

**Magento-järjestelmän  
SugarCRM-moduulin  
vaatimusmäärittely**  
Case: Trainer4You Oy

Mika Majala

Opinnäytetyö  
Toukokuu 2017  
Luonnontieteiden ala  
Tradenomi (AMK), Tietojenkäsittelyn tutkinto-ohjelma

Tekijä(t) Majala, Mika	Julkaisun laji Opinnäytetyö, AMK	Päivämäärä Toukokuu 2017
	Sivumäärä 48	Julkaisun kieli Suomi
		Verkojulkaisulupa myönnetty: x
Työn nimi <b>Magento-järjestelmän SugarCRM-moduulin vaatimusmäärittely</b> Case: Trainer4You Oy		
Tutkinto-ohjelma Tietojenkäsittelyn tutkinto-ohjelma		
Työn ohjaaja(t) Tuikka, Tommi		
Toimeksiantaja(t) Trainer4You Oy		
Tiivistelmä <p>Liikunta- ja hyvinvointialan palvelut ja koulutukset ovat olleet suuressa suosiossa viime vuosien aikana. Opinnäytetyön toimeksiantaja on Suomessa toimiva Trainer4You Oy, joka kouluttaa uusia valmentajia liikunta- ja hyvinvointialalle. Yritys järjestää mm. personal trainer-, hyvinvointi-, kuntosali- ja ravintovalmentajakoulutuksia.</p> <p>Toimeksiantajalle on luotu vahvat prosessit myynnin ja markkinoinnin parantamiseksi, mutta niitä varten verkkokauppaympäristöön kehitetyt analytiikan ohjelmistoratkaisut eivät ole enää täysin riittäviä, ne ovat hajautuneet useaan osaan ja niiden ylläpito on käynyt hankalaksi ja työlääksi. Opinnäytetyön tavoitteena oli selvittää ja dokumentoida kaikki mahdolliset vaatimukset yhtenäistetyille moduulille, joka kattaisi kaikkien aiemmin kehitettyjen analytiikan ratkaisujen ominaisuudet, mutta olisi säädettävissä verkkokaupan hallintapaneelin käyttöliittymän kautta ja olisi helpommin laajennettavissa jatkoa ajatellen.</p> <p>Opinnäytetyö toteutettiin olemassa olevien ohjelmistoratkaisujen kehittämistutkimuksena. Tutkimuksen aikana perehdyttiin verkkokauppaympäristön tekniseen teoriaan ja etsittiin sen avulla keinoja parantaa aiemmin kehitettyjä ohjelmistoratkaisuja. Samalla otettiin huomioon tiedossa olevat myynnin ja markkinoinnin prosessit ja mahdolliset toiveet niiden parantamiseksi. Teoriaosuudessa käsitellään käytössä olevia järjestelmiä ja niissä käytettäviä tekniikoita, asiakkuudenhallintaa ja vaatimusmäärittelyn prosessia.</p> <p>Tutkimuksen lopputuloksena syntyi vaatimusmäärittelydokumentti, jonka pohjalta moduuli myöhemmin kehitetään. Dokumentin hyväksytty versio on työssä liitteenä, mutta mahdolliset muutokset ja parannukset ovat mahdollisia kehitystyön aikana.</p>		
Avainsanat ( <a href="#">asiasanat</a> )  vaatimusmäärittely, Magento, SugarCRM, asiakkuudenhallinta		
Muut tiedot		

Author(s) Majala, Mika	Type of publication Bachelor's thesis	Date May 2017 Language of publication: Finnish
	Number of pages 48	Permission for web publication: x
Title of publication <b>SugarCRM module requirements specification for Magento system</b> Case: Trainer4You Oy		
Degree programme Business Information Systems		
Supervisor(s) Tuikka, Tommi		
Assigned by Trainer4You Oy		
Abstract  <p>Sports and wellness services have been on high demand in the recent years. Trainer4You Ltd, the assigner of this thesis, is a Finnish company training new trainers for those services. The company provides courses specializing in personal training, wellness training, gym training and nutrition training.</p> <p>The company has developed strong processes to help with sales and marketing, however, the analytics solutions developed for their e-commerce platform are no longer up to the task, since they consist of multiple different parts and are difficult and time consuming to manage. The objective of the thesis was to analyze and document every possible requirement for a unified module that would have all the features found in the current solutions, would be configurable through the e-commerce control panel and could be easily expanded later.</p> <p>The thesis was carried out as a development research. The technical theory of the e-commerce platform was studied during the research to find better ways to develop the current solutions. The processes in sales and marketing as well as possible enhancements in both areas were kept in mind. The theoretical part of the research covers the existing software platforms and their technologies, customer relationship management and the requirements specification processes.</p> <p>The outcome of the research was a requirements specification document for the module that would be developed later. The accepted version of the document is attached to the report, however, changes and enhancements to it during the development process are possible.</p>		
Keywords/tags ( <a href="#">subjects</a> )  requirements specification, Magento, SugarCRM, customer relationship management		
Miscellaneous		

## Sisältö

Käsitteet .....	3
1 Johdanto.....	4
2 Tutkimusasetelma .....	4
2.1 Toimeksiantaja .....	4
2.2 Opinnäytetyön tausta, rajaukset ja tavoitteet.....	5
2.3 Tutkimuskysymykset ja menetelmä .....	6
3 Järjestelmät .....	6
3.1 Magento .....	7
3.1.1 Historiaa.....	7
3.1.2 Perusteet .....	8
3.1.3 Modulaarisuus ja moduulirakenne.....	10
3.1.4 Moduulien konfigurointi.....	11
3.1.5 Luokkien laajentaminen, yli- ja uudelleenkirjoittaminen.....	13
3.1.6 Luokkien käyttäminen .....	14
3.2 SugarCRM .....	16
3.2.1 Asiakkuudenhallinta .....	16
3.2.2 On-Demand .....	17
3.2.3 Teknologiat ja REST -rajapinta .....	17
4 Vaatimusmäärittely .....	18
4.1 Vaatimusmäärittelyn osapuolet.....	19
4.2 Vaatimukset.....	20
4.2.1 Toiminnalliset vaatimukset.....	20
4.2.2 Ei-toiminnalliset vaatimukset .....	20
4.2.3 Rajoitteet .....	21
4.3 Vaatimusten selvittäminen .....	21
4.4 Dokumentointi .....	22

	2
5 Tutkimuksen toteutus .....	23
6 Tutkimuksen tulokset.....	24
6.1 Yleiskuvaus .....	24
6.2 Toiminnalliset vaatimukset .....	25
6.3 Ei-toiminnalliset vaatimukset.....	26
6.4 Muut vaatimukset .....	27
6.5 Rajoitukset.....	28
6.6 Tiedot ja tietokannat .....	28
7 Pohdinta .....	28
Lähteet.....	30
Liitteet .....	32
Liite 1. Moduulin vaatimusmäärittelydokumentti .....	32

## **Kuviot**

Kuvio 1. Magenton MVC-rakenne.....	9
Kuvio 2. Magenton kansiorakenne.....	10
Kuvio 3. Magenton moduulin tyypillinen kansiorakenne .....	11
Kuvio 4. Esimerkki moduulin aktivoivasta XML-tiedostosta .....	11
Kuvio 5. Esimerkki moduulin <i>config.xml</i> -tiedostosta.....	12
Kuvio 6. Magenton poolien latausjärjestys.....	13
Kuvio 7. Esimerkki luokan uudelleenkirjoittamisen konfiguraatiosta.....	14
Kuvio 8. Model-luokan lataamisen ja käytön esimerkki .....	15
Kuvio 9. Useamman tuotteen latauksen esimerkki .....	15
Kuvio 10. Workshopin tutkimuksen sykli .....	22

## **Taulukot**

Taulukko 1. Projektien epäonnistumisten syyt.....	18
---	----

## Käsitteet

<b>Autoloader</b>	Joistakin ohjelmointikielistä ja -kehyksistä löytyvä ominaisuus, joka lataa automaattisesti ohjelman tarvitsemat komponentit käyttöön.
<b>Cache</b>	Välimuisti, jota järjestelmät käyttävät usein esimerkiksi tiedon lataamisen ja näyttämisen nopeuttamiseksi.
<b>CRM</b>	Customer Relation Management eli asiakkuudenhallinta.
<b>CRUD</b>	Lyhenne englannin kielen sanoista Create, Read, Update ja Delete, jolla viitataan tietojen käsittelyyn.
<b>HTTP(S)</b>	Hyper Text Transfer Protocol. Tiedonsiirtoprotokolla, jota esimerkiksi selaimet ja palvelimet käyttävät.
<b>Magento</b>	Verkkokauppa-alusta.
<b>MVC</b>	Model-View-Controller. Ohjelmoinnissa monesti käytössä oleva arkkitehtuurityyli.
<b>Moduuli</b>	Järjestelmän erillinen osa, joka on asennettavissa, päivitettävissä tai poistettavissa erikseen.
<b>Ohjelmistokehys</b>	Ennalta valitun ohjelmointikielen avulla kirjoitettu isompi kokonaisuus, joka tarjoaa valmiita ominaisuuksia yleisille toimenpiteille.
<b>PHP</b>	Palvelimilla ajettava web-sovelluksille tarkoitettu ohjelmointikieli.
<b>REST</b>	Representational State Transfer. Tiedonvälityksessä käytettävä rajapinta, joka hyödyntää HTTP-protokollaa.
<b>SugarCRM</b>	Asiakkuudenhallintajärjestelmä.
<b>XML</b>	Extensible Markup Language. Jäsennetyn, tekstimuotoisen datan tallennusmuoto.

# 1 Johdanto

Verkkokauppa-alustoja ja ratkaisuja on maailmalla nykyisin paljon ja ne vaihtelevat helpoista ja kevyistä aina monimutkaisiin ja raskaisiin. Magento asettuu jonnekin edellä mainittujen ääripäiden väliin. Se sopii parhaiten keskisuurille yrityksille, jotka toivovat verkkokaupalta joustavuutta ja mahdollisuutta monipuolisiin ratkaisuihin. Lisäksi Magento on verkkokauppa-alustana yritysmaailman mittakaavassa halpa. Magenton joustavuus ja siinä olevat valmiit ominaisuudet ovat olleet toimeksiantajalla eräinä valintakriteereinä.

Toimiva verkkokauppa ei kuitenkaan välttämättä takaa myyntiä, vaan se vaatii usein asiakkuudenhallintaa. Siksi toimeksiantajalle, Trainer4You Oy:lle, onkin luotu omat prosessit myynnin parantamiseksi ja sen eteenpäin vieminen on ollut eräs keskeisin asia työsuhteeni aikana aina työharjoittelusta lähtien. Nämä prosessit perustuvat verkkokaupan asiakastililtä saatuun analytiikkaan ja potentiaalisten asiakkaiden erottelemiseen asiakkuudenhallintajärjestelmässä, jotta myyntihenkilöstö tietää, ketkä ovat potentiaalisia asiakkaita ja keihin yhteydenotto on kannattavaa. Vaikka alkuperäiset prosessit ovat olleet pitkään toimivia ja riittäviä, ovat ne ajan myötä paisuneet monimutkaisiksi ja siksi nykyiset ratkaisut ovat käyneet raskaiksi ylläpidon ja kehittämisen kannalta. Toimeksiantajan kanssa onkin pitkään kaavailtu parempaa ratkaisua. Tämän opinnäytetyön tarkoitus on kuvata toteutettavan ratkaisun vaatimusmääritellyn prosessia ja lopputulosta. Lopullinen ratkaisun toteutus tehdään tutkimuksen valmistumisen jälkeen, mutta joitakin osa-alueita toteutetaan ainakin osittain jo opinnäytetyön aikana, jotta ratkaisujen toimivuus saadaan varmistettua ja esiteltä.

## 2 Tutkimusasetelma

Tässä luvussa esitellään toimeksiantaja, käydään läpi tutkimuksen tausta, rajaukset ja tavoitteet, esitellään tutkimuskysymykset sekä kuvataan tutkimusmenetelmä.

### 2.1 Toimeksiantaja

Opinnäytetyön toimeksiantajana on vuonna 2004 alun perin Oulussa perustettu

Trainer4You Oy, jolla on nykyisin toimipisteitä Oulun lisäksi mm. Espoossa, Jyväskylässä ja Lappeenrannassa.

Yritys esittelee itsensä suomalaisena personal training -osaamisen uranuurtajayrityksenä. Trainer4You kouluttaa pääasiassa valmentajia liikunta- ja hyvinvointialalle, mutta myös liikunnasta ja hyvinvoinnista yleisesti kiinnostuneet voivat ilmoittautua mukaan. Tarjolla olevia koulutuksia ovat mm. Personal trainer -koulutus, Kuntosali-valmentaja-koulutus, Hyvinvointivalmentaja-koulutus (nykyisin Kehittävän harjoittelun perusteet) ja Ravintovalmentaja-koulutus. Myös sähköisesti tarjolla olevat verkkokurssit ovat yleistymässä.

## 2.2 Opinnäytetyön tausta, rajaukset ja tavoitteet

### **Tausta ja perustelut**

Koska nykyisten analytiikan ratkaisujen hallinnoiminen vaatii ohjelmointitaitoja ja tietämystä ratkaisujen toimintatavoista, ei niiden sellaisenaan säilyttäminen ole tulevaisuuden kannalta järkevää. Toimeksiantajalla ei myöskään ole muita ohjelmointitaitoisia työntekijöitä, mikä rajoittaa hallinnointia entisestään.

Olemassa olevien ratkaisujen parantaminen tai päivitys ei myöskään ole kannattavaa, sillä se vaatisi suuria määriä koodin uudelleen kirjoittamista, jolloin myös virheiden mahdollisuus kasvaa.

Käyttöliittymän kautta konfiguroitavissa oleva moduuli mahdollistaisi työmäärän hajuttamista ja nopeuttaisi tiettyjen prosessien loppuunsaattamista. Samalla päästään suurimmaksi osaksi eroon tarpeesta tehdä jatkuvasti koodimuutoksia haluttujen toimintojen aikaansaamiseksi. Ainoastaan isot toiminnallisuuksien muutokset tai lisäykset vaatisivat ohjelmointia.

### **Tutkimuksen rajaus**

Tutkimus rajataan käsittelemään jo olemassa olevien järjestelmien ympärille kehitettävän toteutuksen vaatimusten selvittämistä. Tutkimuksessa selvitetään, mitä toiminnallisten ja ei-toiminnallisten vaatimusten määrittelyyn liittyy yleisesti. Lopullisia toiminnallisia ja ei-toiminnallisia vaatimuksia ja ratkaisuja avataan tarkemmin, mikäli



se on aiheellista tutkimuksen kannalta. Teoriaa eri järjestelmistä käydään läpi enimmäkseen Magento-verkkokauppa-alustan osalta ja pintapuolisesti SugarCRM-asiakkuudenhallintajärjestelmästä sekä asiakkuudenhallinnasta yleensä.

### **Tutkimuksen tavoitteet**

Tutkimuksen tavoitteena on selvittää, millaisia toiminnallisia ja ei-toiminnallisia vaatimuksia Magenton hallintapaneelista hallinnoitavissa olevaan moduuliin liittyy toimeksiantajan näkökulmasta. Tutkimustulosten pohjalta luodaan liitteeksi (salattu) vaatimusmäärittelydokumentti, jonka pohjalta on myöhemmin, tutkimustyön päätyttyä, tarkoitus toteuttaa vaatimusmäärittelyiden mukainen moduuli.

## **2.3 Tutkimuskysymykset ja menetelmä**

Tutkimuskysymyksiksi saatiin seuraavat kysymykset:

1. Mitä vaatimusmäärittelyllä yritetään saavuttaa?
2. Mitä toiminnallisia vaatimuksia moduuliin liittyy?
3. Mitä ei-toiminnallisia vaatimuksia moduuliin liittyy?

Tutkimusmenetelmänä opinnäytetyössä käytetään kehittämistutkimusta. Tutkimuksessa kehitetään olemassa olevia ratkaisuja ja ratkaistaan niiden ongelmia vaatimusmäärittelyn kautta. Teoriapohjana tutkimuksessa käytetään käytössä olevien järjestelmien ja vaadittujen teknologioiden taustoja sekä asiakkuudenhallinnan ja vaatimusmäärittelyn teoriaa.

Kehittämistutkimusta tehdään yleensä toiminnan parantamiseksi organisaatioiden sisällä ja kohteina voivat olla esimerkiksi prosessit sekä toiminnot, tuotteet, palvelut ja asiantilat (Kananen 2012, 20). Tämän kehittämistutkimuksen kohdetta voidaan pitää prosessina, johon liittyvät sekä myynnin että markkinoinnin erilliset prosessit.

## **3 Järjestelmät**

Seuraavissa alaluvuissa on taustatietoa ja teoriaa niistä järjestelmistä, joita tullaan tämän tutkimuksen aikana käsittelemään. Kyseiset järjestelmät ovat Trainer4You:lla jo käytössä eikä niille tulla etsimään korvaavia järjestelmiä tutkimustyön aikana. Näitä järjestelmiä ovat Magento ja SugarCRM. SugarCRM käsitellään kevyemmin, kun taas

Magento on tutkimuksen kannalta tärkeämpi ja käsitellään siksi laajemmin.

### 3.1 Magento

Toimeksiantajan verkkokaupan taustalla pyörii Magento 1.X Community Edition, joka on otettu käyttöön 2013-2014 vuoden vaihteessa ja jota on sen jälkeen tarpeiden mukaan kustomoitu ja laajennettu ulkopuolisen palveluntarjoajan ja verkkokaupan ylläpitäjän toimesta, kolmannen osapuolen moduuleilla sekä työsuhteeni aikana toteuttamillani kustomoinneilla ja laajennuksilla. Nykyisin kyseinen järjestelmä onkin melko kaukana perinteisestä Magento-ympäristöstä useiden komponenttien osalta ja sisältää useita ominaisuuksia, joita ei Magentossa oletuksena ole. Näitä ominaisuuksia ei voida tutkimuksessa avata enempää toimeksiantajan pyynnöstä.

#### 3.1.1 Historiaa

Magento on vuonna 2007 Kaliforniassa, Yhdysvalloissa alkunsa saanut e-commerce eli verkkokauppa-alusta. Tuolloin Magenton kehittäjäryitys Varien teki yhteistyötä osCommercen kanssa ja tavoitteena oli kehittää osCommercen alustaa, joka ennen Magenton tuloa oli markkinoiden johtava verkkokauppa-alusta. Eri tavoitteiden ja muiden erimielisyyksien takia Varien päätti kuitenkin rakentaa täysin oman verkkokauppa-alustansa. (Morrone 2015.)

Magenton ensimmäinen julkinen beta julkaistiin elokuussa 2008 ja se nousi välittömästi suosioon, täyttäen suurtenkin yritysten tarpeet. Magentosta on suosion myötä aina ollut tarjolla kahta eri versiota; ilmainen Community Edition ja maksullinen Enterprise Edition. Varien aloitti Enterprise Editionin kehittämisen Community Editionin ohella, koska suurilla yrityksillä oli siihen tarvetta ja ne olivat valmiita maksamaan siitä. (Morrone 2015.)

Enterprise Edition pohjautuu Community Editioniin, mutta se sisältää enemmän ominaisuuksia ja esimerkiksi parempaa tukea suurilla käyttäjä- ja tuotemääriä varten, kuten verkkokaupan hajauttamisen useammalle palvelimelle rasituksen helpottamiseksi ja laajemman, sivukohtaisen välimuistin (cache) käytön sivujen latauksen nopeuttamiseksi. Enterprise Edition maksaa vuodessa noin \$15 500, jonka takia pienemmät

yritykset luultavasti kallistuvat ilmaisen Community Editionin suuntaan. (How to Decide? Magento Enterprise vs. Community n.d.) Pitää silti muistaa, että myös ilmaisessa versiossa on ylläpitomaksut, jotka määräytyvät palveluntarjoajan sekä muiden tekijöiden, kuten esimerkiksi palvelimen kapasiteetin ja nopeuden mukaisesti.

Vuonna 2010 eBay osti Varienista 49% prosenttia ja vuonna 2011 eBay osti yrityksen kokonaan. Pian tämän jälkeen Varien vaihtoi nimensä eBayn omistuksesta johtuvien muutosten takia ja on sittemmin tunnettu nimellä Magento Inc. tai yleisimmin vain Magento. Lisäksi eBayn omistuksen myötä Magentoon lisättiin tuki eBayn omistamalle PayPal-maksuvaihtoehdolle. (Morrone 2015.)

### 3.1.2 Perusteet

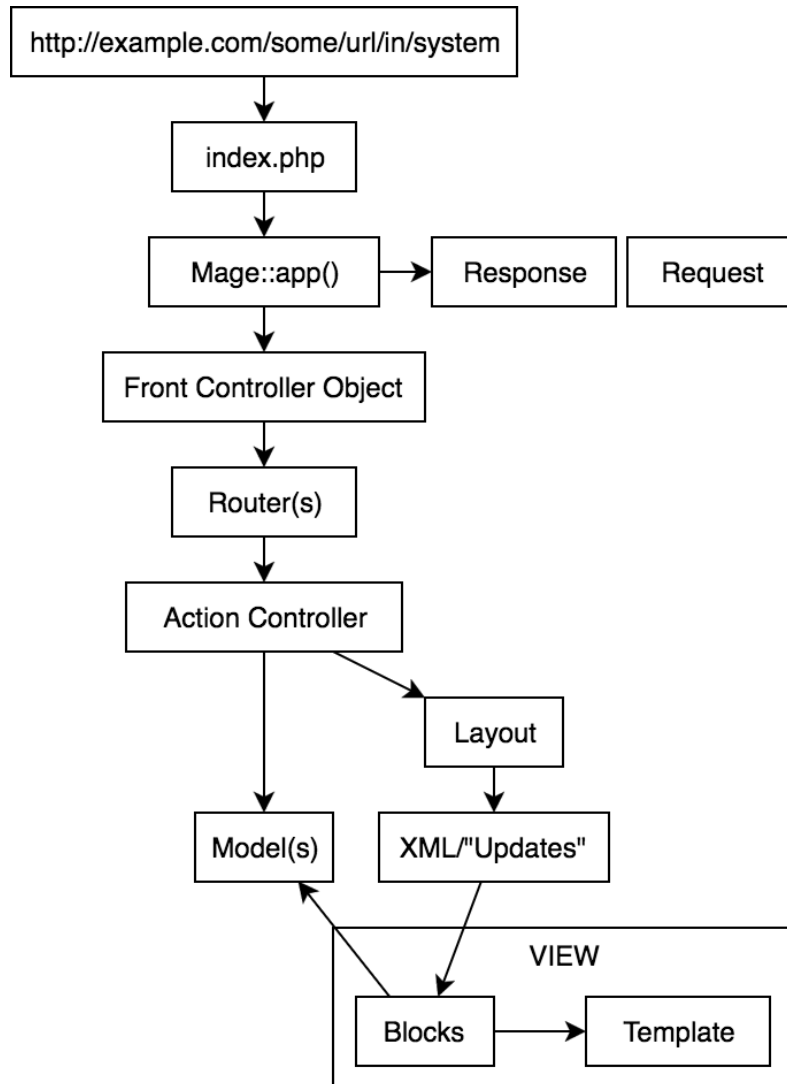
Magento on vahvasti luokkapohjainen, olio-ohjelmointiin perustuva PHP-ohjelmointikielellä kehitetty verkkokauppa-alusta, joka on rakennettu Zend Framework 1:n päälle. Zend Framework taas on PHP-ohjelmointikielellä kehitetty ohjelmistokehys. Magentoa ryhdyttiin kehittämään Zend Frameworkin päälle siinä valmiiksi olevien komponenttien takia. Näitä komponentteja ovat esimerkiksi Zend\_Cache, Zend\_Locale ja Zend\_Currency, yhteensä kuitenkin noin 15 komponenttia. Kyseisiä komponentteja ei käytetä suoraan, vaan niitä on laajennettu tarpeiden mukaan. Nämä laajennukset on tehty Varien\_-alkuisiin luokkiin, jotka laajentavat samalla nimeämisperiaatteella olevia Zend\_-alkuisia luokkia. (MacGregor 2015, 2.)

Zend Frameworkin käyttö mahdollisesti seuraavat periaatteet (MacGregor 2015, 2.):

- **Ylläpidettävyys:** Magenton ydinkoodi saatiin eroteltua kustomoinneista ja kolmannen osapuolen moduuleista.
- **Päivitettävyys:** Magenton modulaarisuus mahdollistaa kustomointien ja moduulien päivittämisen erikseen.
- **Joustavuus:** Mahdollistaa saumattoman kustomoinnin ja yksinkertaistaa uusien ominaisuuksien kehittämistä.

Zend Framework käyttää websovelluksista yleisesti käytössä olevaa Model-View-Controller (MVC) rakennetta, jota Magento ei kuitenkaan käytä aivan perinteisellä tavalla. Sen sijaan Magentossa käytetään kuviossa 1 kuvattua niin sanottua konfiguroi-

tavaa MVC-mallia, jossa XML-muotoiset konfiguraatio ja layout -tiedostot ovat keskeisessä roolissa. Yksinkertaistettuna tämä tarkoittaa sitä, että pelkän tiedoston/luokan luominen ei riitä, vaan on tehtävä konfigurointia, jotta Magento ymmärtää luotujen tiedostojen/luokkien olemassaolon. (MacGregor 2015, 13.)



Kuvio 1. Magenton MVC-rakenne (alkuperäinen kuvio Storm 2009a.)

Lisäksi Magentolle on kehitetty hieman omanlaisensa kansiorakenne, joka poikkeaa myös perinteisestä MVC-mallin kansiorakenteesta jossa kaikki websovelluksen logiikka, kuten kontrollerit, modelit ja viewit, sijaitsevat samojen kansiodien sisällä, usein sovelluksen juuressa tai erikseen määritetyssä kansiossa. Magentossa kyseinen logiikka on siirretty moduulien osiksi, joka on osasyynä siihen miksi siinä täytyy tehdä

niin paljon konfigurointia. (Storm.)

```
-app
  -code
  -design
  -locale
-js
-media
-lib
-skin
-var
```

Kuvio 2. Magenton kansiorakenne (MacGregor 2015, 3.)

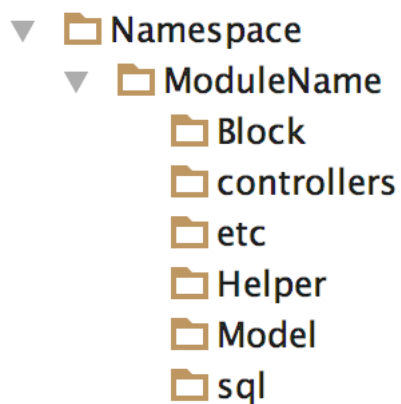
Kuviossa 2 listatuista kansioista tärkein moduulien rakentamisessa on *app*-kansio. Sen sisällä oleva *code*-kansio sisältää moduulien varsinaisen logiikan/koodin, *design*-kansio sisältää tarvittavat tiedostot moduulin visuaalista rakennetta varten ja *locale*-kansiossa on mahdolliset kielikäännökset. (MacGregor 2015, 3.)

### 3.1.3 Modulaarisuus ja moduulirakenne

Magento on modulaarinen. Tämä tarkoittaa sitä, että kaikki Magenton osat on eroteltu omiksi moduuleikseen, jotka ovat asennettavissa, päivitettävissä ja myös poistettavissa toisistaan riippumatta, olettaen että jokin toinen moduuli ei ole riippuvainen poistettavasta moduulista. Lisäksi Magenton *app/code*-kansio on jaettu kolmeen eri osaan tai pooliin (code pool), jotka MacGregor (2015, 6) esittelee seuraavanlaisesti:

- **Core pool** sisältää Magenton ydinkoodin eikä sitä yleisten käytäntöjen mukaisesti saisi muokata suoraan.
- **Community pool** on tarkoitettu kolmannen osapuolen kehittämille moduuleille, joita löytää helposti verkosta.
- **Local pool** on tarkoitettu juuri nykyiseen ympäristöön kehitetyille moduuleille ja mahdollisille ydinkoodin muutoksille.

Jotta moduulit toimivat oikein, niissä on käytettävä Magenton ymmärtämää moduulien kansiorakennetta, jonka tyypillinen rakenne on kuvattu kuviossa 3. Kuvioista näkee, että aiemmin mainittu websovellusten logiikka sijaitsee moduulien sisällä. Kuviossa oleva Namespace viittaa yleensä moduulin tekijään, kuten esimerkiksi yrityksen nimeen. ModuleName taas on moduulin varsinainen nimi. Soveltaen tätä nimeämistapaa aikaisemmin mainittuun code pool -rakenteeseen, esimerkiksi Trainer4You-yrityksen itselleen kehittämä Example-moduuli tulisi sijaita kansiossa *app/code/local/Trainer4You/Example*.



Kuvio 3. Magenton moduulin tyypillinen kansiorakenne

### 3.1.4 Moduulien konfigurointi

Kuten aiemmissa luvuissa on tullut ilmi, Magenton moduulit vaativat konfigurointia toimiakseen. Se tapahtuu XML-tiedostojen avulla. Tiedostoilla on ennalta määrättyt hakemistot ja nimet.

```

<config>
  <modules>
    <Trainer4You_Example>
      <active>true</active>
      <codePool>local</codePool>
    </Trainer4You_Example>
  </modules>
</config>
  
```

Kuvio 4. Esimerkki moduulin aktivoivasta XML-tiedostosta

Kaikilla Magenton moduuleilla on oltava kuvion 4 kaltainen XML-tiedosto kansiossa *app/etc/modules*, joka kertoo Magentolle moduulin olemassaolosta, sen aktiivisuudesta ja poolista. Tämän kansion XML-tiedostoilla on *Namespace\_ModuleName.xml* nimeämiskäytäntö. Täten Trainer4You-yrityksen Example moduulin tiedostonimeksi tulisi *Trainer4You\_Example.xml*. (Storm 2009b.)

```
<config>
  <modules>
    <Trainer4You_Example>
      <version>0.0.1</version>
    </Trainer4You_Example>
  </modules>
  <frontend>
    <routers>
      <trainer4youexample>
        <use>standard</use>
        <args>
          <module>Trainer4You_Example</module>
          <frontName>trainer4youexample</frontName>
        </args>
      </trainer4youexample>
    </routers>
  </frontend>
  <global>
    <models>
      <trainer4you_example>
        <class>Trainer4You_Example_Model</class>
      </trainer4you_example>
    </models>
  </global>
</config>
```

Kuvio 5. Esimerkki moduulin *config.xml*-tiedostosta

Lisäksi moduuleilla tulee olla vähintään kuvion 5 kaltainen *config.xml* niminen tiedosto moduulin *etc* kansiossa. Kyseisessä tiedostossa määritellään moduulin toimintaan vaikuttavat kontrollereiden, mallien, helpereiden sekä muiden komponenttien asetukset (MacGregor 2015, 66). Toinen tärkeä asetustiedosto on *system.xml*, jossa määritellään Magenton järjestelmäasetuksien kautta määritettävät asetukset ja näkymät. (Storm 2010a.)

### 3.1.5 Luokkien laajentaminen, yli- ja uudelleenkirjoittaminen

Magentoon kehitettyjen moduulien luokat yleensä laajentavat jotakin Magenton omista luokista. Näin saadaan varmistettua, että Magenton logiikka säilyy myös moduulin luokkien sisällä. Laajennettava luokka määrittyy yleensä kyseessä olevan moduulin luokan ja käyttötarkoituksen mukaan. Esimerkiksi moduulien frontend controllerit laajentavat yleensä Magenton *Mage\_Core\_Controller\_Front\_Action*-luokkaa.

Moduulien luokkien ylikirjoittaminen (override) on Magentossa yksinkertaisimmillaan helppoa. Esimerkiksi core poolissa olevien luokkien ylikirjoittaminen onnistuu kopioimalla sama kansio-, tiedosto- ja luokkarakenne community tai local pooliin. Myös community poolissa olevat luokat voi ylikirjoittaa kopioimalla sama rakenne local pooliin, mutta local poolissa olevia luokkia ei voi ylikirjoittaa tällä tavalla. Poolien latausjärjestys on kuvattu kuviossa 6.

```
local -> community -> core
```

Kuvio 6. Magenton poolien latausjärjestys (MacGregor 2015, 7.)

Omien ja muidenkin kokemusten mukaan community ja local poolit ovat hieman harmaalla alueella, sillä monesti kolmannen osapuolen kehittäjät ovat rakentaneet moduulinsa käyttämään local poolia. Tämä ei sinänsä ole ongelma toiminnan kannalta, mutta näissä tapauksissa moduulien luokkien ylikirjoittaminen edellä mainitulla tavalla on mahdotonta (Reitsma 2011). Luokkien ylikirjoittaminen ei ole muutenkaan paras mahdollinen tapa muokata niitä, sillä moduulien päivitysten myötä ylikirjoitetut luokat eivät päivity lainkaan. (Storm 2010b.)



```

..
<global>
  <models>
    <customer>
      <rewrite>
        <customer>Trainer4You_Customer_Model_Customer</customer>
      </rewrite>
    </customer>
  </models>
</global>
..

```

Kuvio 7. Esimerkki luokan uudelleenkirjoittamisen konfiguraatiosta (alkuperäinen kuvio Soni 2015.)

Parempi tapa muuttaa Magentoön kirjoitettuja luokkia ja tarvittaessa vain joitakin niissä olevia funktioita on luokkien uudelleenkirjoittaminen (rewrite), jonka konfigurointi on kuvattuna kuviossa 7. Se vaatii siis moduulin kirjoittamista ja konfigurointia, mutta se ei ylikirjoita koko luokkaa vaan vain niiltä osin mitä luokassa on uudelleenkirjoitettu. Tätäkin kutsutaan monesti ylikirjoittamiseksi. Kokonaan uusien funktioiden kirjoittaminen luokkiin ei tässä tapauksessa ylikirjoita mitään (Soni 2015). Uudelleenkirjoittamalla luokkia myös moduulien päivitykset tulevat varmemmin voimaan, olettaen että päivityksessä ei ole muutoksia uudelleenkirjoitettuihin luokkien funktioihin. Näissä tapauksissa päivityksen muutokset täytyy tarkistaa ja lisätä uudelleenkirjoitettuihin funktioihin, jotta ne tulevat voimaan.

### 3.1.6 Luokkien käyttäminen

Magentoossa käytetään sille ominaista luokkien latausta, joka helpottaa kehittäjien työtä monin tavoin. Kyseessä on eräänlainen *autoloader*-luokka, joka on rakennettu PHP5:stä löytyvän `__autoload()`-funktion päälle. Kyseinen luokka lataa automaattisesti käyttöön vain ne luokat, jotka kulloinkin tarvitaan. Tämä on tarpeen, kun kyseessä on yli 20 000 tiedostosta koostuva ohjelmistokehys. (MacGregor 2015, 5.)

Ajatellaan, että haluamme muuttaa jonkin kaupastamme löytyvän tuotteen tiettyä arvoa, kuten esimerkiksi hintaa. Magento tarjoaa moduulien model-luokkien lataamiseen `Mage::getModel()`-metodin, jonka avulla mukana kulkeutuu myös yleisimmin käytettävät toiminnot, kuten tietueiden lataus, arvojen lataus ja asettaminen sekä tallennus. Näin kaikki tietokantatoiminnot saadaan kirjoitettua helposti käyttäen

PHP-kieltä. Samanlaisia metodeita on myös muiden tyyppisille luokille, kuten esimerkiksi helpereille (*Mage::helper()*). (Storm.)

```
$model = Mage::getModel('catalog/product')->load(20);
$price = $model->getPrice();
$price += 5;
$model->setPrice($price);
$model->save();
```

Kuvio 8. Model-luokan lataamisen ja käytön esimerkki (alkuperäinen kuvio Storm.)

Kuviossa 8 ladataan ensin *Mage\_Catalog\_Model\_Product*-luokka käyttäen *getModel()*-metodia ja sen avulla tuote id:llä 20, joka viittaa haluamaamme tuotteeseen. Kaikilla tuotteilla, asiakkailla, tilauksilla ja samankaltaisilla tietueilla on id, joiden avulla ne saadaan ladattua käyttöön. Tämän jälkeen haetaan tuotteen *price*-arvo eli hinta käyttäen *getPrice()*-metodia, lisätään sen arvoa viidellä ja asetetaan uusi hinta *setPrice()*-metodilla. Kyseisiä metodeita ei tässä tapauksessa, kuten ei useasti muutenkaan, ole olemassa luokassa, sillä Magentossa automatisoidaan *get* ja *set*-metodien toiminta hyödyntäen PHP:n *\_\_call*-metodia, jollei kyseisiä metodeita ole erikseen kirjoitettu luokkaan. Myös *save()*-funktio eli tallennus on yleinen toiminto, jota ei tarvitse Magentossa erikseen itse kirjoittaa. (Storm.)

```
$products = Mage::getModel('catalog/product')
->getCollection()
->addAttributeToSelect('*')
->addFieldToFilter('price', '5.00');

foreach($products as $product)
{
    echo $product->getName();
}
```

Kuvio 9. Useamman tuotteen latauksen esimerkki (alkuperäinen kuvio Storm.)

Mikäli halutaan ladata esimerkiksi useampi tuote, täytyy käyttää Magenton tarjoamaa *Collections*-liitäntää (interface), jonka käyttö on esitelty kuviossa 9. Esimerkin *addFieldToFilter()* sisältää säännön, jonka perusteella ladataan vain tuotteet joiden hinta on 5.00. Tämän jälkeen kaikkien löytyneiden tuotteiden tiedot on käytettävissä esimerkiksi PHP:n *foreach*-funktion avulla. Esimerkissä tulostetaan vain tuotteiden nimet, mutta tuotteiden tietojen muuttaminen onnistuisi myös kuvion 8 kaltaisesti. (Storm.)

Magento tarjoaa siis useita automatisointeja ja helpottaa kehittäjien työtä, kunhan logiikan on ensin oppinut ja konfiguroinnit ovat kunnossa. Tämä voi viedä jonkin verran aikaa, etenkin jos on tottunut PHP:n tavanomaiseen ohjelmointiin.

## 3.2 SugarCRM

SugarCRM on toimeksiantajalla käytössä oleva asiakkuudenhallinta- eli CRM-järjestelmä, joka on otettu käyttöön keväällä 2014. Järjestelmän avulla toimeksiantajan markkinointi- ja myyntihenkilöstö pystyy erottelemaan potentiaaliset asiakkaat ja heidän kiinnostuksen kohteensa sekä tarjoamaan heille kohdennetumpaa markkinointia ja myyntiä. Tämä ei kuitenkaan olisi mahdollista ilman nykyisiä analytiikan ratkaisuja, joita tässä tutkimuksessa käsitellään.

### 3.2.1 Asiakkuudenhallinta

CRM (Customer Relationship Management) eli asiakkuudenhallinta ei tarkoita vain teknisiä järjestelmiä joilla sitä hallinnoidaan, vaikka monesti niin mielletäänkin. Sen sijaan pitäisi ajatella, että CRM kattaa yrityksen liiketoimintaprosessit kolmella osa-alueella. Näitä ovat markkinointi, myynti ja asiakaspalvelu. Jokaisella näistä osa-alueista vaaditaan hieman eri toimenpiteitä ja osaamista, jonka takia näitä ei kannattaisi lähteä yhdistelemään vaan kehittää niille omat prosessinsa. (Sahlsten 2012.)

Kunnollinen asiakkuudenhallinta tarjoaa yritykselle selkeämmän käsityksen asiakkaiden tilanteista ja toiveista. Kun nämä ovat selvillä, pystytään asiakkaille tarjoamaan markkinoinnin ja myynnin kautta oikeanlaisia palveluita sekä tuotteita.

### 3.2.2 On-Demand

Toimeksiantajalla SugarCRM-järjestelmä on käytössä eräänlaisena On-Demand -palveluna ulkoisen palveluntarjoajan kautta. SugarCRM On-Demand tarkoittaa sitä, että SugarCRM ylläpitää ja hallinnoi järjestelmää ilman, että asiakkaan tarvitsee murehtia päivityksistä tai muista ylläpitoon liittyvistä asioista. (Multiple Deployment Options n.d.)

SugarCRM On-Demand asettaa kuitenkin asiakkaille melko suuria rajoituksia. Se ei esimerkiksi mahdollista asiakkaalle suoraa pääsyä palvelimelle esimerkiksi SSH tai SFTP-yhteyksien kautta, joka rajoittaa kustomointien tekoa järjestelmään. Etenkään tästä syystä tutkimuksen aiheena olevaa ratkaisua ei yritetä suunnitella tai rakentaa kyseisen järjestelmän puolelle, vaikka se olisi teoriassa mahdollista.

### 3.2.3 Teknologiat ja REST -rajapinta

SugarCRM on kehitetty avointen standardien ja teknologioiden, kuten HTML5:n, PHP:n sekä JavaScriptin päälle ja toimii vahvasti REST-rajapinnan avulla. Rajapinta onkin suositeltu tapa integroida ulkoisia järjestelmiä SugarCRM-järjestelmän kanssa, mutta myös lähes kaikki web-sovelluksen sisällä tapahtuvat normaalit toimenpiteet, kuten tietueiden avaaminen ja tallentaminen, hyödyntävät kyseistä rajapintaa. Toisin sanoen, mikäli jokin toimenpide on mahdollista suorittaa järjestelmän käyttöliittymän kautta, on se mahdollista suorittaa myös REST-rajapinnan avulla. SugarCRM koostuu Magenton tavoin moduuleista. Eri moduuleissa olevat tietueet voidaan linkittää toisiinsa, jolloin saadaan rakennettua suurempia tietueiden kokonaisuuksia. (Sugar Developer Guide 7.7 - Architecture n.d.)

REST on lyhenne englannin kielen sanoista Representational State Transfer ja se on yleisessä käytössä oleva teknologia websovelluskehityksessä. Se mahdollistaa kevyiden, ylläpidettävien ja skaalautuvien www-sovelluspalveluiden (Web services) ja web-sovellusten rakentamisen. REST-rajapinnoissa hyödynnetään yleisesti HTTP-protokollaa tai vaihtoehtoisesti suojattua HTTPS-protokollaa CRUD (Create/Read/Update/Delete) komentojen suorittamiseen. Komennoista saadun vastauksen data on yleensä joko JSON tai XML muodossa, riippuen rajapinnan käyttötarkoituksesta ja halutun datan monimutkaisuudesta. (Vaqqas 2014.)

HTTP-protokolla on nykypäiväisen modernin webin perusta, sillä lähes kaikki verkossa tapahtuva liikenne tehdään sen avulla. Esimerkiksi yleisimmät verkkoselaimet käyttävät kyseistä protokollaa verkkosivujen lataamiseen ja näyttämiseen. HTTP on lyhenne englannin kielen ”Hypertext Transfer Protocol” sanoista ja HTTPS on HTTP-protokollan suojattu versio. (Podila 2013.)

## 4 Vaatimusmäärittely

Vaatimusten määrittely on tärkeä osa jokaista tietojärjestelmäprojektia. Projektien onnistuminen on tiukasti kytköksissä vaatimusten laatuun, sillä mikäli niitä ei ole tehty kunnolla, projektit epäonnistuvat, ovat myöhässä, paisuvat yli budjetin tai luovat järjestelmiä joita ei koskaan käytetä. (Alexander & Stevens 2002, 1.)

Incomplete requirements	13,1%
Didn't involve users	12,4%
Insufficient resources/schedule	10,6%
Unrealistic expectations	9,9%
Lack of managerial support	9,3%
Changing requirements	8,7%
Poor planning	8,1%
Didn't need it any longer	7,4%

Taulukko 1. Projektien epäonnistumisten syyt (alkuperäinen taulukko Alexander & Stevens 2002, 3.)

Ohjelmistojen pienikokoisuuksien takia vaatimuksia ei aivan ohjelmistotuotannon alkuaikoina määritelty, mutta ohjelmistojen kasvaessa ja monimutkaistuessa niiden laatu alkoi laskea huonosti toteutetun vaatimusmäärittelyn takia. Jo vuonna 1976 julkaistun tutkimuksen mukaan riittämättömät, epätäydelliset, epäyhtenäiset tai moniselitteiset vaatimukset ovat vaikuttaneet ohjelmistojen laatuun. (Paakki 2011, 5.)

Vaatimusmäärittely on toki edistynyt 1970-luvun jälkeen, mutta se on silti yksi kehittymättömin ohjelmistokehityksen osa. Tämä johtuu osittain vaatimusmäärittelyn vaikeudesta, mutta nykyisin järjestelmät ovat hyvin monimutkaisia, jolloin vaatimuksia

on paljon ja ne ovat riippuvaisia toisistaan. (Paakki 2001, 6.)

Alun perin Standish Groupin vuonna 1995 julkaisemat luvut, jotka ovat taulukossa 1, kertovat vajanaisten tai huonojen vaatimusten olleen suurin syy tietojärjestelmäprojektien epäonnistumiselle. Tämän lisäksi myöhemmin, vuonna 1999 toteutettu kysely on kertonut, että vain 22% projektipäälliköistä on kokenut vaatimusten määrittämisen tärkeäksi. (Alexander & Stevens 2002, 4.)

Vaatimusmäärittelyn lopullisena tuotoksena on vaatimusmäärittelydokumentti, josta selviävät löydetty vaatimukset. (Paakki 2011, 35.)

#### 4.1 Vaatimusmäärittelyn osapuolet

Vaatimukset ovat pääsääntöisesti tarkoitettu käyttäjiä, kehittäjiä, vaatimusinsinöörejä, asiakkaita ja sidosryhmiä varten. Käyttäjät ovat yleensä loppukäyttäjiä, mutta niillä voidaan viitata myös testaajiin tai ylläpitäjiin lopullisen tuotteen tai järjestelmän ollessa käytössä. Kehittäjillä viitataan kaikkiin kehitysvaiheessa mukana oleviin, kuten järjestelmäinsinööreihin, ohjelmoijiin ja testausinsinööreihin. Vaatimusinsinöörit pääsääntöisesti keräävät vaatimukset eri osapuolilta. (Alexander & Stevens 2002, 6.)

Asiakkaita luetaan yleensä ne, jotka maksavat kehitystyöstä ja ovat enemmän managereita kuin käyttäjiä, vaikka voivat olla kumpiakin. Asiakkaat ovat vastuussa organisaationsa maineesta, osastonsa tuotteliaisuudesta ja mahdollisesti muistakin osaluista, kuten työntekijöidensä moraalista, jotka kaikki voivat riippua käytettävien järjestelmien laadusta. Asiakkaat saattavat myös asettaa rajoja vaatimusmäärittelyn laajuuteen. (Alexander & Stevens 2002, 6.)

Sidosryhmillä viitataan niihin osapuoliin, joiden on syytä pystyä vaikuttamaan vaatimuksiin. Käyttäjät kuuluvat sidosryhmiin lähes poikkeuksetta, mutta myös muut edellä mainitut osapuolet voivat kuuluvat sidosryhmiin. Alexander & Stevens (2002, 7) lukevat sidosryhmiin kuuluviksi myös esimerkiksi:

- Ihmiset, joiden elämään järjestelmä vaikuttaa, kuten asiakkaat ja tavarantoimittajat.
- Järjestelmän kannattavuudesta huolehtivat managerit, vaikka he eivät järjestelmää käyttäisikään.

- Säätelymekimet, kuten hallitukset ja standardien säätelijät, jotka huolehtivat järjestelmän vaikutuksesta ympäristöön.

Vaativusmäärittelyn tärkeimmät tavoitteet voidaan jakaa Alexander & Stevensin (2002, 4) mukaan kuuteen osaan. Vaativusten kautta on mahdollista:

1. esittää sidosryhmien asettamat toiveet
2. antaa sidosryhmille mahdollisuus kertoa toiveensa
3. esittää sidosryhmien näkökulmat
4. varmistaa järjestelmän oikeellisuus
5. mitata projektin etenemistä
6. asettaa kriteerit projektin hyväksymistä varten.

## 4.2 Vaativukset

Vaativukset tulevat aina sidosryhmiltä ja ne voivat olla joko toiminnallisia tai ei-toiminnallisia. Järjestelmälle voidaan asettaa myös joitakin erillisiä rajoitteita vaativusten kautta. (Paakki 2011, 3.)

Vaativusmäärittelyä voidaan toki toteuttaa muillakin tavoin, mutta tämän tutkimuksen kohdalla on päädytty pääasiassa toiminnallisten ja ei-toiminnallisten vaativusten määrittelyyn. Jokaisen projektin kohdalla tulisi kuitenkin soveltaa vaativusmäärittelyn toteutustapaa projektille sopivaksi.

### 4.2.1 Toiminnalliset vaativukset

Toiminnallisilla vaativuksilla viitataan siihen, mitä järjestelmällä tulee pystyä tekemään, toisin sanoen niillä viitataan järjestelmän toimintoihin (Paakki 2011, 27). Esimerkkinä voidaan ajatella sitä, että mitä toimintoja käyttäjän tulee pystyä järjestelmässä tekemään, esimerkiksi käyttöliittymän kautta.

### 4.2.2 Ei-toiminnalliset vaativukset

Ei-toiminnallisilla vaativuksilla tarkoitetaan järjestelmälle määritettyjä ehtoja, jotka kertovat miten tai millä tavoin sille asetetut toiminnalliset vaativukset tulee toteuttaa. Ei-toiminnalliset vaativukset voidaan luokitella Paakin (2011, 30) mukaan seuraavasti:

- Laatuvaatimukset
  - Käyttöturvallisuus, turva (tietoturva), luotettavuus, suorituskyky, liittymät, käyttöliittymä, käytettävyys, käyttömukavuus, tarkkuus
- Mukautuvuusvaatimukset
- Arkkitehtuurivaatimukset
  - Asennettavuus, hajautus
- Kehitystyön vaatimukset
  - Kustannukset, takaraja, monimuotoisuus, ylläpidettävyys, siirrettävyys, uudelleenkäytettävyys, testattavuus

#### 4.2.3 Rajoitteet

Rajoitteet ovat eräänlaisia erityislaatuksia vaatimuksia, jotka eivät sovellu täysin toiminnallisiin tai ei-toiminnallisiin vaatimuksiin. Näitä voivat olla esimerkiksi jotkin järjestelmän laillisuuksiin liittyvät seikat, yhteensopivuuden varmistaminen käytettävien ohjelmointikielten osalta ja laadun varmistaminen standardien avulla. (Alexander & Stevens 2002, 10.)

#### 4.3 Vaatimusten selvittäminen

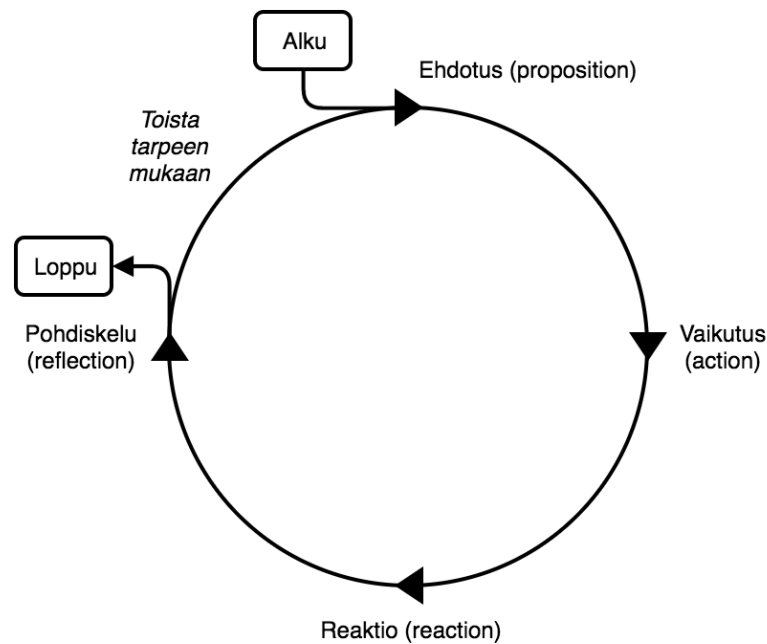
Vaatimusten selvittämiseen on olemassa monia keinoja, joista muutamia ovat esimerkiksi: käyttäjien ja muiden sidosryhmien haastattelut, workshopien järjestäminen, käyttäjien työskentelyn tarkkailu, aiheeseen liittyvien asiakirjojen etsiminen ja käyttäjien tekemien järjestelmien muutosten löytäminen. Arvokkaiksi vaatimusten lähteiksi voivat osoittautua myös ongelmaraportit ja parannusehdotukset, kunhan niiden tärkeys vaatimusten kannalta on selvillä. (Alexander & Stevens 2002, 28.)

Haastatteluissa tulisi olla selvillä haastattelun kohderyhmät ja ne tulisi dokumentoida myöhempää tarkastelua varten joko tekstin, äänen tai videon muodossa. Kysymysten tulisi olla avoimia, kuten ”Miten kuvailisit tämän asian?”, johdattelevien sijaan. Visuaaliset apuvälineet, kuten esimerkiksi diagrammit tai luonnokset, ovat myös monesti tehokkaita työkaluja haastatteluissa. (Alexander & Stevens 2002, 31.)

Workshopien etuna ovat useiden sidosryhmien osallistumisen kautta rakentuvat vaatimukset, jotka ovat monesti laadultaan parempia useiden mielipiteiden ollessa



esillä. Workshopit vaativat yleensä kuitenkin enemmän valmistelua kuin haastattelut. Jokaisella workshopin jäsenellä tulee olla selvillä, mitä heiltä odotetaan ja heillä tulee olla mahdollisuus tuoda oma näkemyksensä esille. Siksi on järkevää noudattaa tiettyä prosessia eri ehdotusten arviointia ja hyväksyntää varten, jonka yksi malli on kuvattu kuviossa 10. (Alexander & Stevens 2002, 39.)



Kuvio 10. Workshopin tutkimuksen sykli (alkuperäinen kuvio Alexander & Stevens 2002, 42.)

Käyttäjien tarkkailussa on tärkeää antaa käyttäjien itse kertoa samalla mitä he ovat tekemässä, mitä ongelmakohtia heidän mielestään on ja mitä olisi hyvä parantaa. Ai-noastaan tarkentavat kysymykset ovat aiheellisia. Haastatteluiden tapaan myös tarkkailun nauhoittamista kannattaa harkita. (Alexander & Stevens 2002, 44.)

#### 4.4 Dokumentointi

Vaatusmäärittelyn dokumentointiin on olemassa useita malleja, mutta tärkeimpänä yhteisenä tekijänä voidaan pitää sitä, että dokumentin on oltava kaikkien sidosryhmien ymmärrettävissä. Siksi siinä tulisi käyttää hyvin jäsenneltyä, otsikoitua ja huolellisesti kirjoitettua arkikieltä. Apuna on hyvä käyttää grafiikkaa siinä määrin

missä se on kannattavaa. (Wiegers 2002, 165.)

Vaatimusmäärittelyn mallien standardeissa, kuten *IEEE 830-1998*, voi kuitenkin olla joitakin puutteita tai ne eivät yksinkertaisesti sovi joka projektiin. Siksi Wiegers (2002, 171) ehdottaakin tarvittaessa etsimään sopivat mallit projektikohtaisesti tai sovelta-  
maan eri malleja projekteihin sopiviksi. Sisällysluettelon ja versiohistorian sisällyttä-  
minen vaatimusmäärittelydokumenttiin on Wiegersin mukaan monesti myös kannat-  
tavaa.

## 5 Tutkimuksen toteutus

Koko tutkimuksen ajan hyödynnettiin jo aiemmin opittua teoriaa ja kertynyttä tek-  
nistä osaamista sekä Magento- että SugarCRM-järjestelmistä. Moduulin kannalta  
uutta osaamista kartutettiin kuitenkin eniten Magenton osalta lukemalla kirjalli-  
suutta ja verkosta löytyviä artikkeleita sekä suorittamalla eri lähteistä löytyneitä tuto-  
riaaleja ja esimerkkejä.

Moduulin vaatimusmäärittely aloitettiin tutkimalla toimeksiantajalla käytössä olevia  
analytiikan ratkaisuja tekniseltä kannalta. Koska ratkaisut on toteutettu toimeksian-  
tajalle talon sisäisesti työsuhteen aikana ja niiden asettamat vaatimukset olivat jo tie-  
dossa, saatiin suuri osa sekä toiminnallisista, ei-toiminnallisista että muista vaatimuk-  
sista selvitettyä tätä kautta. Vaikka yhteneväisyyksiä eri analytiikan ratkaisujen välillä  
olikin, myös eroja huomattiin etenkin eri aktiviteeteista saatujen talletettavien tieto-  
jen muodon ja määrän kannalta.

Samalla otettiin huomioon CRM-järjestelmään suunniteltuja muutoksia, joista tärkein  
on datan hajautus kahteen osaan eli moduuliin: asiakkaan perustietoihin ja analytiik-  
kaan eli asiakkaan aktiivisuudesta saatuihin tietoihin. Tämä on tärkeä muutos nykyi-  
seen verrattuna, sillä analytiikan määrä on ollut jatkuvassa kasvussa ja siitä saatujen  
tietojen tallentaminen suoraan asiakkaan tietoihin kasvattaa tietueita liiallisesti, jol-  
loin jopa tietojen häviäminen on riskinä.

Kun vaatimuksia alkoi olla selvillä, ryhdyttiin selvittämään, miten moduuli tulisi olla  
konfiguroitavissa käyttöliittymän kautta ja miten sen automaation tulisi toimia. Tässä

vaiheessa toimeksiantajan eri osapuolet alkoivat olla enemmän osallisena vaatimusmäärittelyssä, jotta etenkin moduulin käytettävyyteen liittyviä vaatimuksia saatiin selville. Myös käytössä olevia myynnin ja markkinoinnin prosesseja CRM-järjestelmässä otettiin esille mahdollisimman paljon, mutta niiden asettamat vaatimukset saatiin selville enimmäkseen jo teknisten ratkaisujen tutkimuksen kautta.

Lopulta vaatimuksia ryhdyttiin kokoamaan yhteen dokumenttiin ja sitä käytiin läpi yhdessä toimeksiantajan osapuolien kanssa, jotta se voitiin todeta riittävän kattavaksi. Dokumentin pohjana käytettiin useista eri lähteistä löytyneitä malleja ja niistä sovellettiin dokumentille sopiva rakenne.

## 6 Tutkimuksen tulokset

Tutkimuksen lopuksi laadittiin moduulin vaatimusmäärittelydokumentti, joka on työssä liitteenä. Tutkimusongelmana työssä oli pääasiallisesti moduulin toiminnallisten ja ei-toiminnallisten vaatimusten selvittäminen. Seuraavissa alaluvuissa dokumenttia on avattu karsitusti.

### 6.1 Yleiskuvaus

Dokumentin yleiskuvauksesta selviää moduulin käyttötarkoitus, käyttäjät ja käyttöympäristö sekä liittymät muihin järjestelmiin. Sen lukemalla lukijalle selviää moduulin tarkoitus pääpiirteittäin.

Moduulin tarkoituksena on kerätä asiakkaiden aktiivisuudesta saatua dataa verkkokaupan asiakastilillä (kulkee nimellä Tietopankki toimeksiantajan verkkokaupassa) ja tallentaa ne väliaikaisesti verkkokaupan tietokantaan. Tämän jälkeen moduuli siirtää siitä saadun datan kontrolloidusti CRM-järjestelmään, jotta se on myyjien ja markkinoijien käytettävissä. Se miten data tallennetaan väliaikaisesti ja siirretään CRM-järjestelmään, riippuu täysin verkkokaupan hallintapaneelin käyttäjien määrittelemistä säännöistä. Datan siirto tapahtuu automaattisesti tietyin aikaväleihin ja säännöt vaikuttavat siirrettävän datan määrään ja muotoon per aktiviteetti.

Aktiviteetilla viitataan toimeksiantajan Tietopankissa esimerkiksi tiettyjen tiedostojen

lataamiseen, videoiden katselemiseen ja artikkeleiden lukemiseen. Eri tiedostot, videot ja artikkelit kuuluvat eri aihealueisiin, joiden perusteella saadaan tietoa asiakkaan kiinnostuksen kohteista ja aktiivisuudesta.

Moduulin käyttöympäristönä toimii pääasiassa verkkokaupan hallintapaneeli ja sen käyttäjinä toimivat verkkokaupan ylläpitäjät, joille sallitaan moduuliin pääsy. Nämä käyttäjät ovat yleensä joko myynnin tai markkinoinnin henkilöstöä. Epäsuorasti moduulin käyttöympäristönä toimii myös CRM-järjestelmä, sillä ilman sen oikeanlaista konfigurointia moduuli ei toimi oikein ja tietoja ei saada tallennettua CRM-järjestelmään.

Moduulin tarvetta perustellaan sillä, ettei nykyisten reaaliaikaisten ratkaisujen ylläpito tai kehittäminen ole kannattavaa. Suunniteltua ja jopa tarvittua datan hajauttamista CRM-järjestelmässä ei voida reaaliaikaisella ratkaisulla järkevästi toteuttaa sen aiheuttamien potentiaalisten ja tiedossa olevien ongelmien takia, jotka aiheutuvat pääasiassa CRM-järjestelmässä olevien asiakkaiden tietojen hakemisen ja tallentamisen viiveistä.

## 6.2 Toiminnalliset vaatimukset

Moduulilla on oltava tiettyjä toimintoja ja ominaisuuksia, joita käyttäjien ja järjestelmän tulee pystyä tekemään, ja ne kuuluvat vaatimusmäärittelyssä toiminnallisiin vaatimuksiin. Toimintoja avattiin vapaassa tekstimuodossa, mutta myös käyttötapauskaavion ja käyttötapauksen avulla, joista näkee myös toimintojen ehdot ja miten mahdolliset virhetilanteet tulisi käsitellä. Toimintoja on oltava verkkokaupan ylläpitäjille, asiakkaille ja taustalla toimivalle järjestelmälle.

Ylläpitäjien on pystyttävä luomaan, muokkaamaan ja poistamaan moduulin sääntöjä hallintapaneelin kautta. Myös aktiviteetista saadun datan tarkastelu tulee olla mahdollista sitä varten luotavasta listanäkymästä, mutta datan muokkaamista ei mahdollisteta. Moduulilla tulee olla myös joitakin yleisiä asetuksia, joita ylläpitäjien täytyy pystyä muuttamaan tarpeen vaatiessa verkkokaupan järjestelmäasetuksien kautta. Järjestelmän tulee pystyä noutamaan asiakkaiden aktiviteetit, moduulin säännöt sekä asetukset ja niitä hyödyntäen siirtämään tiedot automaattisesti CRM-järjestelmään.

Asiakkaiden aktiivisuudesta saatu data täytyy saada tallennettua tiettyjen toimintojen tapahtuessa ja vain jos toiminnolle on asetettu sääntö, joka on aktiivisena. Asiakkaiden toiminnot eivät ole välttämättä täysin asiakkaiden tietoisesti käyttämiä, sillä ne yritetään sulauttaa Tietopankin sisällön selaamiseen niin, ettei käyttäjällä ole aavistustakaan tekemästään toiminnosta.

Toiminnallisiin vaatimuksiin liittyen on esitelty myös toiminnallisesta näkökulmasta eräs tärkein ongelma nykyisten ratkaisujen kannalta: ratkaisujen hallinnointi on hankalaa, hajautunutta ja vaatii ohjelmointitaitoja. Tästä syystä visuaalinen käyttöliittymä onkin eräs moduulin tärkeimmistä vaatimuksista.

Mahdollisiksi lisätoiminnoiksi, joita ei alustavasti yritetä kehittää mukaan moduulin ensimmäisiin versioihin, päätyivät seuraavat toiminnot ja ominaisuudet: manuaalinen tiedonsiirto ja aktiveettien tarkastelu asiakkaiden tiedoista. Manuaalinen tiedonsiirto saattaisi olla kannattava ominaisuus, mikäli automaattinen tiedonsiirto ajastettaisiin tapahtumaan hyvin harvoin ja ylläpitäjille pitäisi saada jokin keino käynnistää tiedonsiirto tarvittaessa. Asiakkaiden aktiviteeteista saatujen tietojen tarkastelu asiakastilien tietojen takaa, eikä vain erillisen listan takaa, ei nähty tärkeänä ominaisuutena ainakaan aluksi.

### 6.3 Ei-toiminnalliset vaatimukset

Ei-toiminnallisia vaatimuksia saatiin kerättyä seuraaviin osa-alueisiin: ohjelmointikielien, kirjastojen ja merkintäkielet, käytettävyys, tietoturva, toimintavarmuus, ylläpidettävyys ja huollettavuus, siirrettävyys, laajennettavuus ja uudelleenkäytettävyys sekä konfiguroitavuus. Nämä vaatimukset määrittelevät enemmän kehitystyössä huomiota otettavia seikkoja, kuin mitä moduulilla pitäisi pystyä tekemään.

Ohjelmointikielten, kirjastojen ja merkintäkielten osalta vaatimukset ovat hyvin selkeitä: on käytettävä niitä, joita Magentossa voidaan käyttää. Käytettävissä olevia ohjelmointikieliä ovat PHP ja JavaScript sekä joissakin tapauksissa JavaScriptiin pohjautuva jQuery-kirjasto. Merkintäkielenä, jota käytetään moduulin konfiguroinnissa, käytetään XML-kieltä. Näin moduuli saadaan pysymään Magenton standardien mukaisena ja pystytään hyödyntämään Magenton valmiiksi tarjoamia ominaisuuksia muun muassa tiedon haussa, tallentamisessa ja poistamisessa. Myös käytettävyydeltään

moduuli rakentuu hyvin pitkälti Magenton hallintapaneelin käyttöliittymän mukaisesti sen näkymien ja virheilmoitusten suhteen.

Tietoturvan kannalta moduulissa tulee ottaa huomioon riittävät suojaukset tietojen muokkauksen ja muun hallinnoinnin kannalta. Magento pitää tästä huolen hallintapaneelin kautta muokattavien tietojen osalta, mutta asiakkaiden suhteen eri toiminnolle täytyy tehdä erillisiä varmistuksia. Kun järjestelmä siirtää tietoja automaattisesti, sen tulee hyödyntää SugarCRM-järjestelmän tarjoamaa REST-rajapintaa, jossa hyödynnetään sen omia suojauksia tiedon haussa ja siirrossa.

Yleisesti ottaen toimeksiantajan verkkokauppaympäristö on hyvin toimintavarma ja moduulin tulee toimia vähintään yhtä varmasti. Joitakin poikkeustapauksia voi olla, kuten ympäristön päivitykset tai samanaikaisten käyttäjien (asiakkaiden) yllättävä kasvu hetkellisesti.

Ylläpidettävyyden ja huollettavuuden suhteen moduulin täytyy olla niin paljon Magenton standardien mukaisesti kehitetty kuin mahdollista. Tämä varmistaa sen, että myös ulkoiset kehittäjät pystyvät tekemään moduuliin muutoksia ja päivityksiä.

Koska moduulia kehitetään käytettäväksi toimeksiantajan olemassa oleviin prosesseihin ja järjestelmiin, ei sitä yritetä kehittää niin, että se olisi suoraan käytettävissä joskin toisessa Magento-ympäristössä. Moduulin laajennettavuus täytyy kuitenkin säilyä, sillä toimeksiantajalla saattaa tulla tarpeita lisätä analytiikan määrää tai muuttaa sitä.

Pääasiassa konfiguroitavuudella tarkoitetaan kehitettävän moduulin kannalta sen yleisiä asetuksia, jotka vaikuttavat koko moduulin toimintaan. Näitä ovat esimerkiksi ajastetun siirron kytkentä päälle tai pois, käsiteltävien tietojen määrä per ajo ja asiakasryhmät joiden aktiviteetit tallennetaan. Nämä asetukset määrittellään Magenton järjestelmäasetuksien taakse tulevan moduulikohtaisen näkymän takaa.

## 6.4 Muut vaatimukset

Näitä vaatimuksia ei voida välttämättä suoraan asettaa toiminnallisiin tai ei-toiminnallisiin vaatimuksiin. Ne kuitenkin tarkentavat joitakin vaatimuksia ja ovat siksi dokumentoituna. Muiksi vaatimuksiksi saatiin moduulin suorituskyky, rajapinnat ja käyttöliittymät. Suorituskyvyn vaatimukset tarkentavat jonkin verran käytettävyyden

ja toimintavarmuuden vaatimuksia asettamalla sille tietyt ehdot esimerkiksi latausaikoihin. Samaten rajapintojen vaatimukset tarkentavat tietoturvan vaatimuksia ja käyttöliittymän vaatimukset käytettävyyden vaatimuksia.

Käyttöliittymiä avattiin enimmäkseen visuaalisesti käyttäen kuvia käyttöliittymäkartasta ja käyttöliittymäsivuista. Vaikka ylläpitäjät eivät sitä normaalisti käyttäisikään, niin myös palvelimelta löytyvän komentorivipohjaisen käyttöliittymän kautta ajettava moduulin skriptiltä, jota ajetaan normaalisti automaattisesti, tulee löytyä ohjeet sen käyttöä varten, jotta sitä pystytään ajamaan myös manuaalisesti.

## 6.5 Rajoitukset

Rajoitukset tarkentavat ei-toiminnallisia vaatimuksia entisestään standardien ja ohjelmistorajoitusten kautta. Standardeissa tarkennetaan millä tavoin mitäkin ohjelmointikieltä saatetaan hyödyntää, missä muodossa dataa tulisi käsitellä ja millaisia yhteyksiä käyttäen kyseistä dataa tulee siirtää järjestelmästä toiseen. Ohjelmistorajoituksissa on selkeyden vuoksi listattuna ohjelmointikielten, kirjastojen ja merkintäkielten lisäksi myös käytettävät käyttöjärjestelmät ja käyttöliittymät.

## 6.6 Tiedot ja tietokannat

Hieman ehkä tavallisesta vaatimusmäärittelystä poiketen dokumenttiin otettiin määriteltäväksi myös moduulin tiedot ja tietokannat. Tietokanta määrittyy ympäristön mukaisesti, mutta talletettavien tietojen, ja etenkin moduulille määritettävien sääntöjen, määrittelyssä täytyi ottaa huomioon kaikki nykyisistä ratkaisuista saadut vaatimukset. Sääntöjen tietorakenne muuttui useaan otteeseen ennen kuin se todettiin riittäväksi ja saattaa muuttua myös kehitystyön aikana.

## 7 Pohdinta

Tutkimuksen alkuperäinen aihe muuttui moduulin kehittämisestä suunnitteluun ja myöhemmin vaatimusten määrittelyyn, kun alkoi selvitä, että moduulilla onkin enemmän vaatimuksia kuin oli alun perin osattu kuvitella yhdessä toimeksiantajan

kanssa. Tästä syystä haluttiin varmistaa, että kaikki tarvittavat ominaisuudet ja vaatimukset saadaan selville ennen varsinaisen kehitystyön aloittamista. Näin moduulista saadaan varmemmin sellainen kuin vaaditaan eikä siihen tarvitse tehdä heti jälkeensä suuria muutoksia.

Vaikka Magenton tekninen osaamiseni oli jo muutaman vuoden työsuhteen aikana kehittynyt huomattavasti, antoi tutkimustyö lisää aikaa sen kartuttamiseen ja mahdollisten ratkaisujen selvittämiseen. Näin vaatimuksista saatiin sellaisia, että ne ovat myös toteutettavissa ilman ulkoistetun kehitystyön tarvetta. Tutkimustyön aikana kertynyt osaaminen avasikin monia mahdollisuuksia Magenton kehitystyöhön liittyen, eikä ainoastaan tutkimustyön kohteena olleen moduulin osalta.

Eräs selkeä havainto oli esillä lähes koko tutkimustyön ajan: moduulin suunnittelu ja kehitystyö olisi pitänyt toteuttaa jo silloin kun analytiikan määrä oli vähäisempää. Näin moduulista olisi voitu kehittää ensin kevyt versio ja laajentaa sitä tarpeen mukaan, eikä moduulin suunnittelu olisi ollut yhtä hankalaa. Tähän liittyy kuitenkin muutamia ongelmia. Tuolloin itselläni ei ollut riittävää Magenton teknistä osaamista moduulin suunnitteluun eikä etenkään sen kehittämiseen. Myöskään nykyisen analytiikan määrää ei osannut kukaan toimeksiantajan henkilöstöstä ennustaa, joten moduulin tarvetta ei osattu arvioida.

Vaatimusmäärittely tuntui prosessina jokseenkin turhan laajalta, ottaen huomioon sen, että kohteena ei ollut järjestelmä vaan yksi järjestelmän osa. Vaatimusmäärittely on varmasti kannattavaa, kun kyseessä on erittäin suuri kokonaisuus, mutta yksittäisen moduulin kohdalla se saattaa vaikuttaa ylimääräiseltä työltä. Olen kuitenkin vakuuttunut, että vaatimusmäärittely oli järkevää, sillä moni asia moduulin toiminnallisiin ja ei-toiminnallisiin vaatimuksiin liittyen ehti muuttua ja tarkentua sen aikana. Mutta, kuten monesti kehitystyön aikana, nämä vaatimukset voivat edelleen tarkentua tai muuttua täysin.



## Lähteet

Alexander, I. F. & Stevens, R. 2002. Writing Better Requirements. Harlow: Pearson Education Ltd.

How to Decide? Magento Enterprise vs. Community. N.d. Artikkelel Magentoil ilmaisen ja maksullisen version eroista. Viitattu 28.3.2017. <http://www.customerparadigm.com/magento-enterprise-vs-magento-community/>

Kananen, J. 2012. Kehittämistutkimus opinnäytetyönä. Kehittämistutkimuksen kirjoittamisen käytännön opas. Jyväskylä: Jyväskylän ammattikorkeakoulu.

MacGregor, A. 2015. Magento PHP Developer's Guide, Second Edition. Birmingham: Packt Publishing Ltd.

Morrioni, L. 2015. The History of Magento and Its Rise to Ubiquity. Viitattu 28.3.2017. <https://morrioni.com/blog/history-of-magento/>

Multiple Deployment Options. N.d. Esittely SugarCRM-järjestelmän julkaisuvaihtoehtoista. Viitattu 28.3.2017. <https://www.sugarcrm.com/feature/multiple-deployment-options>

Paakki, J. 2011. Ohjelmistojen vaatimusmäärittely, diat. Viitattu 6.4.2017. <https://www.cs.helsinki.fi/u/paakki/Vaatimus-11-Luentokalvot-1.pdf>

Podila, P. 2013. HTTP: The Protocol Every Web Developer Must Know - Part 1. Viitattu 28.3.2017. <https://code.tutsplus.com/tutorials/http-the-protocol-every-web-developer-must-know-part-1--net-31177>

Reitsma, J. 2011. When use the community pool and when the local pool? Viitattu 28.3.2017. <https://www.yireo.com/blog/1412-when-use-the-community-pool-and-when-the-local-pool>

Sahlsten, P. 2012. Asiakkuudenhallinta eli CRM – mistä oikein on kysymys? Viitattu 28.3.2017. <http://www.myynti20.fi/asiakkuudenhallinta-crm-mista-on-kysymys/>

Soni, S. 2015. Understand Overriding in Magento: Models. Viitattu 28.3.2017. <https://code.tutsplus.com/tutorials/understand-overriding-in-magento-models--cms-23354>

Storm, A. 2009a. Magento Controller Dispatch and Hello World. Viitattu 28.3.2017. [http://alanstorm.com/magento\\_controller\\_hello\\_world/](http://alanstorm.com/magento_controller_hello_world/)

Storm, A. 2009b. The Magento Config. Viitattu 28.3.2017. [http://alanstorm.com/magento\\_config/](http://alanstorm.com/magento_config/)

Storm, A. 2010a. Custom Magento System Configuration. Viitattu 29.3.2017. [http://alanstorm.com/custom\\_magento\\_system\\_configuration/](http://alanstorm.com/custom_magento_system_configuration/)

Storm, A. 2010b. Magento System Overrides and Upgradability. Viitattu 28.3.2017. [http://alanstorm.com/magento\\_upgrade\\_rewrite\\_override/](http://alanstorm.com/magento_upgrade_rewrite_override/)

Storm, A. Magento for Developers: Part 1—Introduction to Magento. Viitattu 28.3.2017. <http://devdocs.magento.com/guides/m1x/magefordev/mage-for-dev-1.html>

Sugar Developer Guide 7.7 - Architecture. N.d. Dokumentaatio SugarCRM-järjestelmän arkkitehtuurista. Viitattu 28.3.2017. [http://support.sugarcrm.com/Documentation/Sugar\\_Developer/Sugar\\_Developer\\_Guide\\_7.7/Architecture/](http://support.sugarcrm.com/Documentation/Sugar_Developer/Sugar_Developer_Guide_7.7/Architecture/)

Vaqqas, M. 2014. RESTful Web Services: A Tutorial. Viitattu 28.3.2017. <http://www.drdoobs.com/web-development/restful-web-services-a-tutorial/240169069>

Wieggers, K. E. 2003. Software Requirements, Second Edition. Redmond: Microsoft Press.

## Liitteet

Liite 1. Moduulin vaatimusmäärittelydokumentti



VAATIMUSMÄÄRITTELY

# Trainer-Databank (Analytics) moduulin vaatimusmäärittely

Laatija(t): Mika Majala

Päiväys: 2.5.2017

Hyväksynyt: Teemu Karjaluoto

Versio: 1.0.0

## Sisältö

<b>1 Muutoshistoria</b>	<b>3</b>
<b>2 Johdanto</b>	<b>3</b>
<b>3 Määritelmät ja termit</b>	<b>3</b>
<b>4 Yleiskuvas</b>	<b>4</b>
4.1 Käyttötarkoitus	4
4.2 Käyttäjät ja toimintaympäristö	4
4.3 Liittymät muihin järjestelmiin	4
<b>5 Toiminnalliset vaatimukset</b>	<b>4</b>
5.1 Yleiskuvas moduulin toiminnasta	5
5.2 Käyttäjien kertomat ongelmat	5
5.3 Käyttötapauskaavio	5
5.4 Käyttötapaukset	6
5.5 Mahdolliset lisätoiminnot	7
5.6 Ei-toteutettavat toiminnot	8
<b>6 Ei-toiminnalliset vaatimukset</b>	<b>8</b>
6.1 Ohjelmointikielet, kirjastot ja merkintäkielet	8
6.2 Käytettävyys	8
6.3 Tietoturva	8
6.4 Toimintavarmuus	9
6.5 Ylläpidettävyys ja huollettavuus	9
6.6 Siirrettävyys, laajennettavuus ja uudelleenkäytettävyys	9
6.7 Konfiguroitavuus	9
<b>7 Muut vaatimukset</b>	<b>10</b>
7.1 Suorituskyky	10
7.2 Rajapinnat	10
7.3 Käyttöliittymät	10
7.3.1 Käyttöliittymäkarta	10
7.3.2 Käyttöliittymäsivut	11
7.3.3 Komentorivipohjainen käyttöliittymä	14

---

<b>8 Rajoitukset</b>	<b>14</b>
8.1 Standardit	14
8.2 Ohjelmistorajoitukset	14
<b>9 Tiedot ja tietokannat</b>	<b>14</b>
9.1 Tietokannat	14
9.2 Talletettavat tiedot	15

## 1 Muutoshistoria

Henkilö	Päiväys	Versio	Kommentti
Mika Majala	24.2.2017	0.0.1	Dokumentti luotu, johdantoa, termejä, kuvausta ja pääkohtien selostusta aloitettu
Mika Majala	27.2.2017	0.0.2	Ei-toiminnallisia vaatimuksia aloitettu
Mika Majala	1.3.2017	0.0.3	Ei-toiminnallisia vaatimuksia hiottu ja viimeistelty
Mika Majala	3.3.2017	0.0.4	Toiminnallisia vaatimuksia aloitettu, tietokannan tauluja aloitettu, käyttötapauskaavion ensimmäinen versio
Mika Majala	7.3.2017	0.0.5	Muita vaatimuksia aloitettu, toiminnallisia vaatimuksia täydennetty
Mika Majala	8.3.2017	0.1.0	Tarkennuksia katselmuksen aikana, keskeneräisen version hyväksyntä
Mika Majala	9.3.2017	0.1.1	Käyttöliittymäkuviauksia
Mika Majala	13.3.2017	0.1.2	Käyttöliittymäkuvausten hiomista, käyttötapauskaavion uusimista ja käyttötapausten kirjoittelua. Talletettävien tietojen täydentämistä. Yleistä hiomista
Mika Majala	12.4.2017	0.1.3	Säännöille tallennettävien tietojen yhtenäistämistä
Mika Majala	28.4.2017	1.0.0	Alustava viimeinen versio
Teemu Karjaluoto	2.5.2017	1.0.0	Hyväksytty versio

## 2 Johdanto

Tämän dokumentin tarkoituksena on kuvata Magento-järjestelmään kehitettävän CRM analytiikkaan liittyvän moduulin pääpiirteet ja tärkeimmät toiminnot vaatimusmäärittelydokumentin muodossa. Dokumentti on tarkoitettu pääasiassa Trainer4You Oy:n sisäiseen käyttöön, mutta liitetään liitteenä osaksi laatijan (Mika Majala) opinnäytetyötä.

## 3 Määritelmät ja termit

Alla on listattu ja kuvattu tämän vaatimusmäärittelydokumentin ymmärtämisen kannalta tärkeimmät määritelmät ja termit.

Termi	Kuvaus
Magento	Verkkokauppa-alusta, joka on käytössä Trainer4You Oy:n verkkokaupassa
SugarCRM	Asiakkuudenhallintajärjestelmä, joka on käytössä Trainer4You Oy:llä
Moduuli	Sovellukseen / järjestelmään kehitettävä / liitettävä laajennus, tässä dokumentissa kohteena Magento-järjestelmä
Rajapinta	Ohjelmointirajapinta (API) on joukko määritelmiä / protokollia / työkaluja, joiden avulla järjestelmät keskustelevat keskenään esimerkiksi siirtääkseen tai hakeakseen dataa toistensa välillä
Funktio / metodi (eli aliohjelma)	Aliohjelma on ohjelmoinnissa itsenäinen ohjelman osa, joka suorittaa tietyn toiminnon ja jota voidaan kutsua eri puolilta pääohjelmaa tai muista aliohjelmissa
Tietopankki	Trainer4You verkkokaupan asiakastili

## 4 Yleiskuvaus

Moduuli kehitetään osaksi Trainer4You Oy:n verkkokauppaa, jossa on käytössä Magento-järjestelmä. Moduulin tarkoituksena on siirtää asiakkaiden aktiivisuudesta saadut tiedot verkkokaupasta SugarCRM-asiakkuudenhallintajärjestelmään.

Ulkoisesti moduuli kulkee nimellä Trainer/Databank (tai Databank), mutta yrityksen sisällä voidaan myös puhua Analytics-moduulista, sillä analytiikka on moduulin pääasiallinen tarkoitus. Databank nimitys tulee termistä Tietopankki.

### 4.1 Käyttötarkoitus

Moduulin tarkoituksena on tallentaa asiakkaiden aktiivisuutta Tietopankissa verkkokaupan tietokantaan väliaikaisesti ja siirtää tiedot SugarCRM-järjestelmään aika-ajoin tiettyjen ennalta määriteltävissä olevien sääntöjen mukaisesti. Nämä säännöt määritellään Magenton hallintapaneeliin kehitettävien näkymien kautta.

Aktiivisuuden väliaikainen tallennus ja ajoitettu siirtäminen tehdään siksi, että asiakkaiden perustiedot ja analytiikkaan liittyvät tiedot saadaan hajautettua SugarCRM-järjestelmässä eri moduuleihin. Näin päästään eroon kyseisen järjestelmän asettamista rajoituksista, jotka ovat olleet useaan otteeseen hidasteena analytiikan parantamisessa. Suurimpana rajoitteena on kenttien rajoitettu määrä per moduuli (kuten Kontaktit), johon liittyy myös vaara tietojen häviämisestä. Ajoitettu tietojen siirtäminen täytyy tehdä, jotta aktiivisuudesta saatujen tietojen oikeellisuus säilyy tietojen hajauttamisen jälkeenkin. Nykyinen, reaaliaikainen tietojen siirtäminen on altis virheille kun tiedot hajautetaan, mikäli asiakas liikkuu liian nopeasti Tietopankissa.

### 4.2 Käyttäjät ja toimintaympäristö

Käyttöympäristönä toimii pääsääntöisesti Magento-järjestelmän hallintapaneeli. Toissijaisesti myös SugarCRM-järjestelmään täytyy tehdä valmisteluita ja muutoksia kaikkien toimintojen oikeanlaisen toiminnon ja tietojen säilymisen takaamiseksi, mutta näitä valmisteluita ei kuitenkaan tämän dokumentin aikana tarkenneta.

Käyttäjinä toimivat Trainer4You Oy:n myynnin ja markkinoinnin henkilöstö, joilla on pääsy Magento-järjestelmän hallintapaneeliin sekä tarve pystyä muokkaamaan siellä määriteltävissä olevia sääntöjä. Moduulin käyttötarkoitus ja toimintatapa tulee olla tuttuja ennen käyttöä.

### 4.3 Liittymät muihin järjestelmiin

Moduuli on yhteydessä SugarCRM-järjestelmään sen tarjoaman REST-rajapinnan (API) kautta. Rajapintaa hyödynnetään toisen Magento-järjestelmään kehitetyn moduulin, Trainer/SugarCRM, avulla. Kyseinen moduuli tarjoaa tarvittavat ominaisuudet rajapinnan käyttöön ja se on tarkoitettu ottaa käyttöön muissakin rajapintaa tarvitsevilla moduuleilla sekä käyttötapaüksissa yhteneväisyyden saavuttamiseksi.

## 5 Toiminnalliset vaatimukset

Toiminnalliset vaatimukset tarkoittavat toimintoja ja ominaisuuksia, joita järjestelmässä tulee olla. Nämä ovat siis varsinaiseen käyttöön liittyviä vaatimuksia ja vaikuttavat järjestelmän rakenteeseen.

### 5.1 Yleiskuvaus moduulin toiminnasta

Magenton ylläpitäjä voi asettaa moduulille yleiset asetukset, joiden perusteella moduuli toimii.

Magenton ylläpitäjä voi lisätä analytiikan sääntöjä haluamallaan tiedoilla. Kun sääntö on tallennettu, voidaan sitä alkaa käyttämään Tietopankissa.

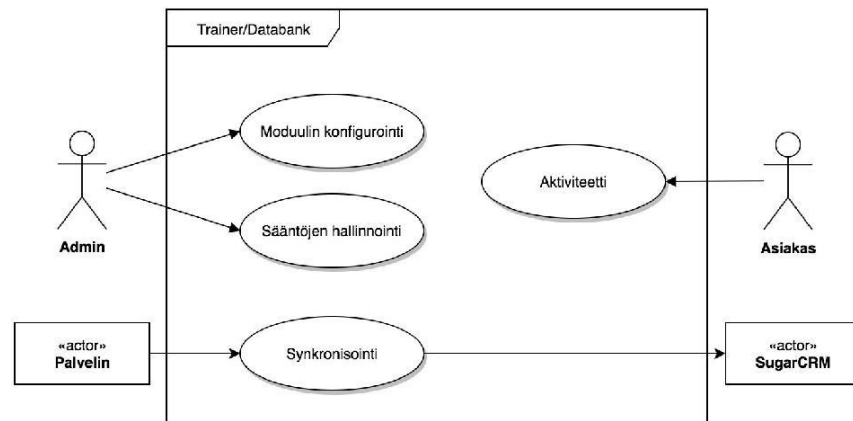
Sääntöjä käytetään esimerkiksi tarjoamalla asiakkaille sääntökohtainen linkki (kuten pikaoppailla), jota klikkaamalla kyseinen aktiviteetti tallentuu tietokantaan ja asiakas ohjautuu eteenpäin. Esimerkiksi videoiden ja artikkeleiden aktiivisuus tallentuu eri tavoin; tarvittavilla sivuilla on JavaScript/Query skripti, joka tallentaa aktiivisuuden tietyin ehdoin (klikkaus tai muu).

Palvelimella pyörii ajastettu skripti, joka tarkastelee aika-ajoin (esimerkiksi 10min välein) onko Tietopankista syntynyt uutta aktiviteettia. Skripti käy määritettävissä olevan määrän (esimerkiksi 50) aktiviteetteja läpi säännöissä määritettyjen asetusten mukaisesti. Skripti tarkistaa jokaisen aktiviteetin kohdalla SugarCRM-järjestelmästä asiakkaan sekä asiakkaan analytiikan tiedot ja päivittää tai luo niitä tarvittaessa.

### 5.2 Käyttäjien kertomat ongelmat

Nykyisin mitään hallinnointia analytiikan kannalta ei voi tehdä kuin koodin kautta ja sekin on hyvin hajautunutta. Tämä ei ole optimaalista käyttäjien näkökulmasta eikä oikeastaan mahdollistakaan, ellei ole ohjelmointitaitoinen.

### 5.3 Käyttötapauskaavio





## 5.4 Käyttötapaukset

<b>KÄYTTÖTAPAUSET:</b>	Moduulin konfigurointi
<b>YHTEENVETO:</b>	Ylläpitäjä muuttaa moduulin yleisiä asetuksia.
<b>TOIMIJA:</b>	Ylläpitäjä
<b>EHDOT:</b>	Käyttäjä on kirjautunut Magenton hallintapaneeliin ylläpitäjänä ja hänellä on oikeus muokata moduulin asetuksia.
<b>KUVAUS:</b>	Käyttäjä siirtyy Magenton hallintapaneelissa järjestelmän asetuksiin ja sieltä moduulin asetuksiin. Käyttäjä muuttaa siellä haluamiaan asetuksia.
<b>POIKKEUKSET:</b>	Jokin/jotkin tallennettavista tiedoista ei ole oikeanlaisessa muodossa: järjestelmä ilmoittaa mitkä kentät on tarkistettava.
<b>LOPPUTULOS:</b>	Asetukset tallentuvat järjestelmään.

<b>KÄYTTÖTAPAUSET:</b>	Säännön luominen
<b>YHTEENVETO:</b>	Ylläpitäjä luo säännön moduuliin.
<b>TOIMIJA:</b>	Ylläpitäjä
<b>EHDOT:</b>	Käyttäjä on kirjautunut Magenton hallintapaneeliin ylläpitäjänä ja hänellä on moduulin käyttöoikeus.
<b>KUVAUS:</b>	Käyttäjä siirtyy moduulin sääntöjen listanäkymään ja siirtyy sieltä luomaan uutta sääntöä.
<b>POIKKEUKSET:</b>	Jokin/jotkin tallennettavista tiedoista ei ole oikeanlaisessa muodossa: järjestelmä ilmoittaa mitkä kentät on tarkistettava.
<b>LOPPUTULOS:</b>	Sääntö tallentuu järjestelmään.

<b>KÄYTTÖTAPAUSET:</b>	Säännön muokkaaminen
<b>YHTEENVETO:</b>	Ylläpitäjä muokkaa sääntöä moduulissa.
<b>TOIMIJA:</b>	Ylläpitäjä
<b>EHDOT:</b>	Käyttäjä on kirjautunut Magenton hallintapaneeliin ylläpitäjänä ja hänellä on moduulin käyttöoikeus. Moduuliin on aiemmin luotu sääntöjä.
<b>KUVAUS:</b>	Käyttäjä siirtyy moduulin sääntöjen listanäkymään ja siirtyy muokkaamaan sääntöä valitsemalla sen listasta.
<b>POIKKEUKSET:</b>	Jokin/jotkin tallennettavista tiedoista ei ole oikeanlaisessa muodossa: järjestelmä ilmoittaa mitkä kentät on tarkistettava.
<b>LOPPUTULOS:</b>	Sääntö tallentuu järjestelmään.

<b>KÄYTTÖTAPAUSET:</b>	Säännön poistaminen
<b>YHTEENVETO:</b>	Ylläpitäjä poistaa säännön moduulista.
<b>TOIMIJA:</b>	Ylläpitäjä
<b>EHDOT:</b>	Käyttäjä on kirjautunut Magenton hallintapaneeliin ylläpitäjänä ja hänellä on moduulin käyttöoikeus. Moduuliin on aiemmin luotu sääntöjä.

<b>KUVAUS:</b>	Käyttäjä siirtyy moduulin sääntöjen listanäkymään ja valitsee listasta säännön tai säännöt jotka haluaa poistaa. Valittuaan haluamansa säännöt, käyttäjä valitsee listanäkymän toiminnoista poiston.
<b>POIKKEUKSET:</b>	Sääntöä ei enää ole järjestelmässä (jos esimerkiksi toinen käyttäjä on ehtinyt poistamaan sen); järjestelmä näyttää virheen käyttäjälle.
<b>LOPPUTULOS:</b>	Sääntö poistuu järjestelmästä.

<b>KÄYTTÖTAPAUS:</b>	Aktiviteetin tallentuminen
<b>YHTEENVETO:</b>	Asiakkaan aktiviteetti tallentuu moduuliin.
<b>TOIMIJAT:</b>	Asiakas
<b>EHDOT:</b>	Käyttäjä on kirjautunut Tietopankkiin. Käyttäjän aktiviteetin mukainen sääntö on luotu hallintapaneelin kautta ja on käytössä.
<b>KUVAUS:</b>	Käyttäjä kirjautuu sisään Tietopankkiin. Käyttäjä lataa pikaoppaan, katsoo videon, lukee artikkelisarjaa tai tekee muuta seurattavaa aktiviteettia tietopankissa.
<b>POIKKEUKSET:</b>	Käyttäjän istunto on vanhentunut sivun avaamisen ja aktiviteetin välillä: joko käyttäjä ohjataan uudelleen kirjautumissivun kautta tai aktiviteettia ei yksinkertaisesti tallenneta (riippuu aktiviteetista).
<b>LOPPUTULOS:</b>	Aktiviteetti tallentuu järjestelmään.

<b>KÄYTTÖTAPAUS:</b>	Aktiviteetin synkronointi
<b>YHTEENVETO:</b>	Aktiviteetti synkronoidaan moduulista SugarCRM-järjestelmään.
<b>TOIMIJAT:</b>	Palvelin
<b>EHDOT:</b>	Synkronisointi on asetettu päälle moduulin asetuksissa. Tietopankissa on syntynyt asiakkaiden aktiviteettia. SugarCRM rajapinnan käyttöä varten saadaan OAuth2-token. SugarCRM-järjestelmään on luotu tarvittavat kentät aktiviteetin tallentamista varten (vaihtoehtoinen, mutta vaadittu oikeaa toimintaa varten).
<b>KUVAUS:</b>	Palvelimella cron-ajastettu skripti käynnistyy ja käyttää moduulin luokkia aktiviteettien lataamiseen, käsittelyyn ja synkronointiin. Aktiviteetit käsitellään niiden luomisjärjestyksessä.
<b>POIKKEUKSET:</b>	SugarCRM rajapinta lakkaa vastaamasta (päivitys tai muu) tai OAuth2-token invalidoituu: synkronointi keskeytetään ja aktiviteetit yritetään synkronoida seuraavan ajon aikana.
<b>LOPPUTULOS:</b>	Aktiviteetit synkronoituvat SugarCRM-järjestelmään ja aktiviteetit merkitään käsitellyiksi.

## 5.5 Mahdolliset lisätoiminnot

Manuaalista tiedonsiirron aktivoinnin mahdollisuutta ei yritetä toteuttaa ensisijaisesti, mutta saattaa olla mahdollinen lisäominaisuus. Haasteena ajoitetun skriptin ja manuaalisen ajon mahdollinen päällekkäisyys ja siitä aiheutuvat mahdolliset ongelmat.

Aktiviteeteillä ei ole tietokantatasolla relaatiota asiakkaisiin eli eivät ole varsinaisesti liitoksissa asiakkaisiin eivätkä ole siksi myöskään tarkasteltavissa asiakkaan tietojen takaa. Tämä saattaa muuttua myöhemmin, mikäli nähdään tarpeelliseksi tai hyödylliseksi.

### 5.6 Ei-toteutettavat toiminnot

Asiakkaiden aktiviteetista saatujen tietojen muokkaamista ei mahdollisteta. Ne ovat vain esikatseltavissa ja poistettavissa.

## 6 Ei-toiminnalliset vaatimukset

Ei-toiminnalliset vaatimukset tarkoittavat esimerkiksi sitä, millä tavoin järjestelmän tulee toimia ja mitä tekniikoita (kuten ohjelmointikieliä) siinä tulee käyttää. Nämä vaatimukset ovat tärkeitä varsinaisen kehitystyön kannalta.

### 6.1 Ohjelmointikieliet, kirjastot ja merkintäkielet

Moduulin ohjelmointikielinä käytetään Magentossa toimivia kieliä, kuten PHP ja tarvittaessa JavaScript/Query (käyttöliittymässä). XML-merkintäkieltä käytetään moduulin konfiguroinnissa. Tietojen haussa, tallentamisessa ja poistamisessa tulee käyttää Magenton tarjoamia metodeita niin paljon kuin mahdollista.

### 6.2 Käytettävyys

Käyttöliittymältään moduuli määrittyy Magenton hallintapaneelin käyttöliittymän mukaisesti ja tulee vastata käytettävyydeltään muita vastaavanlaisia moduuleita näkymien osalta. Näitä näkymiä ovat lista- ja lomakenäymät sekä järjestelmäasetusnäkyvä.

Mikäli mahdollista, eri näkymissä tulee näkyä vain oleellimmat tiedot ja kentät. Lomakenäkymissä kenttien yhteydessä tulee olla riittävät kuvaukset kenttien käyttötarkoituksista käytön helpottamiseksi.

Virhetilanteissa moduulin näkymissä tulee näkyä virheen syy ja mahdolliset jatkotoimenpiteet virheen poistamiseksi, mikäli mahdollista. Myös onnistuneet tietojen syötöt tulee ilmoittaa käyttäjälle. Ilmoitusten tulee vastata Magenton ilmoituksia. Virheet eivät saa aiheuttaa muiden tietojen menetystä tai muuttumista.

### 6.3 Tietoturva

Moduulin tulee käyttää tietoturvan osalta Magenton ja SugarCRM:n tarjoamia suojauksia.

Moduulin asetusten ja sääntöjen määrittely tulee onnistua vain Magenton hallintapaneelin kautta. Asiakkaiden aktiviteetin tulee tallentua vain kirjautuneille käyttäjille ja vain sellaisilla säännöillä, jotka on hallintapaneelin kautta asetettu käyttöön. Muissa tapauksissa moduulin tulee hylätä sille annetut syötteet ja keskeyttää niiden käsittely tai tallennus.

Ajastettu tietojen siirtäminen SugarCRM-järjestelmään tulee tapahtua sen tarjoaman REST-rajapinnan suojausten mukaisesti käyttäen OAuth2-tokeneita. Tästä pitää enimmäkseen huolen Trainer/SugarCRM-moduuli, mutta tokeneiden käyttö täytyy ottaa huomioon myös tässä moduulissa.

## 6.4 Toimintavarmuus

Moduulin tulee toimia luotettavasti kaikkina aikoina, jolloin Magento-järjestelmään ei olla tekemässä muutoksia tai päivityksiä. Mikäli luotettavuus kärsii muina aikoina, tulee ongelmakohtat selvittää ja korjata mikäli mahdollista. Näitä tilanteita varten moduulin täytyy kirjoittaa virhelogia, jotta ongelmien ratkominen helpottuu.

Isoimmat ongelmat saattavat ilmetä asiakkaiden aktiivisuuden tallentamisessa, riippuen käyttäjämäärästä. Mikäli käyttäjämäärä kasvaa hyvin suureksi lyhyellä aikavälillä, saattaa joiakin tietoja jäädä tallentumatta. Näin ei saisi kuitenkaan käydä normaalin aktiivisuuden aikana.

Asetusten ja sääntöjen tallentamisessa ei pitäisi syntyä ongelmia vähäisen käyttäjämäärän takia.

## 6.5 Ylläpidettävyys ja huollettavuus

Moduulin tulee olla niin selkeä kuin mahdollista ja Magenton standardien mukaisesti rakennettu, jotta se on ylläpidettävissä ja huollettavissa ulkopuolisellakin kehittäjällä. Moduulin muutokset, lisäykset ja päivitykset pitää siis olla mahdollisia myös moduulin käyttöönoton ja alun ylläpitovaiheen jälkeen ilman suurempia katkoksia.

## 6.6 Siirrettävyys, laajennettavuus ja uudelleenkäytettävyys

Pääasiassa moduulia ei ole tarkoitettu siirrettäväksi toiseen ympäristöön. Moduuli saattaa olla tietyiltä osin liian tiukasti kiinni olemassa oleviin prosesseihin, jotta se voisi olla siirrettävissä helposti mihin tahansa Magento ympäristöön.

Moduulin tulee olla laajennettavissa mahdollisten analytiikan lisäyksien takia. Moduulin eri osien tulee käyttää mahdollisimman paljon samaa rakennetta/logiikkaa ja samoja tiedonhaussa, -käsittelyssä ja -siirrossa käytettäviä metodeja.

## 6.7 Konfiguroitavuus

Vähintään moduulin ajastettu tiedonsiirto tulee olla kytkettävissä pois päältä tarvittaessa ja ajastettujen skriptien tulee kunnioittaa asetusta. Moduulilla tulisi olla myös muita yleisiä koko moduulin toimintaan vaikuttavia asetuksia. Yleisten asetusten tulee olla asetettavissa Magenton hallintapaneelin Järjestelmäasetuksissa ja noudettavissa käyttöä varten Magenton konfiguraatiosta. Asetuksilla tulee olla oletusarvot moduulin XML-konfiguraatiossa ja muokatut arvot tallentuvat tietokantaan, joiden tallentamisesta Magento pitää huolen.

Alla on listattu tarpeelliset yleisesti konfiguroitavat asetukset:

Kenttä	Kentän tyyppi	Esimerkkiarvot
Ajastettu siirto	Pudotusvalikko	Päällä / Pois päältä
Käsiteltävien tietojen määrä per ajo	Numerokenttä	50
SugarCRM Analytics moduulin nimi	Tekstikenttä	t4ua_Analytics
SugarCRM Contacts-Analytics linkityksen nimi	Tekstikenttä	t4ua_analytics_contacts
Uudelleenlatauksen aikaraja (päivinä)	Numerokenttä	30
Asiakasryhmät joiden aktiviteetit tallennetaan	Monivalinta	Prospekti, Customer jne.

Sääntöjen konfiguroitavuus kuvataan toiminnallisissa ja muissa vaatimuksissa, sillä ne ovat määriteltävissä yksittäin ja vaikuttavat yksittäisiin sääntöihin eivätkä koko moduuliin.

## 7 Muut vaatimukset

### 7.1 Suorituskyky

Latausajat tulevat paranemaan pikaoppaiden osalta huomattavasti, sillä tiedot tallennetaan paikallisesti ja käsitellään tarkemmin myöhemmin SugarCRM-järjestelmään siirrettäväksi. Muutoin asiakkaille näkyvää muutosta ei tule.

Moduulin näkymien suorituskyky täytyy vastata vähintään muita näkymiä Magenton hallintapaneelissa.

Taustalla pyörivä skripti ei saa olla niin raskas, että Magento tai SugarCRM-järjestelmissä tapahtuu huomattavia hidasteluita.

### 7.2 Rajapinnat

Magento ja SugarCRM-järjestelmien välinen liitäntä kehitetään SugarCRM:n tarjoaman REST-rajapinnan ympärille.

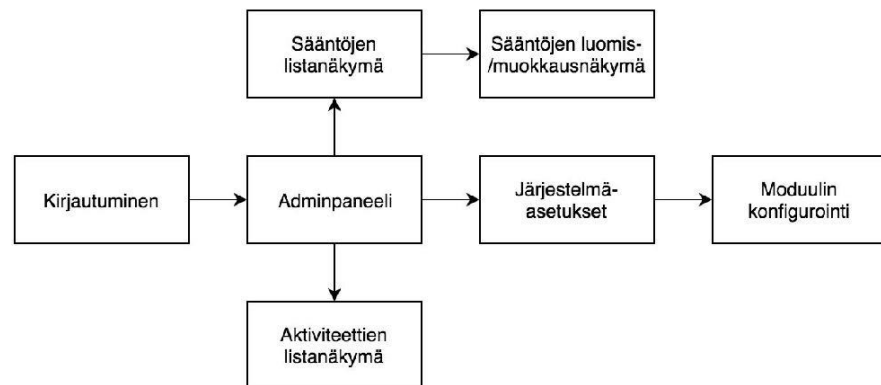
REST-rajapintaa hyödynnetään [https://\\*\\*\\*\\*.sugaropencloud.eu/rest/v10/](https://****.sugaropencloud.eu/rest/v10/) osoitteen kautta ja vaatii OAuth2-tokeneiden käyttöä. Kutsujen valtuutus tarkistetaan oauth-token:in avulla ja tarvittaessa oauth-token uusitaan refresh-token:in avulla. Tokeneiden käyttämisestä ja uusimisesta, kuten myös rajapinnan kutsujen suorittamisesta, pitää huolen Trainer/SugarCRM-moduuli.

Magenton omia rajapintoja ei ensisijaisesti yritetä käyttää, vaan luodaan tarvittaville toimintoille omat Action()-metodit kontrollereihin.

### 7.3 Käyttöliittymät

Alaluvuissa käydään läpi moduulin käyttöliittymän rakenteita.

#### 7.3.1 Käyttöliittymäkartta



## 7.3.2 Käyttöliittymäsivut

Sääntöjen listanäkymä							
							Luo uusi sääntö
Toiminnot							
	ID	Nimi	Nimi (frontend)	Tyyppi	Tila	Muokkaus aika	Luomisaika
Haku							
<input type="checkbox"/>							
<input type="checkbox"/>							
<input type="checkbox"/>							
<input type="checkbox"/>							

Sääntöjen listausnäköymästä ylläpitäjä näkee kaikki olemassa olevat säännöt. Listassa tulee näkyä vähintään sääntöjen nimet, tyyppi ja tila sekä luomis- ja muokkausajat. Listanäkymän kautta voidaan siirtyä luomaan uusia sääntöjä napin kautta tai muokkaamaan olemassaolevia valitsemalla ne listasta. Myös sääntöjen haku ja poistaminen onnistuu listasta.

**Sääntöjen luomis-/muokausnäky**

Nimi	<input style="width: 90%;" type="text" value="Ammattina Personal Trainer -opas"/> <small>Ohje</small>
Nimi (frontend)	<input style="width: 90%;" type="text" value="ammattina-personal-trainer"/> <small>Ohje</small>
Tila	<input style="border: 1px solid black; border-radius: 5px; text-align: center; padding: 2px 10px;" type="button" value="Käytössä"/> ▼ <small>Ohje</small>
Tyyppi	<input style="border: 1px solid black; border-radius: 5px; text-align: center; padding: 2px 10px;" type="button" value="Pikaopas"/> ▼ <small>Ohje</small>
Kategoria	<input style="border: 1px solid black; border-radius: 5px; text-align: center; padding: 2px 10px;" type="button" value="PT"/> ▼ <small>Ohje</small>
URL	<input style="width: 90%;" type="text" value="https://www.trainer4you.fi/..."/> <small>Ohje</small>
Paikkakunta kentät	<input style="width: 90%;" type="text" value="pt_paikkakunta"/> <small>Ohje</small>
Päivämäärä kentät	<input style="width: 90%;" type="text" value="apt_esite"/> <small>Ohje</small>
Numero kentät	<input style="width: 90%;" type="text"/> <small>Ohje</small>
"Käsittämätön" kentät	<input style="width: 90%;" type="text" value="kasittelematon"/> <small>Ohje</small>

Sääntöjen luomis-/muokausnäkyssä ylläpitäjä voi luoda tai muokata luomiaan sääntöjä. Näkyman kautta voidaan myös poistaa yksittäisiä sääntöjä. Jokaisen kentän alapuolella on kuvaus kentän tarkoituksesta ja tarvittaessa esimerkkiarvo.

### Moduulin konfigurointi

Save Config

Ajastettu siirto	<div style="border: 1px solid black; padding: 2px; display: inline-block;">Päällä ▼</div> <small>Ohje</small>
Käsiteltävien tietojen määrä per ajo	<input style="width: 90%;" type="text" value="50"/> <small>Ohje</small>
SugarCRM Analytics moduulin nimi	<input style="width: 90%;" type="text" value="t4ua_Analytics"/> <small>Ohje</small>
SugarCRM Contacts-Analytics linkityksen nimi	<input style="width: 90%;" type="text" value="t4ua_analytics_contacts"/> <small>Ohje</small>
Uudelleenlatauksen aikaraja (päivinä)	<input style="width: 90%;" type="text" value="30"/> <small>Ohje</small>
Asiakasryhmät joiden aktiviteetit tallennetaan	<div style="border: 1px solid black; padding: 2px;"> <div style="background-color: #2196f3; color: white; padding: 2px;">Prospekti</div> <div style="background-color: #2196f3; color: white; padding: 2px;">Customer</div> <div style="padding: 2px;">Ryhmä A</div> <div style="padding: 2px;">Ryhmä X</div> <div style="padding: 2px;">Ryhmä Y</div> </div> <small>Ohje</small>

Moduulin konfigurointinäkymään pääsee Magenton hallintapaneelin järjestelmäasetuksien kautta. Näkymässä voidaan määrittää moduulin yleisiä asetuksia. Jokaisen kentän alapuolella on kuvaus kentän tarkoituksesta ja tarvittaessa esimerkkiarvo.

Aktiviteettien listanäkymä							
Toiminnot							
	ID	Asiakasnumero	Nimi	Tyyppi	Tila	Muokausaika	Luomisaika
Haku							
<input type="checkbox"/>							
<input type="checkbox"/>							
<input type="checkbox"/>							
<input type="checkbox"/>							

Aktiviteettien listanäkymästä ylläpitäjä voi esikatsella Tietopankissa syntynyttä aktiviteettia. Tarvittaessa aktiviteetteja voidaan poistaa listanäkymän toimintojen avulla.



### 7.3.3 Komentorivipohjainen käyttöliittymä

Moduulilla tulee olla shell-skripti Magenton /shell-kansiossa. Tätä skriptiä ajetaan ajoitetusti Linux-järjestelmistä löytyvän crontab-ohjelman avulla, jolla voidaan määrittää tiettyjen tiedostojen/komentojen ajaminen tietyin aikavälein. Vaikka skripti on ensisijaisesti tarkoitettu tätä varten, voidaan sitä ajaa myös manuaalisesti. Siksi siltä tulee löytyä --help parametrilla saatava ohje.

Skriptillä tulee olla vähintään yksi komento: run(), joka ajaa funktioita Trainer/Databank-moduulin luokista, jotka käsittelevät uudet analytiikan tiedot. Komennolle täytyy pystyä antamaan vähintään yksi parametri: --force (esimerkiksi), jonka avulla synkronointia voidaan ajaa silloinkin kun se on asetettu pois päältä moduulin asetuksista. Yleisesti komennon asetukset määritellään aina Magenton järjestelmäasetuksissa sekä säännöissä.

## 8 Rajoitukset

### 8.1 Standardit

Käytetään Magento- ja SugarCRM-järjestelmien standardeja.

Trainer/Databank-moduuli ohjelmoidaan pääsääntöisesti PHP-kielillä ja konfiguroidaan XML-standardin mukaisesti. Joitakin admin käyttöliittymän toimintoja saatetaan joutua ohjelmoimaan JavaScript-kielen avulla, käyttäen jQuery-kirjastoa. jQuery on kuitenkin pääsääntöisesti käytössä Tietopankin puolella tietojen tallennusta varten AJAX:ia käyttäen niissä tilanteissa, joissa asiakkaan aktiviteettia ei tallenneta linkin avaamisen seurauksena.

Haut SugarCRM-järjestelmään tehdään PHP:n cURL-funktioiden avulla (Trainer/SugarCRM). SugarCRM palauttaa tiedot objektimuodossa, joka on muutettavissa PHP:n array() muotoiseksi tai käsiteltävissä sellaisenaan. SugarCRM-järjestelmään palautettava data on oltava SugarCRM-dokumentaation ja -järjestelmän hyväksymässä muodossa, yleensä JSON-muodossa.

Tietoliikenteen protokollana tulee käyttää ensisijaisesti SSL-suojattuja yhteyksiä eli HTTPS-protokollaa.

### 8.2 Ohjelmistorajoitukset

Ohjelmointikieliet, kirjastot ja merkintäkielet: PHP, HTML, JavaScript/jQuery, XML

Käyttöjärjestelmät: Linux

Käyttöliittymät: Magento hallintapaneeli

## 9 Tiedot ja tietokannat

### 9.1 Tietokannat

Tietokantana toimii Magento-järjestelmän käyttämä tietokanta. Tietokantaan luodaan uusia tauluja moduulin sääntöjä ja asiakkaiden aktiviteetteja varten. Luominen tehdään Magentossa käytettävien moduulin asennusskriptien avulla. Moduulin yleiset asetukset, jotka asetetaan Magenton järjestelmäasetuksissa, tallentuvat Magenton omaan konfigurointitauluun.

## 9.2 Talletettavat tiedot

Tietokantaan luotavien taulujen alustavat rakenteet on kuvattu seuraavaksi.

Hallintapaneelista määritettävien sääntöjen tiedot:

trainer_databank_rules				
Kenttä	Tyyppi	Pakollinen	Esimerkkiarvot	Kuvaus
rule_id	BIGINT	Kyllä	1 (automaattinen)	ID
is_active	SMALLINT	Kyllä	1	Onko sääntö päällä. Kyllä / Ei valinta.
name_admin	TEXT	Kyllä	Ammattina personal trainer -opas	Nimi (admin). Selkeyttä varten.
name_front	TEXT	Kyllä	ammattina-personal-trainer	Nimi (frontend). Tämän perusteella asiakkaan aktiivisuus tallentuu. Oltava yhteen ja mielellään uniikki.
type	TEXT	Kyllä	pikaopas	Minkä tyyppinen sääntö on. Pikaopas, video, artikkeli (pudotusvalikko). Vaikuttaa liidinlähteeseen.
category	TEXT	Kyllä	pt	Mihin kategoriaan sääntö kuuluu. PT, RAV, LL (pudotusvalikko).
redirect_url	TEXT	Ei	<a href="https://www.trainer4you.fi/">https://www.trainer4you.fi/</a> ...	Mihin (tiedostoon) asiakas ohjataan. Ensisijaisesti siis vain pikaoppaita varten. Tyhjänä ei ohjata.
place_fields	TEXT	Ei	pt_paikkakunta	Paikkakunta tiedon kenttä. Tyhjänä ei aseteta. Useampi pilkuin eroteltuna.
date_fields	TEXT	Ei	apt_esite	SugarCRM päivämääräkenttä aktiviteetille. Tyhjänä ei aseteta. Useampi pilkuin eroteltuna.
date_first_fields	TEXT	Ei	esite_first	SugarCRM päivämääräkenttä ensimmäiselle aktiviteetille. Tyhjänä ei aseteta. Useampi pilkuin eroteltuna.
int_fields	TEXT	Ei		SugarCRM numerokenttä aktiviteetille (esim. videon katselumäärä). Tyhjänä ei aseteta. Useampi pilkuin eroteltuna.
unprocessed_fields	TEXT	Ei	kasittelematon	Mihin kenttään asetetaan kontaktille "Käsittelemätön". Tyhjänä ei aseteta. Useampi pilkuin eroteltuna.
updated_at	TIMESTAMP	Kyllä	2017-01-01 13:10:00	Päivitysaika
created_at	TIMESTAMP	Kyllä	2017-01-01 13:00:00	Luomisaika

Asiakkaan aktiivisuudesta syntyvän datan tiedot:

trainer_databank_activity				
Kenttä	Tyyppi	Pakollinen	Esimerkkiarvot	Kuvaus
activity_id	BIGINT	Kyllä	1 (automaattinen)	ID
customer_id	INTEGER	Kyllä	1	Asiakkaan ID
name	TEXT	Kyllä	ammattina-personal-trainer	Aktiviteetin nimi. Täsmäyttävä säännön "name_front" kentän kanssa.
type	TEXT	Kyllä	pikaopas	Aktiviteetin tyyppi. Varmistaa, että käsitellään oikean tyyppin aktiviteettia (esimerkiksi pikaopas vs. video).
count	INTEGER	Kyllä	1	Aktiviteettien määrä. Oletuksena 1. Käytössä mm. videoiden ja artikkeleiden kanssa.
status	INTEGER	Kyllä	0	Aktiviteetin status. Esimerkiksi: 0 = uusi (oletus) 1 = käsitelty 2 = tiedon siirto epäonnistunut 3 = virheellinen tieto
updated_at	TIMESTAMP	Kyllä	2017-01-01 13:00:00	Päivitysaika
created_at	TIMESTAMP	Kyllä	2017-01-01 13:00:00	Luomisaika