toteuttajalla ei ole aikaisempaa kokemusta Odoosta eikä Pyt-hon-ohjelmointikielestä, kuten ei myös yleisestikään ohjelmoinnista ja ke-hittämisestä siinä laajuudessa, mitä toteutus tulee vaatimaan.



### 3 ODOO

Odoo on belgialaisen Odoo S.A:n toteuttama avoimen lähdekoodin toiminnanohjaus- ja sisällönhallintajärjestelmä sekä ohjelmistoalusta kehittäjille. Odoo on aikaisemmin tunnettu nimillä OpenERP sekä TinyERP. Odoo on huomionnut omassa kehityksessään, että järjestelmän tulisi soveltua niin pienille kuin suurille yrityksille. Odoon modulaarisuus ja muokattavuus mahdollistaa asiakkaiden tarpeiden mukaiset räätälöidyt ratkaisut helposti ja ketterästi (Kuva 2). Odoon aikaisempien versioiden ominaisuudet ja toiminnallisuudet tulisi toimia vaivattomasti uusimpien ja tulevien versioiden kanssa. Erityisesti Odoon viimeisimmässä versioissa on ollut paljon uusia ominaisuuksia. (Odoo development cookbook 2018, 16.)



Kuva 2. Odoon oletusmoduulit.

#### 3.1 Toiminnanohjausjärjestelmästä yleistä

Toiminnanohjausjärjestelmä, joka tunnetaan myös nimellä ERP (Enterprise Resource Planning), on yrityksille suunnattu tietojärjestelmä. Se sisältää mm. kirjanpitoa, laskutusta, varastonhallintaa, tuotannonohjausta kuin prosessien, materiaalien ja resurssien hallintaa. Nykyiset ERP-järjestelmät on mahdollista muokata yrityksen ja henkilöiden tarpeiden mukaisesti. Muokattavuuden mahdollistavat järjestelmästä löytyvät moduulit, jotka vahvistavat ERP-asennuksen kokonaisuutta. Toiminnanohjausjärjestelmiin on nykyään mahdollista liittää muitakin järjestelmiä, kuten asiakkuudenhallintajärjestelmä, joka tunnetaan myös nimellä CRM (customer relationship management).

Toiminnanohjausjärjestelmillä pystytään lisäämään yrityksen tehokkuutta sekä toiminnallisesti että taloudellisesti. Järjestelmän isoimpia vahvuuksia

on eri toimintojen sekä järjestelmien välisien tietojen tallentaminen yhteen paikkaan, mikä tehostaa yrityksen toimintaa kun, kaikki eri osastot saavat tiedot samaan aikaan. (Accountorenterprise n.d.)

### 3.2 Sisällönhallintajärjestelmä

Sisällönhallintajärjestelmä, joka tunnetaan myös nimellä CMS (Content Management System), on verkkosivuihin suunnattu työkalu, jonka avulla sivuston ylläpito sekä erilaisten sisältöjen luominen ja muokkaaminen onnistuu erittäin helposti selaimen avulla ilman teknistä osaamista. Sisällönhallintajärjestelmää voidaan kutsua myös julkaisujärjestelmäksi. (Omnipartners 2018.)

Sisällönhallinta on järjestelmä, jonka tavoitteena on tarjota koko yrityksen organisaatiolle sisällönhallintaa, joka ei ole keskittynyt vain yhteen asiaan tai osa-alueeseen. Erilaiset sisällönhallintajärjestelmät vaihtelevat toisistaan, mutta yleensä ne jaetaan kolmeen eri osa-alueeseen: koostamisjärjestelmään, sisältökokoelmaan tai tietovarastoon sekä yllämainittuun julkaisujärjestelmään. (Webopas 2018.)

Odoo 10 tarjoaa käyttäjille Website Builder -lisäosan, jolla on mahdollista muokata sivustoa ilman, että tekijä olisi vuorovaikutuksessa suoraan sivuston sisäiseen toimintaan. Tätä työkalua kutsutaan Odoossa yleisesti CMS:ksi. (Working with odoo 10 second edition, 247.)

### 3.3 Odoo-teknologia

Odoossa on olemassa kaksi pääkehittäjää Odoo S.A sekä Odoo OCA, jotka ovat perustettu tukemaan ja kehittämään Odoo -yhteisön ja käyttäjien tarpeita. Ohjelmiston kehittäjänä Odoo S.A tarjoaa ohjelmiston vapaasti ladattavaksi, mutta Odoosta on tarjolla myös maksullinen SAAS-versio (Kuva 3). Tällä hetkellä Odoon ylläpidetyt versiot ovat 9.0 (10/2015), 10.0 (10/2016) sekä 11.0 (10/2017).

Strengths <input type="checkbox"/> Insert	Weaknesses <input type="checkbox"/> Insert
Odoon erittäin laaja yritysohjelmisto	Suomessa vielä vähän tunnettu järjestelmä
Odoolla löytyy todella kattava kehitysyhteisö	Suomalaiset referenssit ovat pienet
Toimittajarippumaton	
Avoimen lähdekoodin ohjelmisto	
Opportunities <input type="checkbox"/> Insert	Threats <input type="checkbox"/> Insert
Lisenssi vapaa sekä toimittajarippumaton, joka mahdollistaa ERP-järjestelmien joustavuuden	Useita toimittajia, jotka perustuvat edelleen suljettuun järjestelmään
Mahdollistaa digitalisoinnin yrityksille joilla ei ole ennen ollut siihen mahdollisuutta	Menestys riippuu käyttäjäkunnan vastaanottamisesta
Avoimen lähdekoodin ansiosta mahdollisuus integroida muihin järjestelmiin ja täten luoda uusia digitaalisia alustoja	Markkina kilpailussa vaikea ennustaa

Kuva 3. SWOT-analyysi toiminnanohjausjärjestelmä Odoosta.

Odoon on toteutettu Python-ohjelmointikielellä ja järjestelmä on tarkoitettu kaiken kokoisille yrityksille. Odoon perustuu PostgreSQL-tietokantaan, joka toimii järjestelmän taustana sovelluspalvelimena. Odoon on hankkinut erittäin laajan ja menestyvän tuki- ja kehitysyhteisön, joka tarjoaa apuaan erilaisten ongelmien kanssa ja kehittää uusia moduuleja järjestelmään. Odoon tarjoaa myös kattavan dokumentaation, jossa kerrotaan esimerkkien avulla järjestelmän toiminnallisuuksista.

Järjestelmän laajuus ja modulaarisuus sekä sen muokattavuus asiakkaan tarpeiden mukaisesti tekee Odoosta kilpailukykyisen ja toimivan ratkaisun yrityksille. Odoon sisältää muun muassa laskutuksen, kirjanpidon, projektinhallinnan sekä varastohallinnan. Nykyään Odoon ei ole enää pelkästään ERP-järjestelmä, vaan se sisältää paljon muitakin erilaisia työkaluja, joita voidaan asentaa järjestelmään ja ottaa yritysten käyttöön. (Archlinux.org 2018.)

Odoon-toiminnanohjausjärjestelmä on rakennettu noudattamaan MVC-ohjelmistoarkkitehtuuria, jossa ohjelmisto tullaan jakamaan kolmeen osaan eli malliin, käsittelijään ja näkymään. MVC-arkkitehtuurin ensisijaisia tavoitteita on erottaa yrityksen visuaalinen näkymä liiketoimintasäännöistä ja taustalla olevien tietojen hallinnoinnista. Tämän avulla käyttäjä voi muuttaa erilaisten tietojen järjestystapaa malleissa ilman, että joudutaan tekemään isoja muutoksia näkymien puolella. (Working with odoo 10 second edition, 17.)

### 3.3.1 Historia

Belgialaissyntyisellä Odoon perustajalla Fabien Pinckaersilla oli nuorena unelmana luoda yritys, joka johtaisi yrityksiensä hallinnan markkinoita avoimeen lähdekoodin perustuvalla ohjelmistolla sekä saada kokoon noin 100 työntekijää. Vuonna 2005 Fabien alkoi kehittämään TinyERP -ohjelmistoa

tehdn 13 tunnin työpäiviä ja kolmen vuoden päästä Pinckaers päätti nimetä TinyERP:in uudelleen OpenERP -nimiseksi. Suurin syy järjestelmän nimen vaihtoon oli se, että hän ei voi muuttaa maailmaa, jos yritys ja yrityksen nimi kertoo pienestä. Jo tässä vaiheessa järjestelmä oli kasvanut ja kehittynyt huomattavasti, koska OpenERP sisälsi jo kymmeniä moduuleja ja avoimen lähdekoodin yhteisö kasvoi todella merkittävästi.

Vuonna 2010 Pinckaers oli kerännyt kokoon noin 100 työntekijää, jotka myivät OpenERP palveluita. Samaan aikaan Pinckaers päätti muuttaa koko liiketoimintamallia, lopetti palvelujen tuottamisen asiakkaille ja alkoi keskittyä erittäin vahvasti kumppaniverkon ja ylläpidon rakentamiseen. Tämä oli erittäin merkittävä päätös Odoon kannalta, koska Odoo on kasvanut melkoisella vauhdilla ja moduuleja tuotetaan koko ajan lisää. Vuonna 2014 maaliskuussa OpenERP nimettiin uudelleen Odooksi. (Odoo n.d.)

### 3.3.2 Toiminnallisuus

Odoo-toiminnanohjausjärjestelmän toiminnallisuuden ydin perustuu täysin moduuleihin. Juuri asennettu odoo-järjestelmä on hyvin suppea toiminnallisuuksiltaan, sillä asennuspaketti sisältää vain muutaman valmiin moduulin. Odoo kuitenkin mahdollistaa monien valmiiden lisäosien asennuksen järjestelmään omasta sovelluskirjastosta sekä järjestelmään on mahdollista asentaa myös ulkopuolisten kehittäjien luomia moduuleja.

Avoimeen lähdekoodin perustuva Odoo-järjestelmä tarjoaa erittäin paljon ulkopuolisten kehittäjien sekä kehittäjäyhteisön luomia lisäosia. Github on yksi isoimmista kolmannen osapuolen lisämateriaali pankeista, jossa myös Odoo OCA ylläpitää satoja eri moduuleja. (Odoo development cookbook 2018, 59-63.)

### 3.3.3 Git-versionhallinta

Git on versionhallintajärjestelmä, johon projektin kaikki tiedostot ovat mahdollista tallentaa sekä jälkeenpäin palata haluttuun tallennus kohtaan korvaten viimeksi tehdyt muutokset. Versionhallinnassa on mahdollista tallentaa muutoksia tiedostoihin tai vain yksittäiseen tiedostoon.

Ohjelmoijille sekä web-suunnittelijoilla versionhallinta on erittäin hyvä tallennuspaikka projektille. Versionhallinnassa on mahdollista katselmoida muokkaushistoriaa ja ongelman ilmetessä tutkia ongelman aiheuttajaa. Versionhallinnassa on mahdollista katsoa myös muiden henkilön tekemiä muokkauksia.

Github on verkkosivusto palvelu, joka tarjoaa Git-versionhallintaa kaikille sitä käyttäville ohjelmistokehittäjille, jonne voidaan tallentaa kaikki projektit. Git itsessään on tarkoitettu komentoriville, mutta Github tarjoaa käyttäjille kokonaan graafisen käyttöliittymä ympäristön. Toinen erittäin toimiva versionhallinta palvelu on GitLab, joka on aivan vastaavanlainen toiminnollisuuksiltaan kuin Github. (Git-scm.com n.d.)

### 3.3.4 Avoin lähdekoodi

Avoin lähdekoodi on menettelytapa, joka tarkoittaa sitä, että ohjelmoitu lähdekoodi on vapaasti käytettävissä sekä jaettavissa. Henkilö, joka käyttää avointa lähdekoodia, voi vapaasti myös kopioida sekä muokata avoimen lähdekoodin ohjelmaa ilman, että siitä tulisi käyttäjälle erillisiä kustannuksia. Avoimessa lähdekoodissa ei ole olemassa vain yhtä henkilöä tai yritystä, vaan kehittäjinä toimii koko avoimen lähdekoodin yhteisö ja monet yritykset, joka myös edesauttaa ohjelmistojen joustavuutta sekä vähentää huomattavasti riskejä. Erityisesti laajan kehittäjäyhteisön ihmiset auttavat ohjelmistoja olemaan alusta alkaen todella laadukkaita. Lisäksi ohjelmistojen tietoturva on myös parempi ja ohjelmistojen sekä sovellusten yhteentoimivuus muiden järjestelmien kanssa on mahdollista. (coss.fi n.d.a.)

Avoimessa lähdekoodissa ei ole olemassa yhtä tarkkaa määriteltyä termiä, jolla se voitaisiin kertoa. Avointa lähdekoodia on kutsuttu sekä open source että free software nimillä. Eri sektoreiden näkökulmasta termeillä ei ole huomattavia eroja ja molemmat termit sisältyvät samaan käyttöön. Avoimen lähdekoodin selkeimmät vaatimukset ovat, että ohjelma tulee olla vapaasti käytettävissä sekä jaettavissa. Samoin ohjelmien lähdekoodi tulee olla muille ihmisille vapaasti saatavilla eikä käyttötarkoituksia saa lainkaan rajoittaa. (coss.fi n.d.b.)

Avoin lähdekoodi sisältää kymmeniä eri lisenssejä. Lisenssien suurin ero avoimessa lähdekoodissa on niiden säännöissä kerrotut muuntelu sekä yhdistämismahdollisuudet muihin moduuleihin. (coss.fi n.d.b.)

Odoon käyttää LGPL-lisenssiä, jonka tarkoituksena on mahdollistaa ohjelman vapaa levittäminen samalla tavalla kuin GPL:n ohjelman. Vaikka järjestelmässä käytettäisiin suljettua ohjelmaa, niin lisenssi mahdollistaa erilaisten vapaiden kirjastojen käytön. (Linux.fi n.d.)

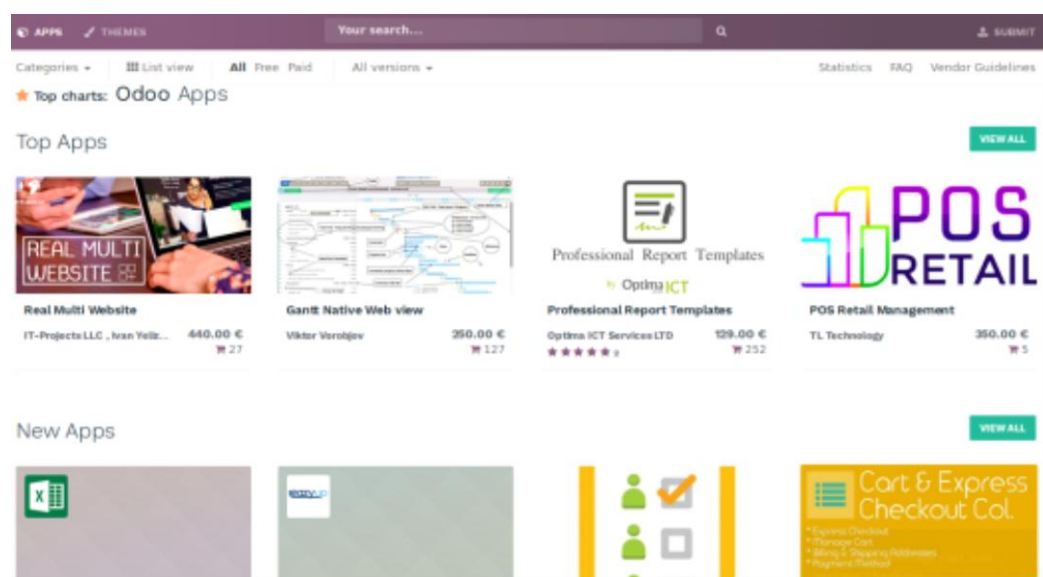
### 3.3.5 Modulaarisuus

Odoon modulaarisuus mahdollistaa tuotteen muokattavuuden että räätälöinnin yrityksille nopeasti ja vaivattomasti yrityksen tarpeen mukaisesti. Odoon alkuperäisasennus sisältää 30 moduulia, jotka kaikki ovat Odoon S.A:n tuottamia moduuleja. Tämän lisäksi Odoossa on yli 4500 moduulia,

jotka ovat eri yritysten, henkilöiden tai Odoon kehittäjäyhteisön luomia lisämoduuleja.

Odoon moduuleista suurin osa on saatavilla ohjelmiston omilta kotisivuilta (Kuva 4). Osa kyseisistä sovelluksista ovat maksullisia, mutta kuitenkin suurin osa on ilmaisia käyttäjille ja kuka tahansa pystyy sen lataamaan. Kehittäjät voivat jakaa itse tekemiään ja kehittämiään moduuleita myös Odoon kotisivuilla.

Odoossa on tärkeää ymmärtää modulaarisuudesta puhuttaessa käsitteet sovellus sekä moduuli ja niiden erot. Sovellus tarkoittaa Odoossa korkeamman tason komponenttia, johon kuuluu monia erilaisia moduuleja. Moduuli on taas komponentti, jonka tarkoituksena on toteuttaa jonkinlainen toiminto ja laajentaa olemassa olevaa sovellusta. (Odoon n.d.)



Kuva 4. Odoon omia tarjoamia moduuleja.

### 3.3.6 Laajennettavuus

Odoon avoin lähdekoodi mahdollistaa järjestelmän laajennettavuuden ja muokattavuuden asiakkaan tarpeen mukaisesti. Kehittäjäyhteisöön luotuja moduuleja löytyy satoja kappaleita, mutta ennen käyttöönottoa tulee aina varmistaa, että ne toimivat oikein ja ovat laadukkaasti toteutettuja. Nimenomaan moduulien laadussa sekä toiminnallisuuksissa löytyy huomattavia eroja. Osa moduuleista on saattanut olla jo kauan olemassa ja ovat yhteensopivia muiden versioiden kanssa, mikä tekee moduulista taakkaamman ja luotettavamman.

Jonkun toisen tuottamaa moduulia asennettaessa tai ostettaessa Odoon tarjoaa mahdollisuuden asentaa omalle tietokoneelle oma kehitysympäristö, jossa on mahdollista testata kyseistä moduulia ennen päätöksen tekoa ja varmistaa, onko moduuli juuri sitä, mitä oletettiin. Kaikki toiminnallisuudet on myös suositeltavaa pilotoida ennen varsinaista käyttöönottoa

ja moduuleita voidaan laajentaa sekä muokata tarpeen mukaan myös käyttöönoton jälkeen. (Itewiki 2018.)

### 3.3.7 Räätelöitävyys

Räätelöitävyys tarkoittaa usein yritykselle lisäkuluja, koska ohjelmistoa muokataan toimimaan tietyn tarpeen mukaan. Monet yritykset käyttävät mahdollisimman paljon out of the box -ratkaisuja ja muokkaavat toimintatapaansa ohjelmiston mukaisesti, jos mahdollista. Monilla yrityksillä ja organisaatioilla saattaa olla todella uniikkeja prosesseja, jotka vaativat räätälöintiä ohjelmistoon, jotta heidän työprosessit toimivat vaivattomasti arjessa. Tämän vuoksi ERP-järjestelmä, joka mahdollistaa räätälöinnin, on paras ratkaisu uniikkeihin liiketoiminnan tarpeisiin. Monesti ajatellaan, että kaikilla yrityksillä on samanlaiset prosessit, mutta lopulta mikään yritys ei toimi täysin samalla tavalla kuin toinen.

Räätelöity Odoo voi tuoda yritykselle paljon etuja. Voidaan automatisoida prosesseja, jotka tehostavat yrityksen liiketoimintamallia. Kun Odoo ERP on muokattu asiakkaan tarpeen mukaisesti, se tulee vähentämään turhautumisen todennäköisyyttä ja on sitä kautta investointi yritykselle, koska järjestelmä koetaan mielekkääksi käyttää.

Jotta ERP-ratkaisut ovat mahdollisimman optimaalisia, tulee niihin useimmiten lisätä lisäominaisuuksia. Nykyisin eri järjestelmien välisillä integroinneilla pystytään tuomaan järjestelmään uusia lisäominaisuuksia ja tämä parantaa huomattavasti koko organisaation suorituskykyä. Tämän kaltaiset muutokset ovat aina jatkuva prosessi, jossa tulee löytää uusia innovatiivisia tapoja virtaviivaistaa liiketoimintaa. Tämä myös estää järjestelmän vanhentumisen tulevaisuudessa, lisää käyttömahdollisuuksia sekä takaa järjestelmän joustavamman reagoinnin uusiin muutoksiin.

Pelkkä räätälöitävyys ei kuitenkaan ratkaise kaikkia ongelmia. Lisäksi nykypäivänä mobiilitoiminta on kasvanut erittäin paljon järjestelmien mukana, mikä tulee huomioida asiakkaiden tarpeissa. Odoo itsessään toimii myös mobiilisti, mutta erilliset räätälöidyt mobiilinäkymät hyödyntävät usein yrityksen prosesseja. (Bistasolutions.com 2018.)

### 3.3.8 Lokalisointi

Odoo tukee useita eri kieliä, mikä mahdollistaa myös käyttöliittymän kielen valitsemisen käyttäjäkohtaisesti. Tästä on todella paljon hyötyä monikansallisissa yrityksissä. Yhteisöstä on ladattavissa kielipaketteja ja useat kehittäjät osallistuvat yhteisön ohessa käännoistyöhön, esim. suomalaiset Odoon toimittajat ja kehittäjät osallistuvat suomen kielipaketin kääntämiseen.

Osa toimittajista pyrkii itse ylläpitämään omia kielipaketteja, pitäen sitä kilpailuetuna, vaikka lopulta tästä toimintamallista kärsivät kaikki osapuolet, niin kehittäjät kuin loppukäyttäjät. Ohjelmistoa kehitetään yleensä englanniksi, minkä jälkeen julkaistut moduulit tulee lokalisoida, kääntää halutuille kielille. (Itewiki n.d.a.)

Lokalisointi on myös muuta kuin kielikäännöksiä ohjelmistoon. CRM eli asiakkuudenhallinta toimii hyvin samalla tavalla eri maissa, jolloin kielikäännöksistä hyötyvät useat kehittäjät järjestelmää toteuttaessaan, kuin myös loppukäyttäjät tarkastellessaan sivustoa. Kirjanpidon puolen lokalisointi ei ole vain käännoistyötä, vaan osa toiminnallisuuksista pitää lokalisoida siten, että ne on käytettävissä esim. Suomessa ja huomioida mm. arvolisäverokoodit, tilikartat, tulos- ja taselaskelmat ja suomalainen pankkiviite, jotka on kehitettävä/muokattava ominaisuuksina järjestelmään, ei vain kääntää suomen kielelle. (Itewiki n.d.b.)

### 3.3.9 Käyttöoikeudet

Odoossa käyttöoikeudet ovat aina mallikohtaisia ja niillä on mahdollista määrittää ketkä käyttäjät voivat käsitellä haluttua mallia. Käyttöoikeudet löytyvät aina security-hakemiston alta. Käyttöoikeuksista vastaava malli on ir.model.access-malli, johon kaikki tarvittavat käyttöoikeudet ladataan. Kyseisessä tiedostossa kerrotaan mitä mallia käyttöoikeuksien tulee koskea ja minkälaiset oikeudet annetaan malliin sekä mitkä eri käyttäjäryhmät saavat tarvittavat oikeudet. Käyttäjille on mahdollista määrittää luonti-, haku- sekä lukuoikeus sekä myös muokkaus- ja poisto-oikeudet.

Oikeuksien määrittämissäkin löytyy eroja. Tässä tulee huomioida, onko kyseessä globaali- vai ryhmäsääntö. Globaaleissa säännöissä kaikkien on aina sovittava yhteen, jotta tietue olisi saatavilla. Luoduissa ryhmäsäännöissä taas riittää, että vain yksi asia vastaa tietuetta, niin on mahdollista päästä samaan paikkaan. (Odo 2018.)



## 4 CONEXTING-KESKUSTELUSEINÄN TOTEUTTAMINEN

Tässä luvussa perehdytään opinnäytetyön toiminnalliseen osioon, jossa käsitellään ConExting-keskusteluseinän toteuttamista. Luvussa käydään läpi projektin lähtökohta ja määrittely sekä määrittelyn mukaisen moduulin toteuttaminen ja moduulin sisältö. Kuvien avulla pyritään havainnollistamaan toteutusta Python-ohjelmointikielellä.

### 4.1 Määrittely

Niin kuin opinnäytetyön alussa todettiin, ConExting-palvelu on alun perin toteutettu 2010–2012 Yii-ohjelmistokehitysalustalla. Sen ylläpito on todettu erittäin haasteelliseksi sekä osa verkkopalvelun ominaisuuksista on havaittu tarpeettomiksi.

ConExting-keskusteluseinä on palvelu, joka on tarkoitettu käytettäväksi erilaisissa tapahtumissa, konferensseissa tai seminaareissa. Yritykset ja yhteisöt voivat ostaa itsellensä oikeuden seinään ja luoda keskustelukanavan tai keskustelukanavia, joissa pystytään kommunikoimaan muiden henkilöiden kanssa tapahtuman tiimoilta ja sen aikana. Seinälle on mahdollista liittää suorat live-lähetykset, jotta myös paikalle pääsemättömät henkilöt voivat seurata esitystä ja osallistua keskusteluun vastaavasti kuin olisivat läsnä tapahtumassa.

Keskustelukanava on maksullinen palvelu, mutta seinän toiminnallisuuksiin ja ominaisuuksiin on mahdollista tutustua ilmaiskäyttäjänä rajatun ajan. Keskusteluseinälle toteutettiin vastaavanlaiset kentät kuin nykyisessä palvelussa, jotta seinän käyttö uudella teknologialla vaikuttaa mahdollisimman tutulta palvelua aikaisemmin käyttäneille. Lisäksi aikaisemman kokemuksen mukaan nykyiset kentät ovat riittävät palvelulle.

Keskusteluseinän omistajalla on mahdollisuus muokata keskusteluseinän asetuksia niin kauan kuin seinä julkaistaan. Seinän omistaja pystyy myös hallinnoimaan seinällä tapahtuvaa viestintää, esimerkiksi huolehtia keskustelun sisällön asiallisuudesta poistamalla asiattomat viestit.

Nykyisessä ConExting-palvelussa seinän asetuksista oli mahdollista määrittää seinälle oma taustakuva ja asemoida se joko vasemmalle, keskelle tai oikealle. Tätä kyseistä ominaisuutta ei toteuteta uuteen palveluun, koska verkkosivusto tulee toimimaan responsiivisesti ja kuvan paikka määräytyy päätelaitteen skaalautuvuuden ja kuvan koon mukaan automaattisesti sivustolle. Uuteen palveluun ei tulla myöskään toteuttamaan tekstiviestien lähettämistä seinälle, koska tämä viestintätapa on jo vanhentunut.

Keskusteluseinä rakennetaan Odoon 10 version päälle, joka on toimeksiantajan tämänhetkinen tuotantoversio. Tulevaisuudessa asennus tullaan päivittämään todennäköisesti suoraan Odoon versioon 12.

## 4.2 Odoo-järjestelmän asennus

Odoo-järjestelmän testaukseen ja asennukseen on monia erilaisia mahdollisuuksia. Helppo tapa tutustua Odoo-toiminnanohjausjärjestelmään on runbot palvelun kautta, jossa on mahdollista testata pelkän selaimen kautta Odoon eri versioita. Runbotin demoasennukset sisältävät demo datan, jota voi käyttää tutustuesssa järjestelmän ominaisuuksia. Runbotin tarjoamat versiot ovat ajan tasalla ja mahdollistaa useampaan versioon tutustumisen helposti. Runbot tarjoaa myös mahdollisuuden ladata halutun asennusversion paikallisesti omalle tietokoneelle.

Odoo SAAS tarjoaa yksittäisiä esimerkkiominaisuuksia ilmaiseksi, jonka avulla kehittäjät voivat havaita ja testata Odoota ja tehdä ohjelmointia ilman paikallista asennusta, pelkällä verkkoselaimella.

Odoo tarjoaa sekä Community- että Enterprise-versioista asennusmahdollisuudet Windowsille, Deb-pohjaisille jakeluille, kuten Debian ja Ubuntu, sekä RPM pohjaisille kuten, Fedora, CentOS ja RHEL. Kyseiset asennuspaketit muodostavat automaattisesti kaikki tarvittavat riippuvuudet, joiden ajan tasalla pitäminen saattaa olla hyvin haasteellista. Viralliset yhteisön asennuspaketit, jotka mahdollistavat niin asennukset kuin päivitykset, ovat saatavilla Odoon omilta palvelimilta. Vaihtoehtoisesti sekä Community- että Enterprise asennuspaketit voidaan ladata Odoon omilta lataussivulta. Päästäkseen lataamaan Enterprise version käyttäjän on kirjauduttava sisään maksajana tai yhteistyökumppanina.

ConExting-keskusteluseinää varten asennus luotiin toimeksiantajan asennusautomaatioiden kautta. Kyseisien työkalujen avulla mahdollistettiin asennuksen tekeminen nopeasti ja helposti. Huomioitavat asiat asennuksen aikana olivat sivuston polun määrittely sekä tietokannan nimi ja tietokannan polun määrittely. Odoosta asennettiin versio 10.0, joka on LGPLv3-lisensoitu.

Ennen toteutuksen siirtoa verkkoon keskusteluseinä toteutettiin paikallisessa asennuksessa, johon vain projektin toteuttajalla oli oikeus. Tämä malli on tehokas toimintatapa asennuksen, koodin ja moduulien testaamiseen. Kun moduuli tai ominaisuus on todettu toimivaksi paikallisesti, siirrettiin moduuli kokonaisuudessaan verkossa olevaan testiasennukseen, niin että toimeksiantajan henkilökunta saattoi nähdä projektin kokonaistilanteen.

Niin kuin ConExting-keskusteluseinän tapauksessa Odoon asennus kannattaa yleisestikin suorittaa Ubuntu-käyttäjärjestelmäympäristössä, koska Odoon kehitystiimit toimivat ensisijaisesti myös Ubuntu ympäristössä, joka myös perustuu avoimeen lähdekoodiin. Jos Odoon asentaisi Microsoft ympäristössä, mukana voisi tulla ongelmia Microsoftin lisenssien kanssa. Ubuntu-käyttäjärjestelmä mahdollistaa myös paremman skaalautuvuusmahdollisuuden kuin Microsoft ympäristö.

### 4.3 Keskusteluseinä-moduulin luonti

Keskusteluseinä-moduulin luonnissa tuli tarkkaan rajata, mitä kaikkea halutaan kyseiseen moduuliin sisältävän, koska aina on mahdollista luoda rinnalle toinen moduuli, joka asennetaan samaan asennukseen ja nämä kaksi eri moduulia keskustelevat keskenään.

Keskusteluseinän moduulissa tullaan hyötykäyttämään Odoossa valmista olevaa Discuss-lisäosaa, jota on mahdollista avoimen lähdekoodin ansiosta muokata haluamalla tavalla sekä luoda vaadittuja lisäkenttiä, jotta ratkaisusta saadaan toimiva kokonaisuus. Keskusteluseinä moduuliin tuodaan mukaan myös stream eli livelähetys mahdollisuus.

Twitter toteutuksessa hyödynnetään Odoon coressa olevaa Twitter integraatio moduulia. Twitter Integraatio -moduulia tullaan muokkaamaan siten, että kyseinen moduuli hakee hashtagien perusteella twitter tviiitit. Toteutuksesta luodaan kokonaan oma moduuli, joka asennetaan järjestelmään samassa yhteydessä kuin ConExting-keskusteluseinä moduuli. Koko ConExting-keskusteluseinä tulee rakentumaan yleisistä moduulin hakemistoista.

Keskusteluseinä moduulia asentaessa järjestelmään tulee huomioida, että moduulin manifest-tiedostossa on depends kohtaan määritelty twitter integration moduulin nimi, jolloin keskusteluseinä toimii vasta silloin, kuin twitter moduulikin on asennettu järjestelmään.

Odoossa moduulin asennus tapahtuu Odoon hallinta puolella, jossa ensimmäisenä tulee tarkastaa, että admin-käyttäjälle on aktivoitu Debug -tila, jolloin käyttäjällä on mahdollista päivittää järjestelmän sovelluslista, etsiä haluttu moduuli ja asentaa se järjestelmään. Asennuksen aikana on mahdollista, että jokin virhe tapahtuu, mutta Odoon hyvä puoli on, että se ilmoittaa virheilmoituksen, josta kehittäjät ja ohjelmoijat voivat ratkaista ongelman syyn.

ConExting-keskusteluseinän tapauksessa ensimmäiset versiot ja muutokset toteutettiin projektin toteuttajan lokaalissa asennuksessa, jossa luotiin tarvittavat ominaisuudet keskusteluseinään. Suositeltava toimintatapa on se, että kun jokin asia saatiin toimimaan paikallisesti, tuotos siirrettiin kokonaisuudessaan Gitlabiin, jossa pystytään hallinnoimaan projektin versiohallintaa, mistä mainittu selkeämmin osiossa Git-versionhallinta. Kun projektin toteuttaja totesi ominaisuuden toimivan paikallisesti, ominaisuus siirrettiin verkkoasennukseen ja toimeksiantajan ohjelmistoasiantuntijoille nähtäväksi.

Ominaisuuksien lisääminen verkossa olevaan testiasennukseen tapahtuu erittäin helposti ja vaivattomasti. Toteuttajan tuli kirjautua asennuspalvelimelle, josta otettiin etäyhteys Gitlabissa löytyvään keskusteluseinä mo-

duuliin ja ladattiin projekti palvelimelta löytyvän custom-addons hakemiston alle. Tämän jälkeen on tärkeää, että asennus pysäytetään ja käynnistetään uudelleen, jonka jälkeen kirjaudutaan julkiseen asennukseen, jossa moduulin asennus tapahtuu täysin samalla tavalla kuin paikallisessa asennuksessa. Asentaessa saattaa tapahtua yleensä virheitä, kuten tässäkin tapauksessa projektin toteuttajalla oli koodissa tietyt linkitykset sekä rakenteet paikallisesti eri tavalla kuin olisi pitänyt olla julkisessa asennuksessa.

### 4.3.1 Moduulin rakenne

Moduulin rakenne koostuu aina erilaisista hakemistoista, jotka sisältävät liiketoimintalogiikan, ja tarkastelemalla niitä on mahdollista ymmärtää eri moduulien tarkoitukset (Kuva 5). Moduuli sisältää hakemistot `data/`, `models/`, `controllers/`, `views/` sekä `static/`. Moduulin tiedostojen nimeäminen auttaa ulkopuolisia henkilöitä hahmottamaan paremmin moduulin tiedostojen ja rakenteen toiminnallisuutta.

## Module structure

### Directories

A module is organised in important directories. Those contain the business logic; having a look at them should make understand the purpose of the module.

- `data/`: demo and data xml
- `models/`: models definition
- `controllers/`: contains controllers (HTTP routes).
- `views/`: contains the views and templates
- `static/`: contains the web assets, separated into `css/`, `js/`, `img/`, `lib/`, ...

Other optional directories compose the module.

- `wizard/`: regroupes the transient models (formerly `osv_memory`) and their views.
- `report/`: contains the reports (RML report [**deprecated**], models based on SQL views (for reporting) and other complex reports). Python objects and XML views are included in this directory.
- `tests/`: contains the Python/YML tests

Kuva 5. Odoo moduulin hakemiston rakenne ja tiivistetyt selitteet.

Keskusteluseinä-moduulin datakansio sisältää kaikki eri oikeudet, mitä moduuliin toimintaan vaaditaan. Kansion sisällä olevat tiedostot ovat aina xml muodossa ja tiedostossa on määritelty vaaditut ryhmät, jotka antavat tietyt oikeudet joko kirjautuneille käyttäjille tai ei-kirjautuneille sekä seinän omistajalle. Samalla luodaan navigaatioon pyydytyt valikot ja samoin niille määritetään tarkasti oikeudet, mitä näkyy kenellekin.

Models-kansiossa löytyy `.py`-päätteisiä tiedostoja eli python ohjelmointikielellä toteutettuja tiedostoja. Controllers-kansio sisältää myös aina `.py`-päätteisiä tiedostoja, mutta suositeltava tapa on pitää aina models-kansiossa tiedostot, joissa luodaan tai otetaan yhteys backendissä olevaan moduuliin ja täten voidaan luoda haluttuja kenttiä tai muita vaadittavia ominaisuuksia, joita moduuliin halutaan lisätä. ConExtng-keskusteluseinässä

models-hakemiston tiedostoissa otetaan yhteys Discuss-lisäosaan, johon luodaan kenttiä, jotta kanavien asetukset ovat määrittelyn mukaiset.

Controllers-kansio sisältää aina ohjaimia (HTTP routes), joilla ohjataan sivustolle liikkumista. Samoin jokaisessa omassa http.route osiossa määritetään, onko kyseessä ohjaaminen verkkosivustolle tai onko sivusto näkyvissä vain kirjautuneille käyttäjille vai kaikille. Samoin on mahdollista määrittää kaikki tiedot, mitkä halutaan näkyvän kyseisellä sivustolla. ConExting-keskusteluseinässä haetaan todella paljon yhteyttä Discuss-moduuliin, jotta saadaan kanavat tuotua frontendin puolelle sekä niiden tiedot. Controllers-osiossa on hyvä luoda monia eri tiedostoja ja nimetä ne selkeästi, jotka kuvaavat haluttua toimintoa, jolla pystytään tekemään koodista huomattavasti luettavampaa ja muille koodin tarkastelijoille selkeämpi.

Views-hakemisto sisältää kaikki xml tiedostot, joissa rakennetaan näkymiä sekä verkkosivustoa. Tämänkin kansion tiedostoissa pätee samanlainen nimeämistyyli kuin muissakin. Näissä tiedostoissa on erittäin tärkeää, kuten muulloinkin Odoon ohjelmoimissa, että itse koodi tulee olla oikein sisennetty ja kaikki tarvittavat sulkuumerkit pitää löytyä, mitä esimerkiksi Joomla!:n php-koodi ei vaadi huomioimaan yhtä tarkasti.

Keskusteluseinää varten luotiin kanavanäkymät sekä kaikki muut sivustot, mitä valikko rakenteesta löytyi. Odoossa on hyvä käyttää jotakin bootstrap tyyliä, kun luodaan verkkosivustoa, koska Odoon tukee bootstrapia ja näin sivusto tulee toimimaan oletuksena responsiivisesti. Vaikka myöhemmin lisättäisiin ylimääräisiä tyyli-tiedostoja, jota bootstrap ei tunnista omakseen, responsiivisuus toimii oletuksena oikein.

Static on tarkoitettu verkkosivuston assetseihin, josta löytyy kaikki css-, less-, j-s, img- ja lib-tiedostot ja materiaalit. Nämä mahdollistavat sivuston ulkoasun sekä pienten lisäominaisuuksien teon. Normaalisti verkkosivujen ulkoasua muokatessa käytetään CSS-tyylitiedosto menetelmää, mutta Odoon:ssa kannattaa käyttää LESS:iä, joka on hyvin vastaavanlainen kuin CSS, mutta LESS mahdollistaa erilaisten attribuuttien luomisen ja määrittämisen, jolla pystytään yhdellä muutoksella vaihtamaan esimerkiksi mo-  
neen kohtaan sama väri.

#### 4.3.2 Käyttäjänäkymät

Palvelu tulee sisältämään kolme käyttäjäryhmää: kaksi maksullista käyttäjäryhmää sekä ilmaiskäyttäjät. Jokaiselle käyttäjäryhmälle luotiin tietyt oikeudet, joilla määritetään heidän liikkuminen sivustolla sekä asiat, joita käyttäjät voivat tehdä keskusteluseinällä. Seinän omistaja voi kuulua mihin tahansa näistä kolmesta ryhmästä, mutta se mihin ryhmään käyttäjä kuuluu, tuo henkilölle tietyt rajoitteet tai oikeudet. Rajoitteita tulee olemaan

seinän kestossa, seinän muokkauksessa sekä erilaisten lisäominaisuuksien kanssa.

Keskusteluseinässä tulee olemaan myös ei-kirjautuneita käyttäjiä, jotka tulevat näkemään vain infoa keskusteluseinästä ja mitä se sisältää. Ei-kirjautuneille käyttäjille määritetään rajatut oikeudet, jotta he eivät näe niitä asioita, jotka kuuluvat vain kirjautuneille käyttäjille. Käyttäjä ryhmiä luodessa projektin toteuttajan tuli pohtia tarkkaan montako ryhmää ja mitä eroavaisuuksia eri ryhmissä oikeuksien puolesta on. Ryhmiä voidaan luoda Odoon hallinnanpuolelta graafisesti, mutta tässä tapauksessa, kun tiedettiin tarkkaan montako ryhmää ja millaiset oikeudet, oli mahdollista luoda ryhmät moduulin security-hakemiston sisällä. Tässäkin tapauksessa kannattaa eritellä selkeästi tiedosto, missä ryhmät luodaan sekä tiedosto, missä määritetään oikeudet sivustolla liikkumista varten sekä toiminnallisuksien käyttämisestä.

Tiedostot ovat xml-päätteisiä tiedostoja, joissa tulee selkeästi määritellä ryhmien nimet sekä mihin asiaan oikeudet tai ryhmät kuuluvat. Alla olevassa kuvassa luodaan premium-niminen ryhmä, joka viedään Odoon backendin puolella olevaan group -osioon ja niille käyttäjille, jotka tulevat kyseiseen ryhmään, saavat tietyt oikeudet.

Samoin esimerkkikuvassa 6 määritetään verkkosivuston navigaation oikeuksia, jossa kerrotaan mihin sivuilla ei-kirjautuneelta piilotetaan kokonaan näkyvästä.

```
<!--Premium group-->
<record id="premium" model="res.groups">
  <field name="name">Premium</field>
  <field name="groups" eval="[[4, ref('base.group_user')]]"/>
  <field name="perm_read" eval="True"/>
  <field name="perm_write" eval="False"/>
  <field name="perm_create" eval="False"/>
  <field name="perm_unlink" eval="False"/>
</record>
<!-- Not logged user navigation menu-->
<record id="conexting_not_logged" model="ir.rule">
  <field name="name">Not logged user navigation</field>
  <field name="model_id" ref="website.model_website_menu"/>
  <field name="domain_force">[('url', 'not like', '/page/account'), ('url', 'not like', '/page/channel')]</field>
  <field name="groups" eval="[[4, ref('base.group_public')]]"/>
  <field name="perm_read" eval="True"/>
  <field name="perm_write" eval="False"/>
  <field name="perm_create" eval="False"/>
  <field name="perm_unlink" eval="False"/>
</record>
```

Kuva 6. Ryhmän luominen sekä käyttäjien oikeuksien määrittelyä.

#### 4.3.3 Discuss-moduulin hyödyntäminen

Odoon Discuss-lisäosa moduuli on erittäin yksinkertainen ja helppo sovellus, joka on tarkoitettu viestittelyyn. Discuss-moduulilla pystytään tuomaan kaikki yrityksen sisällä tapahtuva viestittely yhteen paikkaan ja lisäosa on myös mahdollista integroida Odoon foorumiin. Discuss-moduulissa on mahdollista lähettää sekä vastaanottaa viestejä missä tahansa Odoossa, sekä mahdollisuus hallinnoida tulevia viestejä ja ilmoituksia sovelluksesta käsin. Discuss-moduulin avulla on myös mahdollista luoda erilaisia kanavia

eri keskustelu tarkoituksiin. Kanavia on mahdollista käyttää yksittäisten ryhmien tai osastojen tai projektiryhmien kanssa, jossa on edellytyksenä säännöllinen viestintä. Discuss-moduuli tarjoaa kahdenlaista kanava mahdollisuutta. Julkinen kanava on tarkoitettu parhaiten siihen, kun useampi työntekijä tarvitsee informaatiota tai tietoa. Yksityinen kanava on hyvä silloin, kun tieto on arkaluonteista tai tarkoituksena on vain keskustella tiettyjen henkilöiden kanssa.

Discuss-moduuli on alun perin suunniteltu ainoastaan Odoon backend-puolelle. Kuitenkin tätä moduulia pystytään hyötykäyttämään ConeXting-keskusteluseinällä, tuomalla kanavilta tarvittavat tiedot frontend-puolelle. Discuss-kanavalta tuodaan frontendin puolelle kanavien tiedot, jotka toimivat projektissa seinän tietoina. Samoin frontend-puolelle tuodaan kaikki kanavan keskustelut sekä mahdollisuus keskustella frontendin puolella. Jotta kaikki tarvittavat tiedot on mahdollista nähdä, on luotava uusia lisäkenttiä valmiina olevaan moduuliin.

Uusien kenttien luonti tapahtuu Odoossa model-hakemiston alta, jossa määritetään minne Odoossa otetaan yhteys ja mitä sinne tulee luoda. Esimerkiksi kyseisessä tapauksessa luotiin mail\_channel.py tiedosto, jonne määritetään kentän nimi sekä tyyppi.

Opinnäytetyön aikana toteutetussa pilottiversiossa ei tulla toteuttamaan kanavan ostoprosessia frontendin puolelta, koska toimeksiantajan muissa verkkopalveluissa on vastaava toiminnallisuus jo toteutettuna, jota voidaan lähes suoraan hyötykäyttää ConXtingissa. Maksutapoina ovat Paytrailin maksuominaisuudet ja maksusuorituksen jälkeen seinän omistajalle luodaan automaattisesti vakioseinä asetuksineen, jota ostaja pääsee muokkaamaan haluamallaan tavalla. Kuvassa 7 kuvataan ohjelmointi koodi tasolla kanavien tuontia frontend-näkymään.

```
#haetaan aktiivisen käyttäjän kaikki kanavat näkyviin, jotka hän omistaa
#tai on lisätty ryhmään
@http.route('/page/account', type='http', auth='user', website=True))
def my_channels(self, page=1, date_begin=None, date_end=None, **post):
    current user = http.request.env.user
    channels = http.request.env['mail.channel'].search([])

    pager = request.website.pager(
        url='/page/account',
        page=page,
        url_args={'date begin': date begin, 'date end': date end},
        total = len(channels),
    )
    values = dict()
    values.update({
        'channels': channels,
        # 'users': users,
        'pager': pager,
        'date begin': date begin,
        'date end': date end,
        #'channel': channel,
    })
    return request.render("conexting.my_channels", values))
```

Kuva 7. Discuss-lisäosan kanavien tuonti frontend-puolelle.



#### 4.3.4 Website twitter moduulin hyödyntäminen

Vanhassa palvelussa oli mahdollistettu twitter hashtagien hakeminen seinien keskusteluosioon, jolloin asetuksissa oli määritelty haluttu twitter hashtag, jonka avulla hashtagilla merkityt viestit tuotiin seinien keskusteluosioon.

Uudessa ConExting-keskusteluseinässä tullaan hyödyntämään Odoon coresta löytyvää Twitter integraatio moduulia, joka on toteutettu ainoastaan Odoon backend-puolelle. Odoon backend-puolella määritetään twitter sovelluksen avaimet hallintaan sekä käyttäjätunnus, josta viestit haetaan. Moduulin tarkoituksena on hakea ainoastaan viestit, jotka on twitterin puolelta merkitty suosikeiksi eli viestit, joita ei merkitä suosikkeihin, eivät tule näkymään ollenkaan Odoossa. Toteutusta alettiin soveltamaan siten, että moduulin toteutuksesta luodaan oma lisäosa, joka hakee viestit twitter hashtagien perusteella käyttäen twitter API:sta löytyvää "Search Tweets" -osio.

Ominaisuuden toteutus käynnistettiin testaamalla Twitter Intergration moduulilla Odoon ja twitterin välinen toimivuus, jolloin saatiin varmistus, että ominaisuus toimii moitteettomasti. Hyödyntämällä coresta löytyvää moduulia nopeutetaan toteutusta, koska alustavasti kaikki ominaisuudet sekä tarvittavat kentät ja erilaiset "ohjelmointi koodit" olivat jo valmiiksi tukemassa haluttua toteutusta.

Testauksen yhteydessä havaittiin, että yritysryhmästä löytyy moduuli, joka hakee viestit hashtagien mukaan Odoon backend-puolelle, mutta se oli toteutettu Odoon versio 9 päälle, joten moduuli tuli päivittää Odoon versio 10 toimivaksi.

Jotta lisäominaisuutta oli mahdollista kehittää vastaamaan ConExtingin tarpeita, tuli perehtyä tarkasti versioiden välisiin koodimuutoksiin. Päivitetyn moduulin toimivuus tuli testata sekä huomioida, ettei mitään discuss-moduulin tai twitterin vaatimista kentistä puuttunut.

Varmistuksien jälkeen aloitettiin kehittämään moduulin toiminnallisuutta siten, että viestit saadaan haettua Odoon backend-puolelta sivuston frontend-puolelle (Kuva 8). Tässä kohtaa tuli olla selkeä käsitys moduulin attribuuteista, jotta yhteys on mahdollista luoda ja kutsua vaadittuja kenttiä sekä tietoja frontend-puolelle (Kuva 9).

Twitter moduulia voidaan käyttää myös itsenäisesti ilman ConExting-keskusteluseinä moduulia.



```

#Otetaan yhteys backend puolella olemaan Twitter Integration moduuliin
#Haetaan kaikki twitter tweetit tietyn hashtaigin mukaan
TwitterTweets = request.env['website.twitter.tweet'])

tweets = TwitterTweets.search(
  [('website id', '=', request.website.id),
   ('twitter hashtag', '=', twitter_hashtag),],
  limit=int(limit), order="tweet id desc")

tweets dict = dict()
for tweet in tweets:

  tweets dict[tweet.id] = json.loads(tweet.tweet))
print tweets dict[tweet.id]['text'])

if post:
  message = post['comment'])
  channel.message_post(
    subtype='mt comment',
    body=message,
  ))

```

Kuva 8. Twitter Integration moduulin tviittien tuominen frontendille.

```

<div id="discussion_messages">
  <!--Otetaan yhteys discuss lisäosan kanaviin sekä twitter tweet kenttiin-->
  <!--Backendin puolelta tuodaan frontendin puolelle viestit, viestin kirjoittaja sekä aika-->
  <t foreach="channel.channel_message_ids" t-as="message">
    <t foreach='tweets' t-as='tweet'>
      <div class="media" id="discussion">
        <div class="media-left">
          <span t-field="message.write_uid"/><span t-esc="message.write_date"/>
          <span t-esc="tweet['user']['name']" />
          <span t-esc="tweet['user']['created_at']" />
        </div>
      </div>
      <div class="media-body">
        <span t-field="message.body"/>
        <span t-esc="tweet['text']" />
      </div>
    </t>
  </t>
</div>

```

Kuva 9. Viestien ja tarvittavien tietojen saaminen näkymään frontendin templatella.

#### 4.3.5 Tuetut stream palvelut

ConExting-keskusteluseinän tulee mahdollistaa myös erilaiset stream eli livelähetys palvelut. Stream on palvelu sekä tekniikka, jota on tarkoitus käyttää sisällön toistamiseen tietokoneella sekä mobiililaitteilla internetin välityksellä. Stream lähettää jatkuvasti erilaista tietoa kuten ääntä ja videota jatkuvana virtauksena, jotta vastaanottajat voivat katsella tai kuunnella live lähetyksiä reaaliajassa.

Vanhassa ConExting-palvelussa oli mahdollistettuna kolme erilaista stream palvelua, joissa tuli tietää aina live lähetyksen tunnus. Uuteen palveluun toteutusta sovellettiin ja tehtiin käyttäjäystävällisempi ratkaisu.

Tulevassa ConExting-keskusteluseinä palveluun mahdollistetaan yhteensä 7 eri Stream palvelua ja täten ConExting-keskusteluseinä ei ole niin riippuvainen vain tietyistä stream palveluista.

Stream toteutettiin toimimaan iframe upotuksella, joka tarkoittaa sitä, että tuodaan toinen html -elementti sivuston sisälle ja täten seinän omistajan on mahdollista käyttää monia eri stream palveluja, koska suurimmassa osassa live lähetyksiä upotus toimii samalla tavalla. Jotta kyseinen stream ominaisuus oli mahdollista toteuttaa tuli Discuss-moduuliin luoda tyhjä kenttä, johon seinän omistaja voi upottaa ohjeiden mukaan live lähetyksensä URL:in. Kentän ympärille tuli muuten määrittää kaikki tarpeellinen iframe koodista (Kuva 10).

```
<!--Tähän tuodaan livelähetykset!-->
<div class="embed-container">
  <iframe id="video0" title="Video Player" width="400" height="385" t-att-src="
  channel.iframe_stream" frameborder="0"></iframe>
</div>
```

Kuva 10. Iframe channel\_template.xml pohjassa.

Tässä yhteydessä opinnäytetyön tekijä tutustui myös erilaisiin stream palveluntarjoajiin ja miten näissä palveluissa livelähetyksien upotus toimii.

#### 4.3.6 ConExting-keskusteluseinän ulkoasu

Verkkosivuston teemauksen liittyy yleensä tyyliasun tekeminen ja tarvittavat muutokset sekä itse sivu pohjien tekeminen. Joissakin palveluissa on mahdollista asentaa valmiita teemoja, jolloin henkilö, keneltä ei löydy osaamista itse toteuttaa ulkoasua voi hyötykäyttää jonkun toisen toteuttamia ulkoasuja. Oletusteemoja käyttäessä kannattaa testata sivuston toimivuus eri päätelaitteilla, koska teeman toiminnallinen vastuu on toteuttajalla.

Odoossa sivuston rakenne ja sivut toteutetaan xml tiedostoilla, jotka määritetään Views-hakemiston sisälle, josta on mainittu osiossa moduulin rakenne. Erittäin hyvä tapa on käyttää jokaisessa sivussa mahdollisimman samanlaista rakennetta, jotta tulevien virheiden tai ongelmien paikallistaminen sekä korjaaminen on helpompaa ja samoin sivuston tyyliasun muokaus pysyy yksinkertaisena. Odoossa sivuston rakenne kannattaa rakentaa siten, että se hyötykäyttää ja toimii Bootsrapilla, jolla rakennetaan sivustosta responsiivinen eli sivusto tulee toimimaan kaikilla eri päätelaitteilla, millä on vainkin mahdollista päästä internettiin.

Responsiivisuudella tarkoitetaan sivuston rakennetta, jossa sivusto tunnistaa käyttäjien käyttämät laitteet ja täten sivuston rakenne muokkautuu sisältöjen, ulkoasun sekä erilaisten toiminnallisuuksien mukaan käytettäväksi eri päätelaitteilla. Tämä on nykypäivää, jolla on korvattu vanha toimintatapa, jolloin luotiin erikseen mobiilisivusto sekä tietokoneella toimiva sivusto.

Sivuston erilaiset tyylimuutokset toteutetaan yleensä CSS tekniikalla, joka on tyyliohjaiden laji. Tällä pystytään määrittelemään HTML-kielillä toteutetulle sivustolle ulkoasu ja muuttaa esimerkiksi tekstin kokoa tai väriä

(Kuva 11). CSS koostuu aina valitsimesta, attribuutista sekä attribuutin arvosta.

```
html{
  font-family: 'Source Sans Pro', sans-serif;
}

.navbar-default{
  color: #fff;
  background: rgba(24, 23, 22, 0.7);
}
```

Kuva 11. Esimerkki koodi normaalista CSS tyylitiedostosta.

ConExting-keskusteluseinän verkkosivuston tyyliasumuutoksia ei kuitenkaan toteutettu CSS:llä vaan käytettiin LESS:iä, eli ”Leaner Style Sheets”, joka on yhteensopiva laajennus CSS:lle. LESS tyylitiedosto merkintöjen tekeminen muistuttaa erittäin vahvasti normaalia CSS-koodia, mutta LESS:ssä pystytään luomaan valmiita attribuutteja esimerkiksi väreille ja sitten tiedoston oikeissa kohdissa voidaan kirjoittaa vain attribuutin nimi, joka perii kyseisen määrittelyn. Projektin toteuttajalla on tässä tapauksessa vastuu, miten käyttää attribuutteja ja miten eri attribuutit nimetään ja määritetään LESS-tiedostoon, jotta toimeksiantajan muutkin ohjelmistoasiantuntijat ymmärtävät ja lukevat tyylitiedostoa mahdollisimman sujuvasti (Kuva 12).

```
@nice-blue: #5883AD;
@light-blue: @nice-blue + #111;

#header {
  color: @light-blue;
}
```

Kuva 12. Esimerkki koodi LESS tyylitiedostosta.

## 5 KÄYTTÖÖNOTTO

Opinnäytetyön tuloksena toteutetaan pilottiversio ConExting-keskustelu-seinästä, joka sisältää palvelun päätoiminnot: seinän luonti ja asetukset, käyttäjien kutsuminen palveluun, keskustelunäkymän luonti, määritelyjen twitter viestien haku osaksi keskustelua, livelähetyksien liittäminen seinään sekä käyttäjäryhmät. Verkkopalvelun ulkoasua ei ole vielä täysin määritely, mutta sivuston ulkoasu tulee pohjautumaan palvelun markkinointisivustoon.

Opinnäytetyössä ei toteuteta seinän ostoprosessia, koska toimeksiantajan muissa verkkopalveluissa on vastaava toiminnallisuus jo toteutettuna, jota voidaan lähes suoraan hyötykäyttää ConExtingissa.

Ostoprosessin lisäämisen yhteydessä toimeksiantajan ohjelmistoasiantuntijat tarkistavat kokonaisuuden, jotta toteutus vastaa yritysryhmän vaatimuksia ja käytänteitä.

## 6 YHTEENVETO

Tässä luvussa käydään läpi opinnäytetyön tulokset sekä saavutetut tavoitteet. Lisäksi tullaan käymään läpi opinnäytetyön aikana kohdattuja haasteita sekä ongelmia ja jatkokehitysmahdollisuuksia.

### 6.1 Tulokset

Tulokset olivat hyviä siihen nähden, että projektin toteuttajalla ei ollut aikaisempaa kokemusta Odoon ohjelmoinnista tai ylipäätään python ohjelmointikielestä. Opinnäytetyön tuloksena syntyi pilottiversio, jonka kehitystä jatketaan loppuun toimeksiantajan muiden ohjelmistoasiantuntijoiden tuella. Kaikki mitä alkuperäisessä määrittelyssä oli rajattu toteutettavaksi, onnistuttiin toteuttamaan ConExting-keskusteluseinään.

Uusittu ConExting-keskusteluseinä on tarkoitus julkaista kesällä 2018. Opinnäytetyötä kirjoittaessa tarkempi ajankohta on vielä sopimatta. Keskusteluseinä moduulia on mahdollista hyödyntää jatkossa myös muissa projekteissa sellaisenaan tai yksittäisenä ominaisuutena, jolloin toimeksiantajan ei tarvitse tehdä vastaavaa toteutusta alusta alkaen kokonaan.

### 6.2 Haasteet

Kuten edellisessä kappaleessa todettiin, projekti oli todella haastava, koska projektin toteuttajalla ei ollut ennen projektin alkua aikaisempaa kokemusta Odoosta tai sen ohjelmoinnista tai kehittämisestä. Ohjelmointi ja kehittäminen vaativat syvällistä osaamista itse ohjelmasta ja ohjelmointikielestä, jolloin monesti myös ohjeistuksen ymmärtäminen oli haastavaa ja vaati syvällistä tutustumista aiheeseen. Python ohjelmointikieli osoittautui myös hyvin määrämuotoiseksi, joka vaati erityistä tarkkuutta ja huolellisuutta.

Oma haasteensa projektissa oli tuoda toiminallisuus tai ominaisuus järjestelmän backend-puolelta frontend-puolelle samoin kuin Twitter ominaisuuden tuominen palveluun, jossa joutui tutustumaan syvällisesti myös itse Twitterin APIiin. Twitter toteutukseen liittyi javascript-, python- sekä html -kielet, jotka kaikki piti saada toimimaan yhdessä.

Projektin aikana Python-ohjelmointikieli kehittyi kuitenkin huomattavasti toteutettaessa eri toiminnallisuuksia ja niiden ominaisuuksia. Samoin projektin toteuttajan käsitys Odoon ohjelmoinnista kehittyi huomattavasti, ja kyky oppia hyödyntämään valmiita ratkaisuja tai ominaisuuksia sekä Odoon hallintapuolen developer modea omassa työssään.

### 6.3 Jatkokehitysmahdollisuudet

Avoimen lähdekoodin ansiosta palvelua on mahdollista jatkokehittää tulevien tarpeiden mukaisesti. Palvelu on helposti monistettavissa eri asiakkaiden tarpeisiin. Palvelu mahdollistaa myös integraatiot muiden järjestelmien kanssa tarpeen mukaan.

Jatkokehitysideoina tulisi huomioida Twitterin lisäksi myös muita some-kanavia, esimerkiksi Instagramin mukaan tuominen palveluun tai Whatapp-tyyppisten viestimien mahdollistaminen. Samoin palvelua voisi jatkokehittää mahdollistamalla kirjautuminen tai tunnistautuminen google- tai facebook tunnuksilla.

## 7 LÄHTEET

Accountorenterprise (n.d.). Mikä on ERP-järjestelmä. Haettu 16.1.2018 osoitteesta <https://www.accountorenterprise.fi/2017/08/08/mika-erp-jarjestelma/>

Bistasolutions (2017). Why should your business go for odoo customization. Haettu 10.2.2018 osoitteesta <https://www.bistasolutions.com/resources/blogs/why-should-your-business-go-for-odoo-customization/>

Coss (n.d.a.). Avoimen lähdekoodin lisenssit. Haettu 19.2.2018 osoitteesta <https://coss.fi/avoimuus/lisenssit/>

Coss (n.d.b.). Avoin lähdekoodi. Haettu 20.1.2018 osoitteesta <https://coss.fi/avoimuus/avoin-lahdekoodi/>

Git-scm.org (n.d.). Alkusanat versionhallinnasta. Haettu 22.1.2018 osoitteesta <https://git-scm.com/book/fi/v1/Alkusanat-Versionhallinnasta>

IteWiki (n.d.a.). Odoo. Haettu 2.2.2018 osoitteesta <https://www.itewiki.fi/opas/odoo/>

IteWiki (n.d.b.). Odoon lokalisointi Suomeen. Haettu 18.2.2018 osoitteesta <https://www.itewiki.fi/opas/odoo/#Odoon-lokalisointi-Suomeen>

Linux (n.d.). GNU LGPL. Haettu 19.2.2018 osoitteesta [https://www.linux.fi/wiki/GNU\\_LGPL](https://www.linux.fi/wiki/GNU_LGPL)

Odoo development cookbook. (2018) Build effective application by applying Odoo development best practices. Haettu 22.1.2018

Odoo development cookbook. (2018) Build effective applications by applying Odoo development best practices. Haettu 16.1.2018

Odoo (n.d.). About us. Haettu 2.2.2018 osoitteesta <https://www.odoo.com/page/about-us>

Odoo (n.d.). Odoo apps, modules and licensing FAQ. Haettu 5.2.2018 osoitteesta <https://www.odoo.com/forum/help-1/question/odoo-apps-modules-licensing-faq-55449>

Odoo (n.d.). Security in Odoo. Haettu 18.2.2018 osoitteesta <https://www.odoo.com/documentation/10.0/reference/security.html>

Odoo (n.d.). The Odoo story. Haettu 19.1.2018 osoitteesta <https://www.odoo.com/blog/odoo-news-5/post/the-odoo-story-56>

Omnipartners (n.d.). CMS eli Content Management System. Haettu 1.3.2018 osoitteesta <https://omnipartners.fi/sanakirja/cms-eli-content-management-system/>

Webopas (n.d.). Mikä on CMS. Haettu 1.3.2018 osoitteesta <http://www.webopas.net/cms.html>

Wiki.archlinux.org (n.d.). Odoo. Haettu 18.1.2018 osoitteesta <https://wiki.archlinux.org/index.php/Odoo>