

Aulis Ronkainen

Työvuorosuunnittelusovelluksen kehittäminen Enterprise Service Bus -arkkitehtuuriin

Opinnäytetyö
Tietojenkäsittelyn koulutusohjelma


Kesäkuu 2014




MAMK

University of Applied Sciences

KUVAILULEHTI

		Opinnäytetyön päivämäärä 2.6.2014
Tekijä(t) Aulis Ronkainen	Koulutusohjelma ja suuntautuminen Tietojenkäsittelyn koulutusohjelma	
Nimeke Työvuorosuunnittelusovelluksen kehittäminen Enterprise Service Bus -arkkitehtuuriin		
Tiivistelmä Smart Time Oy, joka on tämän opinnäytetyön toimeksiantaja, on mikkeliäinen ohjelmistoalan yritys. Vilttihattu on vuoropäiväkoti, jonka työvuorosuunnitteluprosessi oli raskas ja paljon aikaa vievä. Syntyi idea sovelluksesta, joka automatisoi tuota prosessia. Idean siivittämänä Smart Time Oy ja vuoropäiväkoti Vilttihattu aloittivat yhteistyön lokakuussa 2013 työvuorosuunnittelusovelluksen, joka on koko projektin ensimmäinen vaihe, osalta. Tuloksena syntyi työvuorosuunnittelusovellus, joka auttoi vuoropäiväkoti Vilttihattua keskittämään resursseja tärkeämpiin työtehtäviin. Samalla Smart Time Oy sai arvokasta kokemusta tällaisesta pilottiprojektista ja teknologista hyötyä järjestelmäintegroitipalveluiden tuottamiseen. Tämän opinnäytetyön tarkoitus on tutkia miten vuoropäiväkoti Vilttihatun työvuorosuunnitteluprosessia voisi automatisoida mahdollisimman paljon. Prosessin automatisointi on tärkeää, sillä prosessiin kuluu noin 30 tuntia työaikaa viikossa. Kaikki työaika, joka prosessiin käytetään, voitaisiin sijoittaa tehtäviin, jotka oikeasti vuoropäiväkodin työntekijöille kuuluvat. Opinnäytetyön teoriaosassa tutustutaan Apache ServiceMix -pakettiin ja sen komponentteihin, ja tarkastellaan termiä ESB tarkemmin ja verrataan sitä muihin järjestelmäintegroitiiarkkitehtuureihin. Työssä tutkitaan myös, mikä on REST -tekniikan idea ja mitä sillä voi tehdä. Lisäksi esitellään Drupal, jonka päälle työvuorosuunnittelusovelluksen näkymät on tehty, front-end -työkaluna. Käytännön osana opinnäytetyössä esitellään työvuorosuunnittelusovellus tietojärjestelmänä sekä tutkitaan mitä se pitää sisällään, ja kerrotaan miten teoriaosassa esiteltyjä tekniikoita on käytetty työvuorosuunnittelusovelluksen suunnittelussa ja toteutuksessa hyväksi.		
Asiasanat (avainsanat) Apache ServiceMix, ESB, Apache Camel, REST		
Sivumäärä 32	Kieli Suomi	URN
Huomautus (huomautukset liitteistä)		
Ohjaavan opettajan nimi Jukka Selin	Opinnäytetyöntoimeksiantaja Smart Time Oy	

DESCRIPTION

		Date of the bachelor's thesis 2.6.2014
Author(s) Aulis Ronkainen	Degree programme and option Business information technology	
Name of the bachelor's thesis Development of rota planning software in Enterprise Service Bus -architecture		
Abstract Smart Time Oy, which this thesis was assigned by, is a software development company with emphasis on enterprise application integration. Vilttihattu is a 24-hour day care center in Mikkeli. Vilttihattu had problems with their rota planning process which was very hard and time consuming to do. Idea of a software was born. A software that could automate that painful rota planning process. From that idea Smart Time Oy and 24-hour day care center Vilttihattu started together the development of such software in October of 2013. As a result a rota planning software was made which allowed Vilttihattu to allocate more resources to more important assignments. Purpose of this thesis was to examine how well the process of rota planning could be automated. The automation of the process was very important because every week about 30 hours of working time is wasted by this process. Working time that is lost in this manner should able to be spent on real assignments of employees. As a theoretical part of this thesis Apache ServiceMix software package is introduced with the components it has. Enterprise Service Bus is examined as a concept and compared with other Enterprise Application Integration architectures. Furthermore the idea behind REST is presented and also insides of Drupal are exposed. As a practical part of this thesis the information system behind the rota planning software is described and how are the methods specified in the theoretical part of the thesis implemented on the software.		
Subjectheadings, (keywords) Apache ServiceMix, ESB, Apache Camel, REST		
Pages 32	Language Finnish	URN
Remarks, notes on appendices		
Tutor Jukka Selin	Bachelor's thesis assigned by Smart Time Oy	

SISÄLTÖ

1	JOHDANTO	1
2	APACHE SERVICEMIX -PALVELUALUSTA	2
2.1	Enterprise Service Bus.....	7
2.2	Apache Camel.....	10
2.3	REST.....	11
3	SOVELLUKSEN KÄYTTÖLIITTYMÄ DRUPALILLA	13
4	TYÖVUOROSUUNNITTELU SOVELLUKSEN TOTEUTUS	15
4.1	Arkkitehtuuri.....	18
4.2	Tietokanta	19
4.3	Näkymät.....	21
4.3.1	Viikkonäkymä.....	21
4.3.2	Päivänäkymä	23
4.3.3	Lapset ja ryhmät -näkyä.....	24
4.3.4	Työvuorosunnittelu -näkyä	25
4.4	REST -palvelin	27
5	PÄÄTÄNTÖ	30
	LÄHTEET	31

1 JOHDANTO

Smart Time Oy, joka on tämän opinnäytetyön tilaaja, on ohjelmistoalan yritys, joka keskittyy pääasiassa integrointipalveluiden kehittämiseen ja myymiseen, mutta kehittää myös räätälöityjä sovelluksia tilauksesta. Toiminnanohjausjärjestelmien (Enterprise ResourcePlanning) integraatiot ovat suuri osa Smarttime:n toimintaa, joten syksyllä 2013 Smarttime:lla päätettiin perustaa uusi välityspalvelu, jonka tarkoituksena on toimia ESB:nä (Enterprise Service Bus) integraatioiden tehostamiseksi. ESB on viestikeskeisyydessään loistava hoitamaan esim. ostolaskujen reititystä eri sovellusten välillä. Vaikka ESB on suunniteltu hoitamaan toiminnanohjausjärjestelmien ja muiden yritystason sovellusten välistä integrointia, se soveltuu hyvin hallitsemaan myös räätälöityjen ohjelmistojen integraatiotarpeita.

Vilttihattu on mikkiläinen vuoropäiväkoti, jonka hoitajat ja hoidettavien lasten vanhemmat toimivat loppukäyttäjinä työvuorosuunnittelusovelluksessa. Ennen tätä projektia vuoropäiväkoti Vilttihatun työvuorosuunnitteluprosessi oli täysin manuaalinen. Tämän projektin tarkoitus on automatisoida vuoropäiväkoti Vilttihatun työvuorosuunnitteluprosessi. Todennäköisesti prosessia ei saada kokonaan automatisoitua, johtuen sovelluksen suoritettavista heuristisista toiminnoista, mutta myös osittainen automatisointi auttaa hoitajia tekemään työnsä, lapsien hoitamisen, tehokkaammin.

Opinnäytetyön toisessa luvussa tutustutaan Apache ServiceMix -nimiseen ohjelmistopakettiin, johon päädyttiin Smarttime:lla muiden ESB -ohjelmistojen joukosta. Apache ServiceMix:n päätarkoitus on toteuttaa ESB -arkkitehtuuria palvelukeskeisessä arkkitehtuurissa. Luvussa kerrotaan, mikä ApacheServiceMix on, mihin sitä käytetään ja mitä se pitää sisällään. Työvuorosuunnittelusovelluksen palvelinpään logiikkaa hoitaa juuri Apache ServiceMix ja on siksi tärkeä osa tätä opinnäytetyötä ja koko projektia.

Kolmannessa luvussa tarkastellaan mikä on Drupal, miten se liittyy projektiin ja miten Drupalia voidaan käyttää *front-end* työkaluna. Samalla myös kerrotaan, miksi juuri Drupal on valittu sisällönhallintajärjestelmäksi. Neljännessä luvussa käsitellään työvuorosuunnittelusovelluksen toteutuksen vaiheita, aikataulua ja ratkaisuja, joita tehtiin sovellusta varten, ja sitä miten niissä onnistuttiin. Luvussa on tarkoitus kuvata työvuorosuunnittelusovelluksen tietojärjestelmä kokonaisuudessaan.

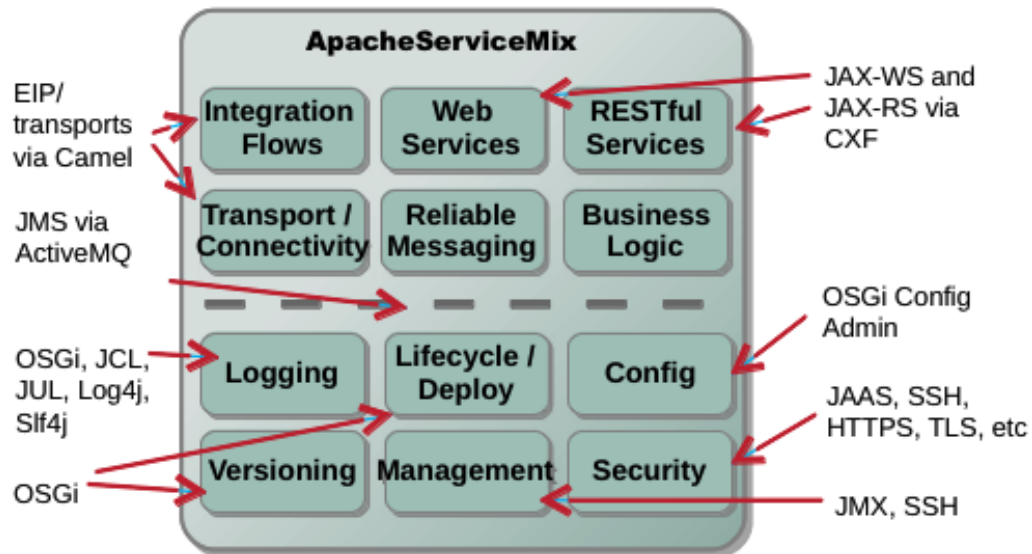
Projekti toteutetaan kolmessa vaiheessa. Ensimmäinen vaihe on neljännessä luvussa kuvailtava työvuorosuunnittelusovelluksen toteutus. Toisena vaiheena on toteutus sovelluksesta, jonka tarkoitus on luoda raportti päivittäisestä ruokatilauksesta. Kolmas vaihe käsittää toteutuksen sovelluksesta, jolla vanhemmat ilmoittavat lastensa hoitoajat. Tätä opinnäytetyötä kirjoittaessa työvuorosuunnittelusovellus on tuotannossa, mutta testaus on vielä kesken eikä se siten ole täysin valmis. Ohjelmointivirheitä korjataan testauksen edetessä. Vaiheet kaksi ja kolme ovat vielä kesken, mutta niiden valmistuminen on olennaista, jotta työvuorosuunnitteluprosessin automatisointi olisi täydellisempää ja projektin tarkoitus ja hyödyt tulisivat esille sellaisina kuin ne ovat suunniteltu.

2 APACHE SERVICEMIX -PALVELUALUSTA

Apache ServiceMix on avoimen lähdekoodin kokoelma integrointikomponentteja, joiden avulla on tarkoitus rakentaa omia integrointiratkaisuja palvelukeskeiseen arkkitehtuuriin. Apache ServiceMix toteuttaa käytännössä ESB -arkkitehtuuria ja on kehitetty toimimaan JBI -spesifikaation (Java Business Integration) mukaan. Kokonaisuudessaan Apache ServiceMix muodostaa kaiken tarvittavan järjestelmäintegraatioiden kehittämiseen ja hallintaan. Apache ServiceMix yhdistää toiminnot seuraavista Apachen projekteista:

- Apache ActiveMQ – Viestitys- ja välitystoiminnot
- Apache Camel – Reititystoiminnot
- Apache CXF–*Web Service* -toiminnot
- Apache Karaf – Käyttöönotto- ja asennustoiminnot

(Mielke 2013; The Apache Software Foundation 2008a.)



KUVA 1. Apache ServiceMix:n teknologiat ja standardit (Mielke 2013)

Kuvassa 1 on esitelty Apache ServiceMix:ssä käytettäviä tekniikoita ja standardeja. Katkoviivan alapuolella olevat ominaisuudet kuuluvat Apache Karaf:iin. Vastaavasti katkoviivan yläpuolella on kuvattuna Apache Camel:n, Apache CXF:n ja Apache ActiveMQ:n ominaisuuksia. Kuten kuvasta voidaan huomata, sisältää Apache ServiceMix kokonaisuudessaan paljon ominaisuuksia ja komponentteja. Apache ServiceMix:n kaikkia komponentteja tai ominaisuuksia ei ole pakko käyttää. Niitä voidaan poistaa käytöstä tai jopa tehdä itse uusia. Apache ServiceMix on modulaarinen kokonaisuus (The Apache Software Foundation 2008b).

Apache ActiveMQ on avoimen lähdekoodin viestinvälittäjä. Apache ActiveMQ:n toiminnan mahdollistaa JMS -rajapinta (Java Message Service), jota Apache ActiveMQ implementoi (The Apache Software Foundation 2004a). JMS itsessään toteuttaa Message Oriented Middleware -mallia (MOM) (Oracle Corporation 2010). Muita välittäjäpalveluja voidaan myös käyttää, mutta Apache ActiveMQ on sopivasti valmiiksi Apache ServiceMix -paketissa (The Apache Software Foundation 2008a; Mielke 2013).

Apache Camel on Apache ServiceMix:n reitityskomponentti. Apache Camel:n toiminta perustuu reittikonfiguraatioihin, joiden mukaan viestejä lähetetään ja vastaanotetaan. Apache Camel käyttää komponentteja, joiden avulla se tukee monia protokollia ja malleja. (The Apache Software Foundation 2004b.) Apache CXF on avoimen lähdekoodin sovelluskehys, joka auttaa kehittämään www-sovelluspalveluita käyttäen JAX-WS- ja JAX-RS -rajapintoja. Apache CXF on osa Apache ServiceMix -pakettia. (The Apache Software Foundation 2008c.)

Apache Karaf on käyttöönotto- ja asennustyökalu, jolla voidaan ottaa käyttöön uusia komponentteja ja sovelluksia. Apache Karaf soveltuu hyvin Apache ServiceMix:n hallintatyökaluksi. Apache Karaf:ssa on mm. seuraavia ominaisuuksia:

- Sovellusten ja komponenttien käyttöönotto lennosta (hot deployment)
- Konsoli, johon komennot syötetään
- Keskitetty lokijärjestelmä
- Dynaaminen konfigurointi
- Etähallinta SSH:n tai asiakasohjelmiston kautta
- Natiivi käyttöjärjestelmäintegraatio (asennus palveluksi)

(The Apache Software Foundation 2008d.)

Apache ServiceMix on saatavilla Windows- ja Linux -käyttöjärjestelmille. ServiceMix:n suorittamiseen tarvitaan JRE (Java Runtime Environment) ja noin 100 megatavua vapaata levytilaa. Jos on tarve kehittää omia integrointisovelluksia, tarvitaan myös JDK (Java Development Kit) ja Apache Maven. Apache Maven on projektihallintatyökalu, joka lataa projektiin tarvittavat paketit konfiguraation mukaan (The Apache Software Foundation 2002). Toisin sanoen Apache Maven pitää huolta projektin komponenttien versioista, jotta komponenttien välinen yhteensopivuus säilyisi. ServiceMix -paketti on ladattavissa ServiceMix:n viralliselta verkkosivulta. Paketissa ei ole asennusohjelmaa, vaan paketti puretaan ja sen jälkeen suoritetaan komentokehoteessa komennolla *servicemix*. Komennon syöttämisen jälkeen ServiceMix käynnistää Apache Karaf:n, jolla asennettua ServiceMix -instanssia voidaan hallita komentokehoteesta (kuva 2). (The Apache Software Foundation 2008e).

```

C:\Users\Student\Desktop\apache-servicemix-5.0.1\bin>servicemix
WARNING: Running karaf on a Java HotSpot Client VM because server-mode is not available.
Install Java Developer Kit to fix this.
For more details see http://java.sun.com/products/hotspot/whitepaper.html#client
servicemix.bat: JAVA_HOME not set; results may vary
Java HotSpot(TM) 64-Bit Server VM warning: ignoring option MaxPermSize=256M; support was removed in 8.0
Please wait while Apache ServiceMix is starting...
 89% [=====>]

ServiceMix

Apache ServiceMix (5.0.1)

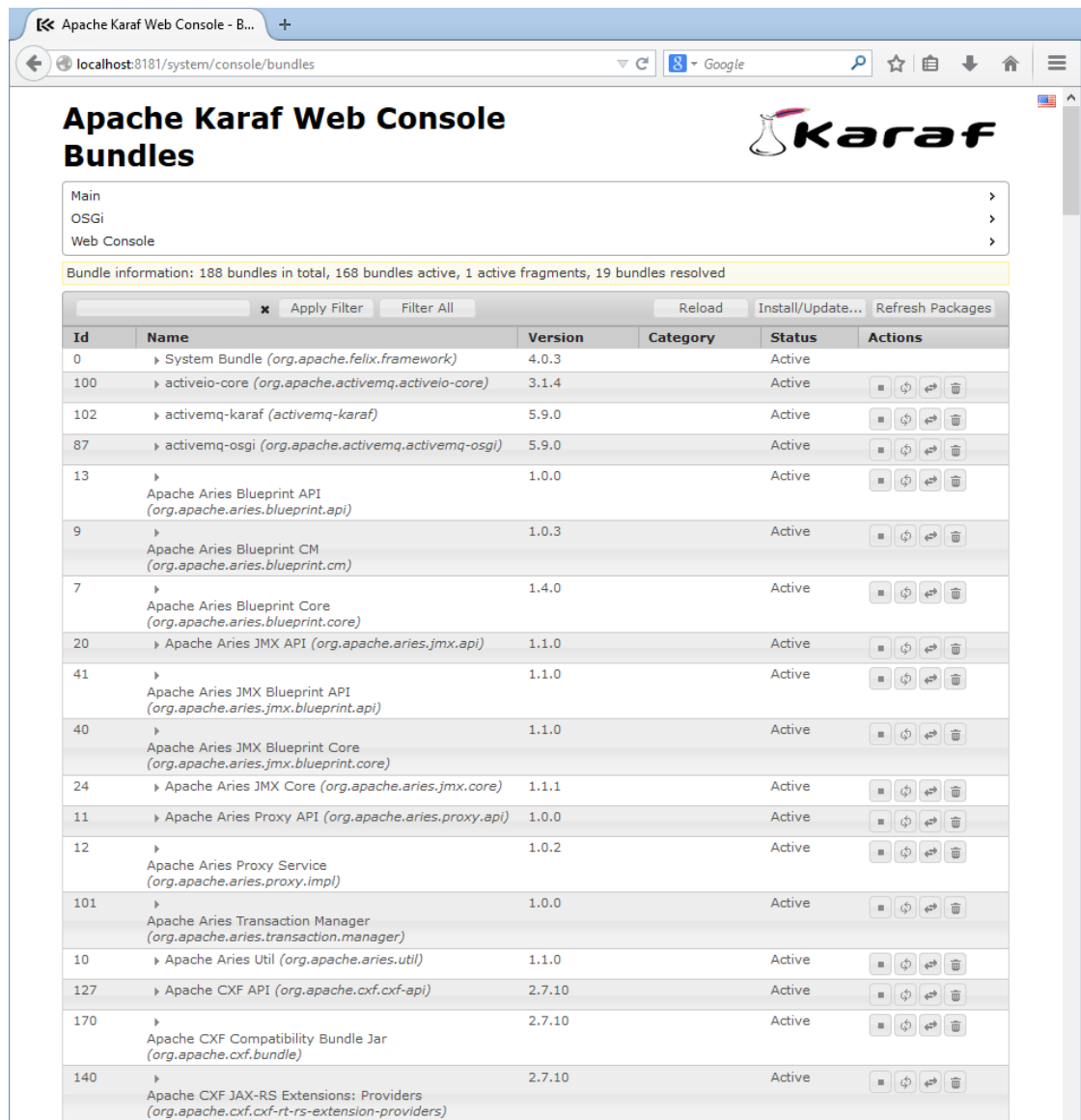
Hit '<tab>' for a list of available commands
and '[cmd] --help' for help on a specific command.
Hit '<ctrl-d>' or 'osgi:shutdown' to shutdown ServiceMix.

karaf@root>

```

KUVA 2. Apache ServiceMix:n suorittaminen Windows (x64) -ympäristössä

Apache Karaf:lla voidaan asentaa erilaisia ominaisuuksia tai komponentteja ServiceMix:iin. Ominaisuudet löytyvät syöttämällä komento *features:list* Karaf -konsoliin. Jos halutaan asentaa esimerkiksi hallintaa helpottava *Web console* -ominaisuus, syötetään Karaf -konsoliin komento *features:installwebconsole*. Kun *Web console* on asennettu, voidaan se avata selaimella osoitteesta *http://localhost:8181/system/console* ja kirjautua tunnuksella *smx* ja salasanalla *smx* (jos oletusasetuksia ei muuteta) (kuva 3). *Web console*:lla voidaan graafisesti hallita ServiceMix:iä samaan tapaan kuin Karaf:lla. (The Apache Software Foundation 2008f).



Apache Karaf Web Console
Bundles

Main >
OSGi >
Web Console >

Bundle information: 188 bundles in total, 168 bundles active, 1 active fragments, 19 bundles resolved

Id	Name	Version	Category	Status	Actions
0	System Bundle (<i>org.apache.felix.framework</i>)	4.0.3		Active	
100	activeio-core (<i>org.apache.activemq.activeio-core</i>)	3.1.4		Active	■ ↻ ↺ 🗑
102	activemq-karaf (<i>activemq-karaf</i>)	5.9.0		Active	■ ↻ ↺ 🗑
87	activemq-osgi (<i>org.apache.activemq.activemq-osgi</i>)	5.9.0		Active	■ ↻ ↺ 🗑
13	Apache Aries Blueprint API (<i>org.apache.aries.blueprint.api</i>)	1.0.0		Active	■ ↻ ↺ 🗑
9	Apache Aries Blueprint CM (<i>org.apache.aries.blueprint.cm</i>)	1.0.3		Active	■ ↻ ↺ 🗑
7	Apache Aries Blueprint Core (<i>org.apache.aries.blueprint.core</i>)	1.4.0		Active	■ ↻ ↺ 🗑
20	Apache Aries JMX API (<i>org.apache.aries.jmx.api</i>)	1.1.0		Active	■ ↻ ↺ 🗑
41	Apache Aries JMX Blueprint API (<i>org.apache.aries.jmx.blueprint.api</i>)	1.1.0		Active	■ ↻ ↺ 🗑
40	Apache Aries JMX Blueprint Core (<i>org.apache.aries.jmx.blueprint.core</i>)	1.1.0		Active	■ ↻ ↺ 🗑
24	Apache Aries JMX Core (<i>org.apache.aries.jmx.core</i>)	1.1.1		Active	■ ↻ ↺ 🗑
11	Apache Aries Proxy API (<i>org.apache.aries.proxy.api</i>)	1.0.0		Active	■ ↻ ↺ 🗑
12	Apache Aries Proxy Service (<i>org.apache.aries.proxy.impl</i>)	1.0.2		Active	■ ↻ ↺ 🗑
101	Apache Aries Transaction Manager (<i>org.apache.aries.transaction.manager</i>)	1.0.0		Active	■ ↻ ↺ 🗑
10	Apache Aries Util (<i>org.apache.aries.util</i>)	1.1.0		Active	■ ↻ ↺ 🗑
127	Apache CXF API (<i>org.apache.cxf.cxf-api</i>)	2.7.10		Active	■ ↻ ↺ 🗑
170	Apache CXF Compatibility Bundle Jar (<i>org.apache.cxf.bundle</i>)	2.7.10		Active	■ ↻ ↺ 🗑
140	Apache CXF JAX-RS Extensions: Providers (<i>org.apache.cxf.cxf-rt-rs-extension-providers</i>)	2.7.10		Active	■ ↻ ↺ 🗑

KUVA 3. Apache Karaf:lla asennettu valinnainen ominaisuus *Web Console*

Apache ServiceMix on yksi ESB -ohjelmistoista. ESB -ohjelmistoja on olemassa maksullisia ja maksuttomia. Yleensä ESB -ohjelmistot, jotka ovat maksullisia, ovat melko kalliita ja järeitä. Siksi nykyään kehittäjät, varsinkin ketteriä projektinhallintamenetelmiä käyttävät, mielellään tutustuvat ilmaisiin avoimen lähdekoodin ESB -implementaatioihin. Maksulliset ESB -ohjelmistot ovat yleensä suljetun lähdekoodin ohjelmistoja, joten ohjelmien kustomointi ja ominaisuuksien lisääminen voi olla vaikeaa tai jopa mahdotonta. Muita ESB -sovelluksia ovat mm.:

- Oracle Service Bus
- Mule ESB
- Fuse ESB – Pohjautuu Apache ServiceMix:iin
- Talend ESB

- WSO2 ESB

(Wähner2013.)

2.1 Enterprise Service Bus

Enterprise Service Bus (ESB) on arkkitehtuurimalli, jolla järjestelmäintegraatioita toteutetaan palvelukeskeisessä arkkitehtuurissa. ESB -mallia käytetään yleensä organisaatioissa, joilla on vähintään kolmen järjestelmän integrointitarve. ESB:tä voi myös käyttää vaikka integrointitarve olisi pienempi, mutta silloin yleensä *Point-to-Point* -integraatiot ovat ketterämpiä ja halvempia. (Mason 2009.)

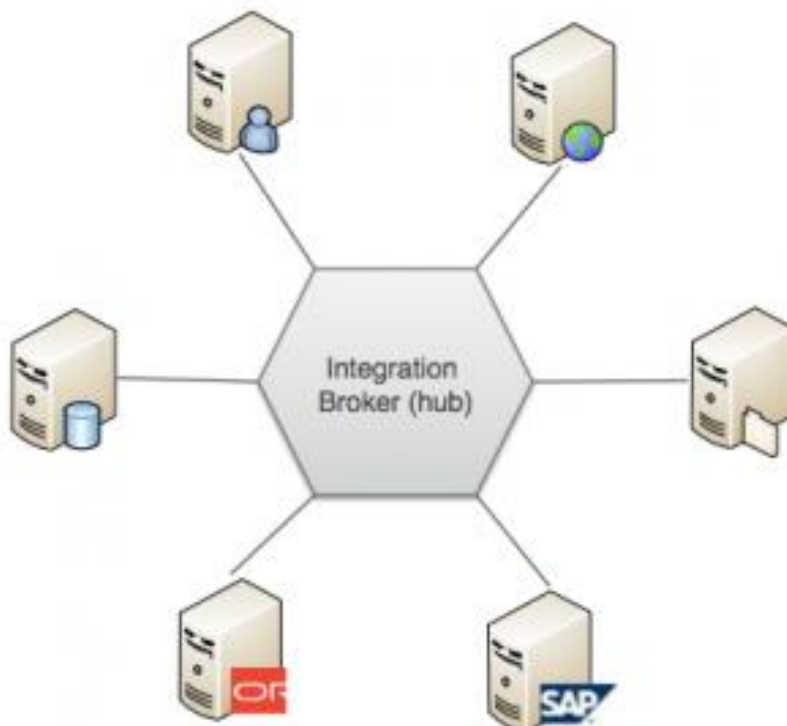
ESB -termiä ei ole virallisesti määritelty, eikä siten standardoitu. Siitä johtuen ohjelmistot, jotka toteuttavat ESB -konseptia, usein sisältävät varsin erilaisia ominaisuuksia. ESB -käsite voi olla nimeltään hieman harhaan johtava, sillä kyseessä ei ole fyysinen väylä, ei ainakaan siinä mielessä missä se normaalisti mielletään. Kyseessä on looginen *wäylä*, joka kuvaa ESB:n integraatiokykyä ja hajautettavuutta (ESB -arkkitehtuuri). (Kress ym. 2013; Mason 2011a.)

Point-to-Point -malli on yksinkertaisin integraatio kahden järjestelmän välillä. *Point-to-Point* -integraatiossa kaksi järjestelmää integroidaan keskenään rajapintojen avulla (kuva 4). Kahden järjestelmän integroiminen *Point-to-Point* -mallilla on yleensä helppoa ja kustannustehokasta, mutta järjestelmien määrän kasvaessa integraatioiden määrä kasvaa eksponentiaalisesti. Kun integroitavia järjestelmiä on vähän, *Point-to-Point* -integraatio voidaan vaivattomasti räätälöidä niille järjestelmille sopivaksi. (MuleSoft 2013.)



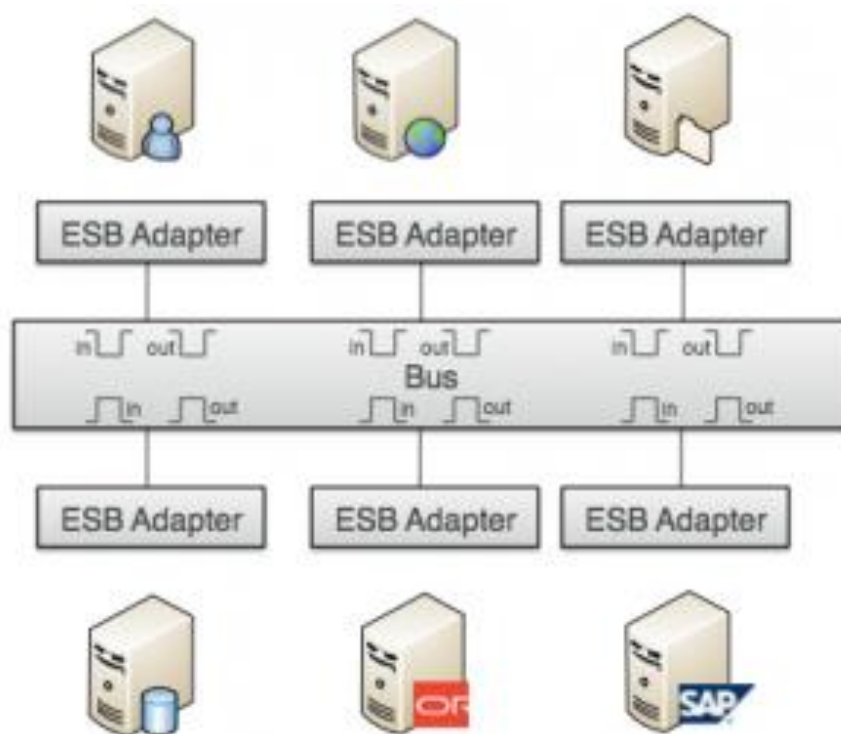
KUVA 4. Point-to-Point -integraatio

Kun integrointitarve tarve lisääntyy ja integroitavia järjestelmiä on paljon, *Point-to-Point* -integraatiot eivät ole enää järkeviä. Silloin tarvitaan integrointimalli, joka eliminoi tarpeen *Point-to-Point* -integraatioille. Tällainen integraatioarkkitehtuuri on *Hub-and-Spoke*, joka käyttää keskitettyä välittäjää (*hubia*) integraatioiden ohjaamiseen (kuva 5). *Hub-and-Spoke* -mallissa keskitetty välittäjä hallitsee kaikkia integraatioita. (Mason 2011b.)



KUVA 5. Hub-and-Spoke -integraatiomalli (Mason 2011a)

Enterprise Service Bus on hyvin paljon samanlainen arkkitehtuuri kuin *Hub-and-Spoke*. Eroja kuitenkin on ja näitä kahta arkkitehtuuria ei pidä sekoittaa keskenään. ESB on tavallaan kehittyneempi versio *Hub-and-Spoke* -arkkitehtuurista. Suurin ero ESB:n ja *Hub-and-Spoke*:n välillä on, että ESB:ssä ei ole keskitettyä välittäjää, vaan ESB:ssä välittäjä toimii hajautetusti. Jokaisella järjestelmällä on oma adapteri, joka vaihtaa datan sellaiseksi, että järjestelmä pystyy sitä käsittelemään (kuva 6). Adaptereista muodostetaan reitit niihin järjestelmiin, jotka halutaan integroida, ja tällä tavalla päästään eroon *Point-to-Point* -integraatioista. (Mason 2011b; MuleSoft 2013.)



KUVA 6. ESB -arkkitehtuuri (Mason 2011a)

ESB on näiden arkkitehtuurien joukosta paras, silloin kun halutaan hyvin skaalautuva järjestelmä ja kun halutaan tehdä palveluiden hallinnasta ja jatkokehityksestä vaivatonta. ESB:n etuja ovat mm.:

- Helppo hallittavuus
- ESB on tilaton; tila kulkee viestin mukana
- Skaalautuvuus ja hajautettavuus
- Tekee *Point-to-Point* -integraatioista turhia
- Versiointi – komponentteja voidaan versioida erikseen
- Tekniikoiltaan vanhojen järjestelmien integrointi mahdollista

(MuleSoft 2013; Mason 2011b.)

ESB, kuten mikään muukaan arkkitehtuuri, ei ole täydellinen. Joskus on parempi implementoida *Hub-and-Spoke*-, *Point-to-Point*- tai jokin muu integraatioarkkitehtuuri. Tärkeintä on aina miettiä, mikä integraatiomalleista soveltuu parhaiten käyttötarkoitukseen. ESB:n huonoja puolia ovat:

- Hitaus, varsinkin jos järjestelmät ovat jo valmiiksi yhteensopivia
- Maksulliset ESB -sovellukset ovat kalliita
- Kehittämisen monimutkaisuus

(MuleSoft 2013; Mason 2011b.)

2.2 Apache Camel

Apache Camel on avoimen lähdekoodin integrointisovelluskehys, joka perustuu tunnettuihin järjestelmäintegraatiomalleihin (EIP). Apache Camel:n tärkein tehtävä on reitittää viestejä. Apache Camel:n toiminta perustuu komponentteihin, joiden avulla Apache Camel tukee monia eri malleja. Camel:iin voidaan määritellä reittejä usealla eri tavalla esim. Java -DSL:llä (Domain-Specific Language) tai Spring XML:llä, jota käytetään työvuorosunnittelusovelluksen tapauksessa. Reitissä määritellään, mistä reitti tulee ja minne se menee. Reitien määrittelyyn on olemassa monia erilaisia ominaisuuksia, joita tässä opinnäytetyössä ei käydä läpi. Helpoin tapa *deploy*:ta eli käynnistää reitti on kopioida reittikonfigurointitiedosto Apache ServiceMix:n *deploy* -kansioon, josta ServiceMix automaattisesti yrittää asentaa paketteja. Esimerkissä reitti lukee *camel/input* -kansioista tiedostoja, jotka sinne ilmaannuttuaan siirtyvät *camel/output* -kansioon (kuva 7). (The Apache Software Foundation 2004b.)

```

<?xml version="1.0" encoding="UTF-8"?>
<blueprint
  xmlns="http://www.osgi.org/xmlns/blueprint/v1.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="
    http://www.osgi.org/xmlns/blueprint/v1.0.0
    http://www.osgi.org/xmlns/blueprint/v1.0.0/blueprint.xsd">
  <camelContext xmlns="http://camel.apache.org/schema/blueprint">
    <route>
      <from uri="file:camel/input"/>
      <log message="Moving ${file:name} to the output directory"/>
      <to uri="file:camel/output"/>
    </route>
  </camelContext>
</blueprint>

```

KUVA 7. Yksinkertainen Apache Camel reittikonfiguraatio (The Apache Software Foundation 2008g)

Periaatteessa Camel:kin voi olla ESB, johtuen laajoista ominaisuuksista, joista suuri osa määritellään kuuluvan ESB -arkkitehtuuriin (The Apache Software Foundation 2004c). Camel:ssa on paljon eri komponentteja. Nämä komponentit mahdollistavat sen, että Camel osaa tulkita ja reitittää dataa, tuli sitten minkälaisesta kohteesta tai minkälaisessa muodossa tahansa. Camel -komponentteja ovat mm. (komponentteja on paljon enemmän, tässä vain muutama esimerkki):

- Atom – lukee Atom -syötettä
- FTP – lähettää ja vastaanottaa tiedostoja FTP -protokollalla
- GMail – lähettää sähköpostia käyttämällä Gmail -tiliä
- IRC – IRC -kommunikointiin
- POP3 – vastaanottaa sähköpostia
- Restlet – käyttää REST -*endpoint*:ia
- SQL – Suorittaa SQL -kyselyjä
- SSH – lähettää SSH -komentoja palvelimelle
- Twitter – käyttää Twitter -*endpoint*:ia

(The Apache Software Foundation 2004d.)

2.3 REST

REST (Representational State Transfer) on tekniikka, jolla käytetään HTTP -protokollaa www-sovelluspalveluiden toteuttamiseen. REST:n yksinkertainen toimin-

tatapa ja helppo implementointi on lisännyt REST:n suosiota ohjelmoijien keskuudessa. REST toimii HTTP -pyynnöillä, joita asiakasohjelmisto lähettää ja joihin REST -palvelin vastaa. HTTP -pyyntöjä on useita ja jokaisella on erilainen tehtävä (taulukko 1). Myös omia metodeja voidaan muodostaa ja käyttää, vaikka REST on suunniteltu käyttämään HTTP/1.1 -spesifikaatioissa mainittuja metodeja. REST -palvelin vastaanottaa pyynnön ja suorittaa pyynnön mukaisia CRUD -tietokantatoimintoja (Create, Read, Update, Delete). Sovellus, joka suorittaa tällaisia operaatioita on RESTful -sovellus. (Singh 2009.)

Metodi	Näkyvyysalue	Kuvaus
GET	Kokoelma	Hakee kaikki kokoelman resurssit
GET	Resurssi	Hakee yhden resurssin
HEAD	Kokoelma	Hakee kaikki kokoelman resurssit
HEAD	Resurssi	Hakee yhden resurssin
POST	Kokoelma	Tekee uuden resurssin kokoelmaan
PUT	Resurssi	Päivittää resurssin
PATCH	Resurssi	Päivittää resurssin
DELETE	Resurssi	Poistaa resurssin
OPTIONS	Mikä tahansa	Palauttaa saatavilla olevat HTTP -metodit ja muut vaihtoehdot

TAULUKKO 1. HTTP/1.1 -pyynnöt (Jansen 2011)

Www-sovelluspalvelut ovat yleistyneet ja aikaisemmin paljon esillä ollut SOAP -protokolla kärsii luottamuspulasta, sillä REST:n suosio kasvaa koko ajan verrattuna SOAP:iin. REST:n matala oppimiskynnys on varmasti vaikuttanut asiaan. Kokemattomankin käyttäjän on helppo havaita REST:ssä ominaisuuksia, jotka ovat helppo oppia. REST:n parhaimpia puolia ovat:

- Kevyt, ei vaadi ylimääräisten ohjelmien asennusta
- Tulokset ovat ihmisten luettavissa

- Helppo kehittää, ei ole monimutkainen tekniikaltaan
- Ei ole ohjelmointikieli riippuvainen

(Singh 2009.)

REST myös SOAP:iin verrattuna käyttää vähemmän tietoliikennekaistaa johtuen REST:n yksinkertaisuudesta. Vaikka vaikuttaa, että REST olisi joka osa-alueella parempi, niin myös REST:ssä on huonoja puolia. Täydellistä protokollaa tai tekniikkaa tuskin on olemassa. REST:n huonoja puolia ovat mm.:

- Ei sovi jokaiseen www-sovelluspalveluun, joskus SOAP on sopivampi
- Hankala lähettää paljon dataa, SOAP lähettää liitteet paremmin
- Ei ole standardi, joten ei ole kunnolla määritelty

(Singh 2009.)

3 SOVELLUKSEN KÄYTTÖLIITTYMÄ DRUPALILLA

Drupal on PHP:llä toteutettu suosittu sisällönhallintajärjestelmä (CMS). Drupal soveltuu mm. www-sivujen toteuttamiseen. Drupalissa on paljon ominaisuuksia, mistä johtuen se on yksi suosituimmista sisällönhallintajärjestelmistä. Drupalia yleensä käytetään verkkosivujen tekemiseen. Erityisen aloittelijaystävällisen Drupalista tekee mahdollisuus luoda verkkosivuja ilman että tarvitsee opetella ohjelmointia. Drupal mahdollistaa monenlaisien verkkosivujen tekemisen, tiettyä mallia ei ole pakko noudattaa. Drupalilla voi myös luoda sivut ohjelmoimalla ja itse Drupalin ohjelmakoodia voi muokata tarvittaessa. Drupalin ominaisuuksiin kuuluvat mm.:

- Käyttäjähallinta
- Sivujen, blogien ja viestien julkaisu
- Dashboard, jolla muokataan sivun ulkoasua
- Moduulit, jotka laajentavat Drupalin ominaisuuksia
- Selaimesta käytettävä käyttöliittymä, jolla Drupalia hallitaan

(Hunter 2008.)

Drupalin asentaminen vaatii verkkopalvelimena Apache:n, IIS:n (Internet Information Services) tai Nginx:n. Lisäksi vaaditaan PHP 5.2 tai uudempi ja MySQL-, PostgreSQL- tai SQLite -tietokannan. Drupal vaatii minimissään 15 megatavua vapaata levytilaa ja 60 megatavua sivulle, jossa on paljon moduuleja ja teemoja asennettuna. Drupal ei kuitenkaan vaadi mitään tiettyä käyttöjärjestelmää, sillä Drupal on tavallaan

itsekin verkkosivu, ja sen hallinta ja verkkosivujen luonti tapahtuu selaimella. (Kennedy 2003.)

Drupalin mukana tulee monia erilaisia moduuleja ja niitä voi myös ladata internetistä. Moduulien tarkoitus on laajentaa tai luoda täysin uusia ominaisuuksia Drupaliin. Moduuleita voi myös tehdä itse. Moduulit ovat kätevä tapa tehdä Drupalista monipuolisemman, mutta moduuleita voi melko nopeasti kertyä Drupaliin liikaa, jolloin soveluksenkehitys on vaikeampaa. Työvuorosuunnittelusovelluksen eri päänäkyvät ominaisuuksineen on toteutettu Drupal -moduuleina (kuva 8). (Hunter 2008.)

KÄYTTÖSSÄ	NIMI	VERSIO	KUVAUS	TOIMENPITEET
<input checked="" type="checkbox"/>	DCP		DayCare Planner Module Vaatii nimä: DCP Rest Client (päällä), DCP End-User (päällä), DCP End-User Calendar Day (päällä), DCP End-User Calendar Week (päällä), DCP End-User Canteen Report (päällä), DCP End-User Group (päällä), DCP End-User Work Time Planner (päällä)	Ohje
<input checked="" type="checkbox"/>	DCP Rest Client		DayCare Planner Rest Client Module Vaatii nimä: DCP (päällä) Vaatii data: DCP End-User (päällä), DCP End-User Calendar Day (päällä), DCP End-User Calendar Week (päällä), DCP End-User Canteen Report (päällä), DCP End-User Group (päällä), DCP End-User Work Time Planner (päällä)	Ohje
DCP END-USER				
<input checked="" type="checkbox"/>	DCP End-User		DayCare Planner End-User Module Vaatii nimä: DCP (päällä), DCP Rest Client (päällä) Vaatii data: DCP End-User Calendar Day (päällä), DCP End-User Calendar Week (päällä), DCP End-User Canteen Report (päällä), DCP End-User Group (päällä), DCP End-User Work Time Planner (päällä)	Ohje Käyttöohjeet
<input checked="" type="checkbox"/>	DCP End-User Calendar Day		DCP End-User Calendar Day Module Vaatii nimä: DCP (päällä), DCP Rest Client (päällä), DCP End-User (päällä)	Ohje Käyttöohjeet
<input checked="" type="checkbox"/>	DCP End-User Calendar Week		DCP End-User Calendar Week Module Vaatii nimä: DCP (päällä), DCP Rest Client (päällä), DCP End-User (päällä)	Ohje Käyttöohjeet
<input checked="" type="checkbox"/>	DCP End-User Canteen Report		DCP End-User Canteen Report Module Vaatii nimä: DCP (päällä), DCP Rest Client (päällä), DCP End-User (päällä)	Ohje Käyttöohjeet
<input checked="" type="checkbox"/>	DCP End-User Group		DCP End-User Group Module Vaatii nimä: DCP (päällä), DCP Rest Client (päällä), DCP End-User (päällä), Chaos tools (päällä)	Ohje Käyttöohjeet
<input checked="" type="checkbox"/>	DCP End-User Work Time Planner		DCP End-User Work Time Planner Module Vaatii nimä: DCP (päällä), DCP Rest Client (päällä), DCP End-User (päällä)	Ohje Käyttöohjeet

KUVA 8. Työvuorosuunnittelusovelluksen moduulit Drupalissa

Työvuorosuunnittelusovelluksessa näkyvät ovat toteutettu Drupalilla. Drupalin oletusnäkyvät ovat ensin poistettut. Hallintänäkyvät pysyvät ennallaan. Sen jälkeen työvuorosuunnittelusovellukselle on tehty oma Drupal -teema, jonka pystyy vaihtamaan nopeasti. Sitten sovelluksen käyttöliittymät on luotu HTML- ja JavaScript -kielillä ja tyyli luotu CSS -tyylitiedostoilla. Ominaisuudet on ohjelmoitu PHP -kielellä Drupaliin, joka lähettää REST -asiakasohjelmiston avulla pyynnöt REST -palvelimelle. Työvuorosuunnittelusovelluksen toteutuksessa Drupal on osoittautunut käteväksi työkaluksi. Drupaliin on helppo ladata ominaisuuksia moduuleiden muodossa, jotka helpottavat ohjelmoijan työtä. Monet sellaiset ominaisuudet, joita aikaisemmin täytyi ohjelmoida itse, onkin joko jo valmiiksi Drupaliin integroitu tai ladattavissa moduuleina. Drupal on valittu *front-end* kehikseksi työvuorosuunnittelusovellukseen mm. seuraavista syistä:

- Drupalissa on sisäänrakennettu käyttäjähallinta, joka käyttää Drupalin tietokantaa

- Drupalin lomake- ja virhekäsittely on pitkälle kehitetty
- Drupalin sisäänrakennettu käännösten hallinta mahdollistaa nopean kielen vaihtamisen tarvittaessa
- Drupalin moduulijärjestelmä mahdollistaa moduulien lisäämisen ja poistamisen tarvittaessa
- Smart Time Oy on käyttänyt Drupalia muissa projekteissa ja se on havaittu toimivaksi *front-end* ratkaisuksi, vaikka muitakin vaihtoehtoja olisi
- Mikkelin kaupungin verkkosivusto on myös Drupal -pohjainen
- Käyttöliittymissä tarvittavat lisäosat ovat helppoja asennettavia

4 TYÖVUOROSUUNNITTELUKESKUSTELUN TOTEUTUS

Vuoropäiväkotin Vilttihaatan hoitajat toimivat loppukäyttäjänä työvuorosuunnittelusovellukselle. Työvuorosuunnittelusovelluksen suunnittelu alkoi lokakuussa 2013, ja tarkoitus oli toteuttaa käyttöliittymät ensin ja toiminnallisuudet jälkepäin. Sovelluksen kehityksessä on käytetty ketterän sovelluskehityksen mallia. Asiakkaan toiveita on pyritty toteuttamaan mahdollisimman tarkasti, kuitenkin luopumatta kustannustehokkuudesta.

Työvuorosuunnittelusovelluksen tarkoituksena on lyhentää, aikaisemmin täysin käsin tehtyä, työvuorosuunnitteluprosessia. Työvuorosuunnittelu vie kahdelta hoitajalta 24 tuntia työaikaa viikossa. Ruokailun tekeminen manuaalisesti vie noin kuusi tuntia viikossa. Työvuorosuunnittelun prosessin automatisoinnilla on muutamia hyötyjä:

- Tehokkuuden lisääntyminen
- Palvelun laadun parantuminen
- Kustannuksien pienentyminen

(Mikkelin kaupunki 2013.)

Aikaisemmin hoitajat joutuivat täyttämään joka viikko viikon alussa työvuorosuunnittelulistoja, joihin he kokosivat hoitoaikoja, jotka lasten vanhemmat ilmoittivat sähköpostilla tai puhelimitse. Näiden hoitoaikojen perusteella hoitajat päättelivät kuinka monta henkilöä kunakin päivänä on syömässä kullakin aterialla, ja kuinka monta hoitajaa milloinkin pitäisi olla töissä, milloin vuorot ovat ja kuka hoitaa minkäkin vuoron. Hoitajien piti vielä ottaa huomioon lasten kertoimet, ettei lakisääteiset hoitorajat ylity ja laskea kuinka monta lasta saa olla per hoitaja. (Kuva 9.)

MantisBT on avoimen lähdekoodin *issuetracker* -ohjelmisto, jolla seurataan sovelluksen kehitystä. MantisBT:llä on helppo seurata ohjelmointivirheitä. Käyttäjät voivat luoda omia *tikettejä*, joille voi asettaa erilaisia ominaisuuksia. Etenemissuunnitelmasta (roadmap) näkee miten sovelluskehitys on edennyt ja mitä *tikettejä* on vielä avoimena ja mitkä ovat suljettuja. MantisBT on projektissa käytössä koko projektin ajan. (Kuva 10.)



Olet sisäänkirjautuneena käyttäjätunnuksella: *aulis* (Aulis Fonkanen - pääkäyttäjä)

[Oma näkymä](#) | [Näytä tapahtumat](#) | [Lisää tapahtuma](#) | [Muutosloki](#) | [Etenemissuunnitelma](#) | [Yhteenveto](#) | [Hallinta](#) | [Käyttäjätiliini](#) | [Kirjaudu ulos](#)

Työntekijätoiminnot - Muutosloki

[Työntekijätoiminnot - 1.1.0](#) (Ei vielä julkaistu) [[Näytä tapahtumat](#)]

Löydetyt puutteet tuotannossa

- [0000541](#): [Työvuorot] Ryhmanäkymään viikonpäivävalittisin ([otto](#)) - suljettu.
- [0000542](#): [Työvuorot] Aloitus ja lopetus aika näkyviin lyhyisiin varauksiin ([otto](#)) - suljettu.
- [0000543](#): [Viikkonäkymä] Varauksen monistaminen saman viikon mülle päiville ([otto](#)) - suljettu.
- [0000544](#): [Työvuorot] Vierailijan tahti häviää jos siirretään toiseen ryhmään ([otto](#)) - suljettu.
- [0000545](#): [Päivänäkymä] Viikkonumero näkymään myös päivänäkymään ([otto](#)) - suljettu.
- [0000546](#): [Työvuorot] Väliaikaisien varauksien lapset näkyvät kaikilla päivillä ryhmänäkymässä vaikka varauksia ei ole kuin osalle päivistä ([otto](#)) - suljettu.
- [0000547](#): [Työvuorot] Ryhmanäkymän ryhmä ei säily jos siirtyy seuraavalle viikolle ([otto](#)) - suljettu.
- [0000548](#): [Työvuorot] Yli 3v viiva häviää jos ruksaa "Näytä vain ajanvaraukset" ([otto](#)) - suljettu.
- [0000549](#): [Päivänäkymä] Vierailijoiden väliaikaiset varaukset näkyvät siirretyissä ryhmässä väärin ([aulis](#)) - suljettu.
- [0000548](#): [Viikkonäkymä] Päivämäärän/viikon säilyttäminen näkymien välillä ([aulis](#)) - suljettu.
- [0000546](#): [Työvuorot] Pystyviivat tulosteisiin ([aulis](#)) - suljettu.
- [0000544](#): [Työvuorot] Työvuoroehdotuksen alkuun päivämäärä ja viikonpäivä ([aulis](#)) - suljettu.
- [0000549](#): [Työvuorot] Viikonpäivä tulosteisiin päivämäärän lisäksi ([aulis](#)) - suljettu.
- [0000545](#): [Työvuorot] Työvuorolaskenta tarjoaa liian vähän resursseja illaksi ([aulis](#)) - suljettu.
- [0000543](#): [Viikkonäkymä] Hoitoajan aloitusajan minuuttien pyöritys alaspäin ([aulis](#)) - suljettu.
- [0000543](#): [Viikkonäkymä] Hoitoajan lopetusajan minuuttien pyöritys ylöspäin ([aulis](#)) - suljettu.
- [0000543](#): [Työvuorot] Ryhmanäkymän valitse ryhmä -pudotusvalikon sijoittelu ([aulis](#)) - suljettu.
- [0000491](#): [Päivänäkymä] Lisättyjen vierailijoiden varausten siirtäminen ei toimi ongelmitta ([aulis](#)) - suljettu.

[18 tapahtumaa]

[Työntekijätoiminnot - 1.0.0](#) (Julkaistu 2014-03-14) [[Näytä tapahtumat](#)]

Tuotantoversio

- [0000527](#): [Työvuorot] Kulmien pyöritykset otsikkotunneista ([aulis](#)) - suljettu.
- [0000508](#): [Viikkonäkymä] Vierailijoiden tyylit ([aulis](#)) - suljettu.
- [0000424](#): [Päivänäkymä] Satunnaiset checkboxien automaattiset valinnat ([aulis](#)) - suljettu.
- [0000509](#): [Työvuorot] Vierailijoiden resurssit ([aulis](#)) - suljettu.
- [0000505](#): [Viikkonäkymä] Vierailijan lisäys ([aulis](#)) - suljettu.
- [0000501](#): [Päivänäkymä] Varausten siirtäminen takaisin alkuperäiseen ryhmään luo uuden varauksen alkuperäiseen ryhmään ([aulis](#)) - suljettu.
- [0000509](#): [Päivänäkymä] Usean päivän varauksen siirto ei toimi oikein ([aulis](#)) - suljettu.
- [0000498](#): [Viikkonäkymä] Kaksoispisteiden kirjoittaminen aikakenttään ([aulis](#)) - suljettu.
- [0000497](#): [Viikkonäkymä] Noutotiedot eivät näy varauksella ([aulis](#)) - suljettu.
- [0000499](#): [Viikkonäkymä] Usean varauksen tabindeksit väärin ([aulis](#)) - suljettu.
- [0000493](#): [Päivänäkymä] Siirrettävien varausten poisto/lisäys checkboxien avulla ([aulis](#)) - suljettu.
- [0000219](#): [Päivänäkymä] Lomakkeiden validoinnit ([aulis](#)) - suljettu.
- [0000209](#): [Päivänäkymä] Varausten siirron ongelmat ([aulis](#)) - suljettu.
- [0000214](#): [Viikkonäkymä] Lisää vierailija dialogista puuttuu mahdollisuus lisätä lapsia ([aulis](#)) - suljettu.
- [0000465](#): [Lapset ja ryhmät] Lapset nimen mukaan järjestykseen ([aulis](#)) - suljettu.
- [0000427](#): [Viikkonäkymä] Hoitoajan syöttölomakkeelle kellonaikojen syöttöön kaksoispiste ([aulis](#)) - suljettu.
- [0000443](#): [Päivänäkymä] Virheilmoitus päivänäkymäisivulla ([aulis](#)) - suljettu.

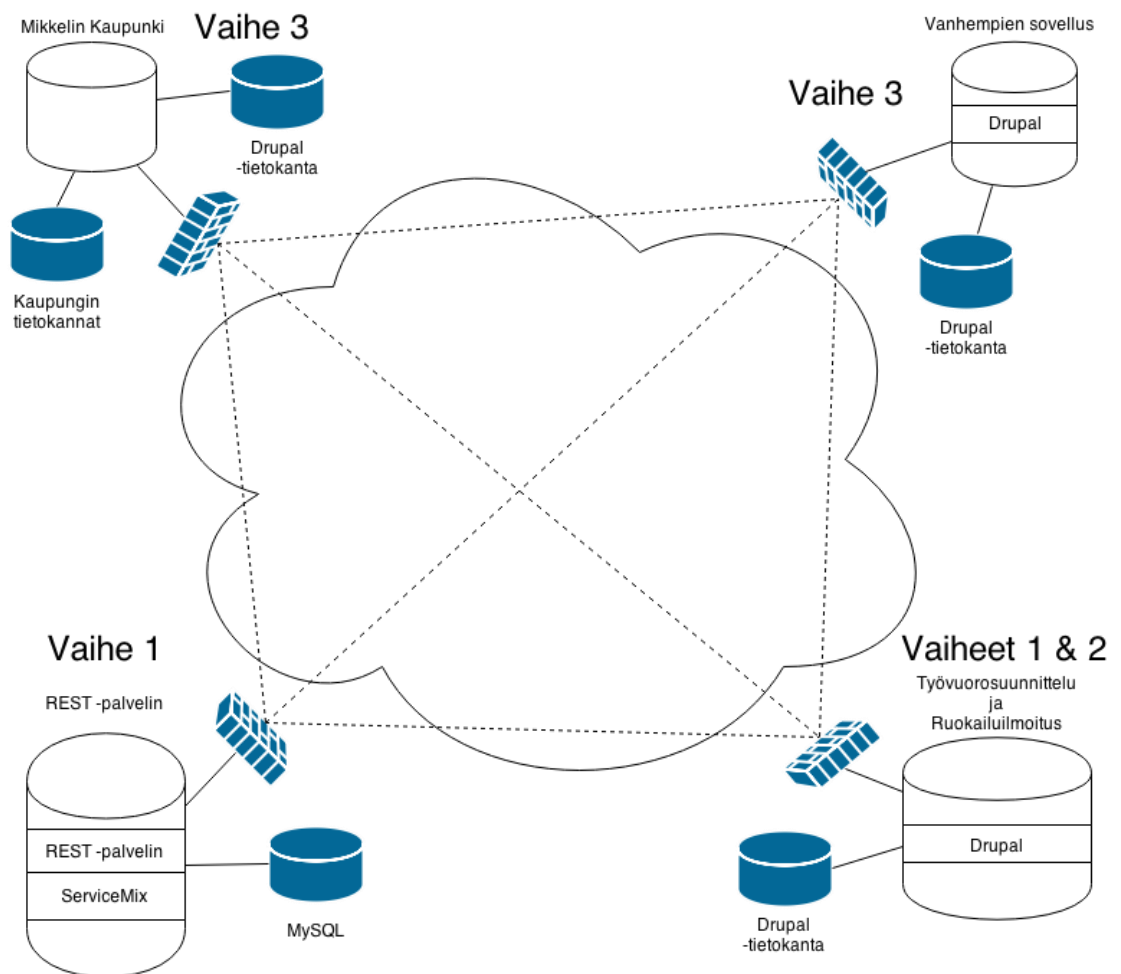
KUVA 10. MantisBT muutosloki

Apache ServiceMix:n komponenteista Apache ActiveMQ:tä ja Apache CXF:ää ei tarvita työvuorosuunnittelusovelluksessa, sillä Apache Camel käyttää *restlet* -komponenttia tarvittavien pyyntöjen reititykseen. Usein *enterprise* -tason järjestelmässä tarvitaan kaikki Apache ServiceMix:n komponentteja ja myös useita Apache Camel:n komponentteja. Tällaiset järjestelmät vaativat paljon suunnittelua, sillä integroitavien järjestelmiä ja Camel -reittejä on paljon. Jo järjestelmän ylläpito on hankalaa, vaikka se on tehty helpoksi Apache ServiceMix:ssä.

4.1 Arkkitehtuuri

Tätä opinnäytetyötä kirjoittaessa ei ole vielä varmaa, että miten palvelut sijoitetaan fyysisesti. Palveluja voidaan myöhemmässä vaiheessa vielä yhdistää tai hajauttaa, joten arkkitehtuurikuvaus ei välttämättä ole täysin tarkka. Mikkelin kaupungin yhteistyö palveluiden ja tietokantojen sijoittamisen suhteen on vielä kysymysmerkki.

Ensimmäisen vaiheen työvuorosunnitteluovellus on tarkoitettu Vilttihatulla töissä oleville hoitajille. Tarkoitus on, että he tekevät työvuorosunnittelunsa sovelluksella. Kuitenkin aluksi hoitajat syöttävät lasten hoitoajat vanhempien sijaan. Työvuorosunnitteluovelluksen arkkitehtuuriin kuuluu Drupalin päälle tehty *front-end* ja Apache ServiceMix:llä toimiva palvelinpää. Toisen vaiheen ruokailuilmotusovellus on myös oma Drupal -moduulinsa ja todennäköisesti sijoitetaan samaan Drupaliin vaiheen yksi kanssa. (Kuva 11.)



KUVA 11. Koko projektin arkkitehtuuri vaiheet 1, 2 ja 3

Vaiheen kolme sovellus tulee fyysisesti sijaitsemaan joko Mikkelin kaupungin palvelimilla tai jos Mikkelin kaupunki haluaa luopua ylläpitovelvoitteesta, voidaan sovellus sijoittaa Smarttime:n osoittamille palvelimille. Vaiheen kolme hoitoaikojen ilmoitussovelluksen arkkitehtuuriin myös kuuluu Drupalilla tehty *front-end*, mutta palvelinpään logiikka ei ole vielä tiedossa. Voi olla, että myös ilmoitussovelluksen palvelinpäätä hoitaa Apache ServiceMix, mutta onko se sama instanssi kuin ensimmäisen ja toisen vaiheen sovelluksilla on epävarmaa. (Kuva 11.)

4.2 Tietokanta

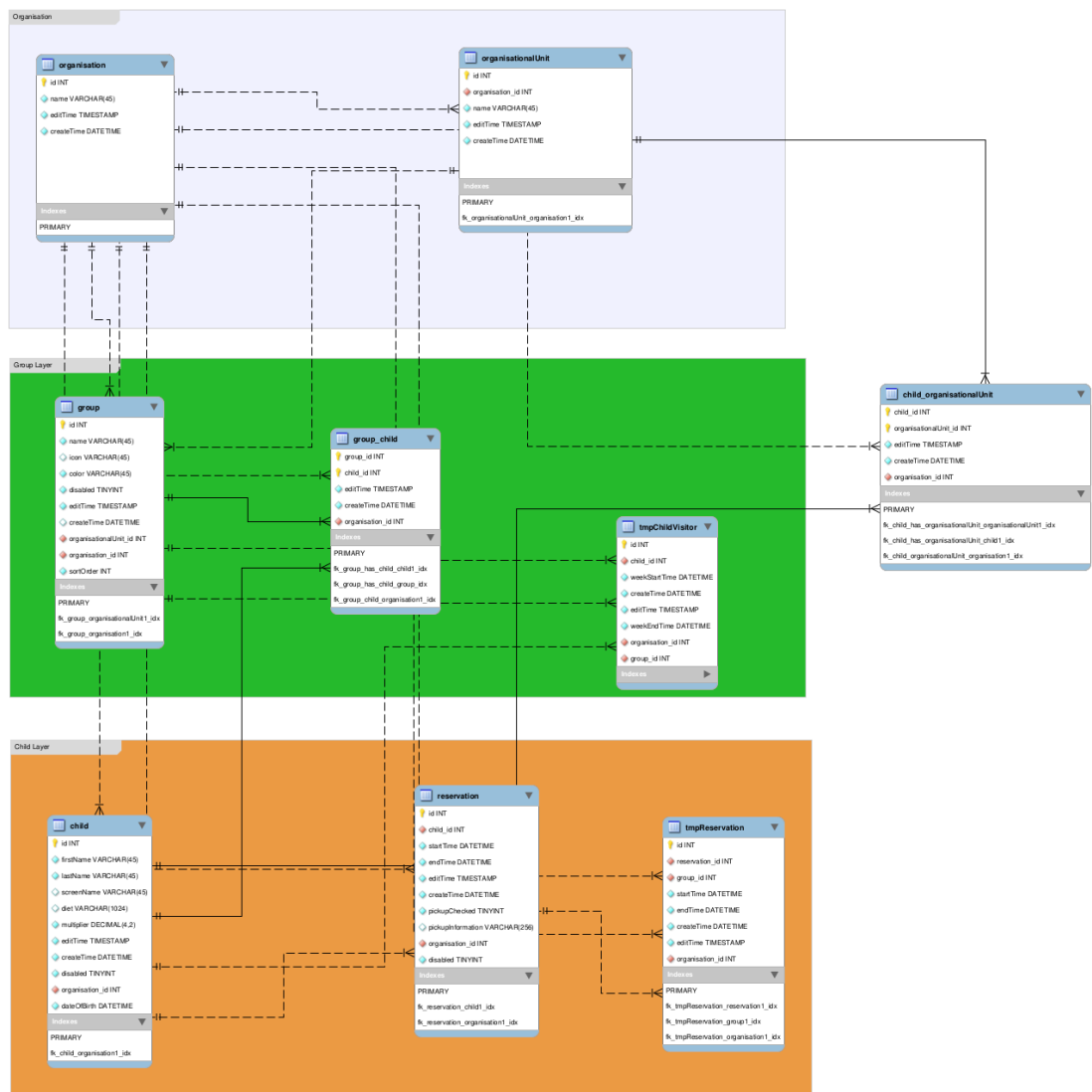
Työvuorosuunnittelusovelluksen ensimmäisen vaiheen tietokannan suunnittelu aloitettiin samaan aikaan käyttöliittymien kanssa. Tietokannaksi valittiin MySQL -tietokanta, koska Smart Time Oy:llä on paljon kokemusta juuri MySQL -tietokannasta ja se on helppo ja nopea ottaa käyttöön. Myös siksi, että MySQL on yleinen valinta Drupal -tietokannaksi. Todennäköisesti myös muut tietokannat tässä projektissa tulevat olemaan MySQL -tietokantoja.

Tietokannassa säilytetään tietoja mm. päiväkodeista, lapsista ja varauksista. Tietokanta sisältää kaikki tarvittavat tiedot, jotta lasten varaukset ja hoitajien työvuorot olisivat helposti hallittavissa. Hoitajien työvuoroja ei säilytetä tietokannassa, vaan ne laskeetaan aina erikseen lasten hoitoaikojen mukaan. Tietokanta on suunniteltu skaalautuvaksi sillä tavalla, että uusia tietokantoja tai tauluja ei tarvitse perustaa vaikka sovellusta käyttävien päiväkotien määrä lisääntyisi. Pienet muutokset voivat olla silti mahdollisia riippuen päiväkotien tarvitsemista toiminnoista.

Organisation -taulussa olevat tietueet ovat kokonaisuuksia, jotka koostuvat yhdestä tai useasta päiväkodista. Melkein jokaisessa muussa taulussa on linkitys *Organisation* -tauluun, jotta SQL -kyselyt olisivat yksinkertaisempia. *OrganisationalUnit* -taulussa on perustiedot kustakin päiväkodista. Yksi päiväkoti kuuluu aina yhteen organisaatioon. *Child* -taulu sisältää perustiedot kustakin lapsesta. Yksilöivänä avaimena on *id* -kenttä, jotta jokaisen lapsen voisi erottaa toisista. Jokainen lapsitietue kuuluu johonkin yhteen organisaatioon, mutta voi kuulua useampaan päiväkotiin. *Child_organisationalUnit* -taululla mahdollistetaan, että kukin lapsi voi kuulua useampaan päiväkotiin. Tämä on hyödyllistä siinä tapauksessa, jos lapsi on useammassa päiväkodissa, niin ei tarvitse luoda uutta lapsitietuetta, jolloin yksikään lapsi ei ole

kahtena tietueena tietokannassa. Poikkeuksena on, jos lapsi kuuluu kahteen organisaatioon, niin tietokannan näkökulmasta sellaiset tietueet ovat eri tietueita. (Kuva 12.)

Group -taulussa on perustiedot kustakin ryhmästä. Lapsitietueet on jaettu ryhmiin, joissa niitä on helppo hallita yhtenä kokonaisuutena. Ryhmärakenne myös kuvastaa täsmälleen päiväkodin ryhmärakennetta. *Group_Child* -taulun tarkoitus on sallia lapsen kuulumiseen useampaan ryhmään kerralla. Tämä ei ole mahdollista Vilttitahulle toteutettavissa sovelluksissa, mutta voi olla mahdollista muissa päiväkodeissa tai organisaatioissa. *TmpChildVisitor* -taulussa on tallennettu vierailijat lapset. Lapsitietueisiin on viitattu yksilöivän kentän avulla, jotta lapsi olisi tunnistettavissa. Taulusta löytyy myös tieto mihin ryhmään lapsi on asetettu vierailemaan ja mille viikolle. (Kuva 12.)



KUVA 12. Tietokanta kokonaisuudessaan

Reservation -taulussa ovat tiedot varauksista (ei väliaikaisista). Varaukset kertovat, millaisia hoitoaikoja lapsille on ilmoitettu. Lapsella voi olla enemmän kuin yksi varaus, mutta ei päällekkäin. Ensimmäisessä vaiheessa projektia hoitajat syöttävät itse varaukset lapsille. Kolmannessa vaiheessa tämä muuttuu siten, että lasten vanhemmat ilmoittavat itse lastensa hoitoajat. Vanhemmilla tulee olemaan omat tunnukset kolmannen vaiheen hoitoaikojen ilmoitussovellukseen. Tunnukset sidotaan lasten tietoihin, jotta vanhemmat voivat ilmoittaa vain omien lastensa hoitoajat. *TmpReservation* -taulussa ovat väliaikaisten varauksien tiedot. Väliaikaisia varauksia voi vain tehdä varsinaisten varauksien ”päälle”. Väliaikaiset varaukset ovatkin tavallaan siirrettyjä varauksia, joiden tiedot poistetaan jos alkuperäinen varaus poistetaan tai sitä muokataan. (Kuva 12.)

4.3 Näkymät

Työvuorosuunnittelusovellus koostuu neljästä päänäkymästä. Jokainen päänäkymä on suunniteltu toteuttamaan erilaisia toiminnallisia ominaisuuksia, jotka yhdessä muodostavat työvuorosuunnittelusovelluksen kokonaisuuden. Päänäkymät ovat:

- Viikkonäkymä
- Päivänäkymä
- Lapset ja ryhmät -näky
- Työvuorosuunnittelu -näky

Viikkonäkymässä käyttäjä pääsee tarkastelemaan lasten hoitoaikoja kokonaisuudessaan. Päivänäkymässä käyttäjä voi tarkastella lasten hoitoaikoja päivä kerrallaan ja luoda uusia väliaikaisia varauksia. Lapset ja ryhmät -näkyssä käyttäjä voi lisätä, muokata ja poistaa lapsitietueita. Työvuorosuunnittelu -näkyssä käyttäjä pääsee tekemään työvuorosuunnittelua.

4.3.1 Viikkonäkymä

Viikkonäkymässä näkyy kaikki ryhmät ja kaikkien ryhmien lapset. Vierailijat merkitään tähdellä. Viikkonäkymän varaustaulukossa näkyy kaikkien lasten varaukset kyseiselle viikolle. Varaukset näkyvät kellonaikoina päivälle, jolle varaukset on tehty.

Yhdellä lapsella voi olla monia varauksia yhdelle päivälle ja varaukset voivat ulottua päivältä toiselle tai jopa viikolta toiselle. Klikkaamalla varaustaulukosta jonkun päivän kohdalta avautuu dialogilaatikko, jossa voi tehdä uusia varauksia. Dialogissa näkyy uuden varauksen luomisen lisäksi kaikki varaukset, jotka ovat kyseisellä lapsella kyseisenä päivänä. Vanhoja varauksia voi muokata. Viikkonäkymässä näkyvät vain alkuperäiset varaukset, ei väliaikaisia varauksia. Viikkonäkymän tarkoitus on tarkastella, luoda uusia ja muokata varauksia. Tämän ominaisuuden tulee osittain korvata toisessa vaiheessa toteutettava sovellus, johon vanhemmat syöttävät lastensa hoitoajat hoitajien sijaan. Ryhmän vasemmassa alakulmassa on ”Lisää vierailija” -linkki, jota klikkaamalla avautuu dialogilaatikko, jossa voi hakea vierailija -ryhmässä olevia lapsia nimellä ja lisätä heitä vierailijoiksi viikko kerrallaan. Jos lasta ei löydy, voidaan uusi lapsitietue luoda samalta dialogilta. (Kuva 13.)

Vuoropäiväkoti
Vilttihattu

Kirjautuneena: smarttime | Kirjautu ufos

Etusivu Hoitokalenteri Lapsen ja ryhmät Työvuorot

Aurinkohatut Tänään 31.03.2014

	Ma 31.3.	Ti 1.4.	Ke 2.4.	To 3.4.	Pe 4.4.	La 5.4.	Su 6.4.
Aamu T. Yli 3v.	07:30-16:30	07:30-14:30	07:30-16:30	07:30-16:30	07:30-14:30		
Aino K.	05:15-15:00	05:15-15:30	05:15-15:00	12:00-22:30	12:00-22:30		
Alina S.	09:30-18:30						
Bella-Donna N.							
Eevi U.	08:30-15:30				13:30-17:30	10:00-16:00	
Emilia M.	10:00-15:45	13:30-17:00	10:00-15:45			10:30-18:45	11:30-17:30
Hilma L.							
Ida M.	13:15-14:45	13:15-22:30	13:15-22:30	20:45-22:30	20:45-22:30		
Janna K.	06:45-15:45	12:00-15:45	12:00-22:15		06:45-15:30	06:45-14:30	
Jesper V.	11:00-20:30	10:00-19:00	11:00-17:00	11:00-18:00	11:45-21:30		
Jimi M.	07:30-15:30	07:30-15:30	07:30-15:30	06:30-15:00	06:30-15:00		
Jooany	08:00-16:30	11:30-20:30		11:30-20:30	06:00-15:00		
Joonatan V.		12:00-22:00	06:30-14:30	12:00-22:00	06:30-14:30		
Kaapo P.	08:30-15:30	08:30-15:30	08:30-15:30	08:30-15:30	08:30-15:30		
Kasper K.							
Kira P.		08:00-17:00	08:00-17:00	07:30-15:30	07:30-15:30		
Luca I.	13:00-21:45	07:00-15:45	07:00-15:45		13:00-21:45	13:00-21:45	13:00-21:45
Lukapa	08:00-16:30	08:00-16:30	08:00-16:30	08:00-16:30	08:00-16:00		
Lukas P.	06:30-15:30	12:30-21:30	06:30-15:30		13:30-21:30	06:30-14:30	
Lumia A.	00:00-14:00 17:00-24:00	00:00-14:00					
Mette A.	00:00-14:00 17:00-24:00	00:00-14:00					
Mico K.							
Minea A.	00:00-14:00 17:00-24:00	00:00-14:00					
Miro V.	08:30-18:30	13:45-20:45	13:30-20:30	13:45-20:45			

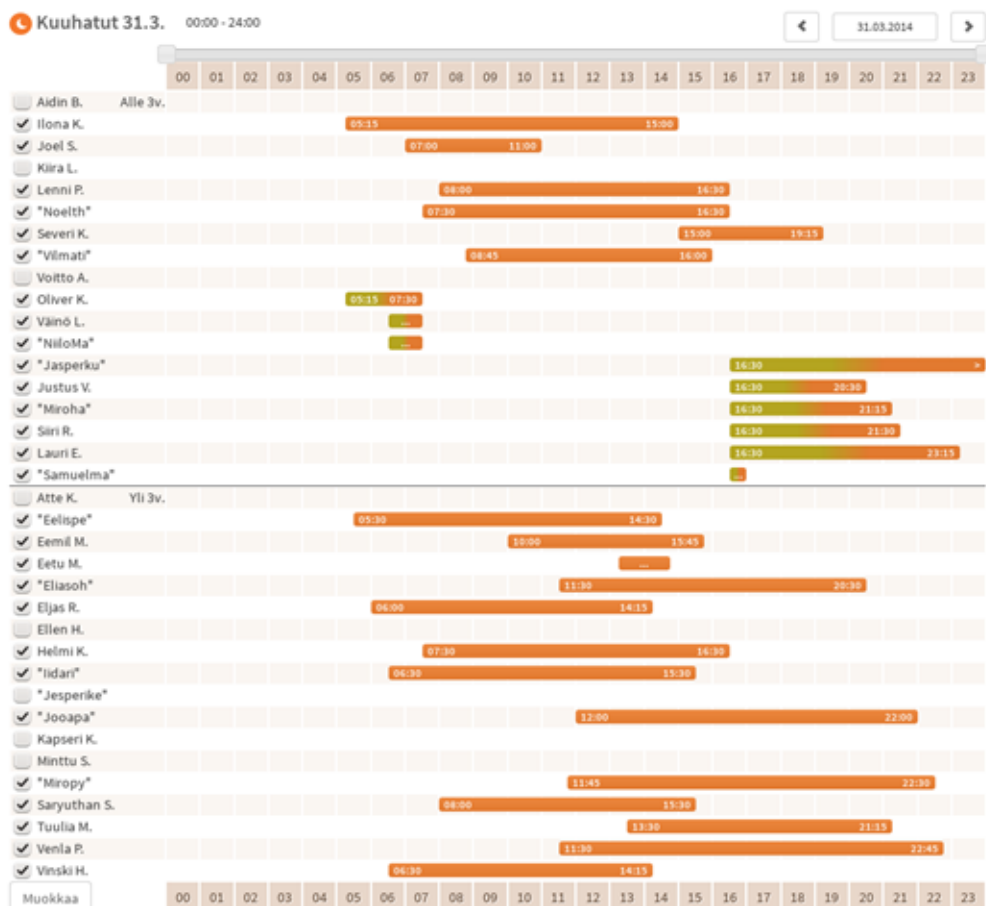
KUVA 13. Osittainen viikkonäkymä

Ryhmiä voidaan supistaa, jotta näytöllä näkyisi muiden kuin ensimmäisten ryhmien tiedot, klikkaamalla ryhmän otsikkoa. Samalla tavalla ryhmä voidaan myös laajentaa.

Kaikki ryhmät ovat oletuksena laajennetussa tilassa. Ohjelma muistaa evästeiden avulla, mitkä ryhmät ovat supistettuja ja mitkä ryhmät ovat laajennettuja. Oikeassa reunassa on kalenteri, josta voi valita päivän. Kun päivä valitaan, sivu päivittyy ja näyttää sen viikon varaukset.

4.3.2 Päivänäkymä

Päivänäkymässä näkyy kaikkien ryhmien lapset ja heidän varauksensa yhdelle tietylle päivälle. Ryhmät ja varaukset ovat värikoodattuja, jotta käyttäjän olisi helppo havaita nopeasti, mihin ryhmään varaukset kuuluvat. Lasten varaukset näkyvät varaustaulukossa piirrettyinä varauspalkkeina, joiden sisällä lukee kyseisen varauksen alku- ja loppumisaika. Jos varaus on niin lyhyt, että varauksen alkamis- ja päättymisaika eivät mahdu hyvin varauspalkin sisään, kyseisen varaukset ajat korvataan kolmella pisteellä (...). Jokaisen ryhmän varaustaulukon yläpuolella on liukusäädin, jota liu'uttamalla voidaan valita varauksia siirrettäväksi. Jokaisessa liukusäätimessä on kaksi säädintä, joilla kellonaikaväliä muutetaan. Kellonaikaväli on aina ryhmäkohtainen. (Kuva 14.)



KUVA 14. Päivänäkymä yhden ryhmän osalta

Jokaisen lapsen nimen edessä on valintaruutu. Tämän valintaruudun valitseminen tarkoittaa, että kyseisen lapsen varaukset, joiden alkamis- tai loppumisaika sopii valittuun kellonaikaväliin, valitaan siirrettäväksi. Sovellus valitsee automaattisesti kaikkien lasten valintaruudut, joiden yksi tai useampi varaus on liukusäätimen kellonaikavälin alueella. Jos ei haluta siirtää jonkun tietyn lapsen varauksia, valintaruutu on käyttäjän itse valittava pois. Muokkaa -painiketta painettaessa aukeaa dialogilaatikko, jossa käyttäjä voi valita ryhmän (paitsi ryhmän, josta varauksia siirretään), johon valitut varaukset siirretään. Hyväksymällä valinnan sovellus siirtää varaukset valittuun ryhmään, jolloin alkuperäisessä ryhmässä siirretyt varaukset merkataan liukuvärillä siten, että alkuperäisen ryhmän väritunnus on aina vasemmalla puolella ja ryhmän, johon varaukset tai varaus on siirretty, väritunnus on aina oikealla puolella.

4.3.3 Lapset ja ryhmät -näky

Lapset ja ryhmät -näkyssä näkyy kaikkien ryhmien (myös vierailijoiden) lapset ryhmittäin. Lapset ja ryhmät -näkyksen tarkoitus on tarkastella ja muokata lapsien perustietoja sekä luoda uusia lapsitietueita. Ryhmät on värikoodattu ryhmien tunnuskoodien mukaan. Jokaisen ryhmäotsikon alta löytyy ”Lisää uusi lapsi” -painike, jota klikkaamalla näkyksen päälle ilmestyy dialogilaatikko, jonka sisällä olevalla lomakkeella lähetetään tietoja uudesta lapsitietueesta. (Kuva 15.)

Vuoropäiväkoti
Vilttihattu

Kirjautuneena: smarttime | Kirjautu ulos

Etusivu Hoitokalenteri Lapset ja ryhmät Työvuorot

Aurinkohatut	Kuuhatut	Piltilhatut	Sädehatut	Tähtihatut
Lisää lapsi	Lisää lapsi	Lisää lapsi	Lisää lapsi	Lisää lapsi
Aamu T. Yli 3v.	Aidin B. Alle 3v.	Aapo N. Alle 3v.	Eveliina S. Alle 3v.	Aaro H. Yli 3v.
Aino K.	Ilona K.	Atem K.	Aino K. Yli 3v.	Aida P.
Alina S.	Joel S.	"Eelisha"	Anni H.	Daniel I.
Bella-Donna N.	Kiira L.	"Eliasse"	Elli K.	Daniel P.
Eevi U.	Lenni P.	"Jasperku"	Enni J.	Eino U.
Emilia M.	"Noelth"	Joel K.	Helmi S.	Johan P.
Hilma L.	Severi K.	"Jonnima"	Jenna V.	Jonne E.
Ida M.	"Vilmati"	Justus V.	"Jimie"	Lukas A.
Janna K.	Voitto A.	Konsta K.	Joseph R.	Minttu P.
Jesper V.	Atte K. Yli 3v.	Laura K.	Leo T.	Nicole P.
Jimi M.	"Eelispä"	Lauri E.	Lilja T.	Niko L.
"Jooany"	Eemil M.	Lenni K.	"Luukasoh"	Ria O.
Joonatan V.	Eetu M.	Lotta K.	Milo E.	Sameela I.
Kaapo P.	"Eliasoh"	Max H.	Otto O.	Tiuhti K.
Kasper K.	Eljas R.	Milja T.	Tytti K.	Topi P.
Kira P.	Ellen H.	"Miroha"	"Veetihä"	"Viivisä"
Luca I.	Helmi K.	Netta N.	Venla P.	
"Lukapa"	"Iidari"	Niilo J.	"Vilmahy"	
Lukas P.	"Jesperike"	"NiiloMa"		
Lumia A.	"Jooapa"	Oliver K.		
Mette A.	Kapseri K.	Onni L.		
Mico K.	Minttu S.	Peetu S.		
Minea A.	"Miropy"	"Samuelma"		
Miro V.	Saryuthan S.	Samuel T.		
Nea N.	Tuulia M.	Siiri R.		
Niklas M.	Venla P.	Urho A.		
Ninni K.	Vinski H.	Väino L.		
Noel H.				

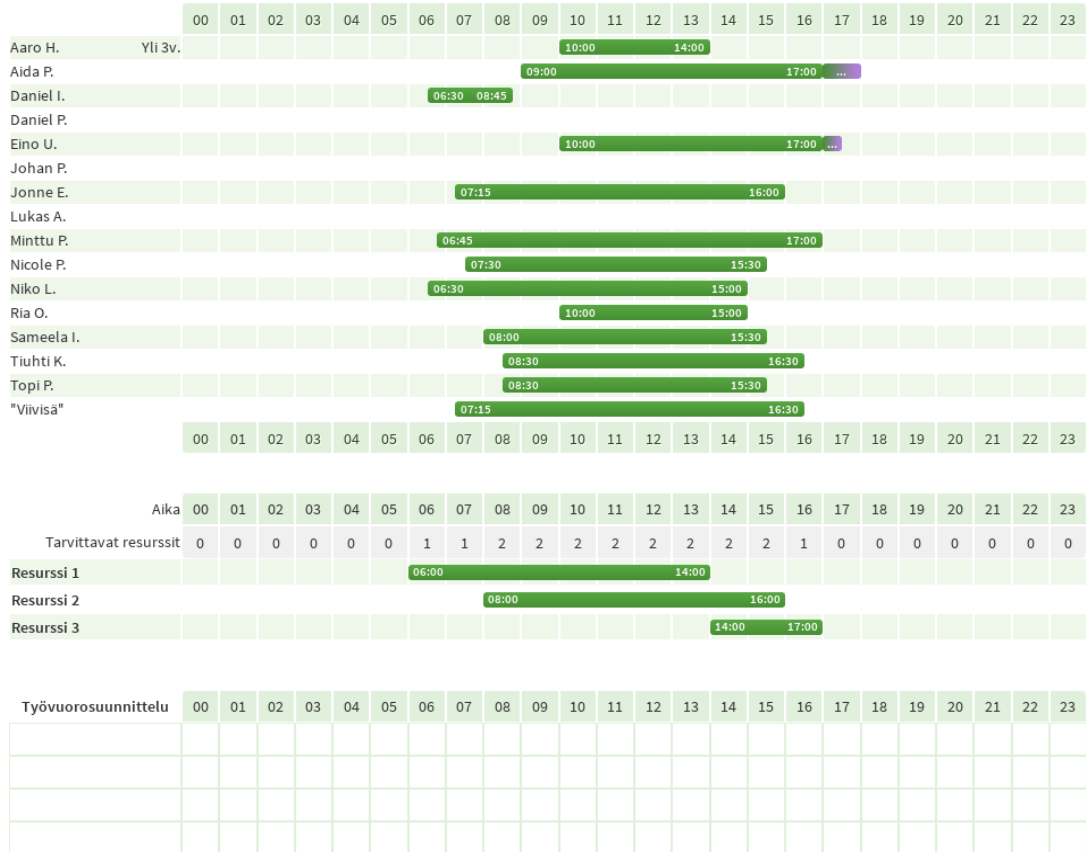
KUVA 15. Lapset ja ryhmät -näkömä

Klikkaamalla lapsen nimeä aukeaa dialogilaatikko, jossa pääsee muokkaamaan lapsen perustietoja kuten nimeä, syntymäaika, kerrointa ja ruokavaliota. Lapsen kerroin vaikuttaa resurssitarpeeseen. Mitä suurempi kerroin sitä enemmän hoitajia tarvitsee olla töissä.

4.3.4 Työvuorosunnittelu -näkömä

Työvuorosunnittelu -näkömässä näkyy oletuksena kaikkien ryhmien kaikki lapset ja heidän varaukset yhdelle viikolle. Työvuorosunnittelu -näkömässä näkyy päivänäkymän kaltaisesti kaikkien lasten varaukset varaustaulukossa. Varaustaulukon jälkeen näkyy resurssitaulukko, jossa on sovelluksen laskemat ehdotukset hoitajien työajoista. Jokainen resurssi tarkoittaa yhden hoitajan yhtä työvuoroa. Resurssitaulukon jälkeen näkyy tyhjä työvuorosunnittelutaulukko, johon hoitajat merkitsevät sovelluksen laskemien ehdotuksien muokkaukset tulostettuaan taulukot. (Kuva 16.)

★ Tähtihatut 4.4.



KUVA 16. Työvuorosunnittelu -näköymä

Työvuorosunnittelu -näköymän yläosassa on ryhmä valitsimia, joita valitsemalla saadaan aikaan erilaisen näköinen työvuorosunnittelu -näköymä. ”Koko viikko” -valitsin valittuna näkyy koko viikon kaiken ryhmien varaukset ellei ryhmävalitsinta ole valittu. Jos ryhmävalitsin on valittu jollekin ryhmälle, näköymä näyttää vain sen ryhmän varaukset. Päivänäkymä -valitsin toimii samalla tavalla. Kun se on valittuna, näkyy vain valitun päivän varaukset ja jos ryhmä on valittuna, näkyy vain sen ryhmän varaukset. (Kuva 17.)

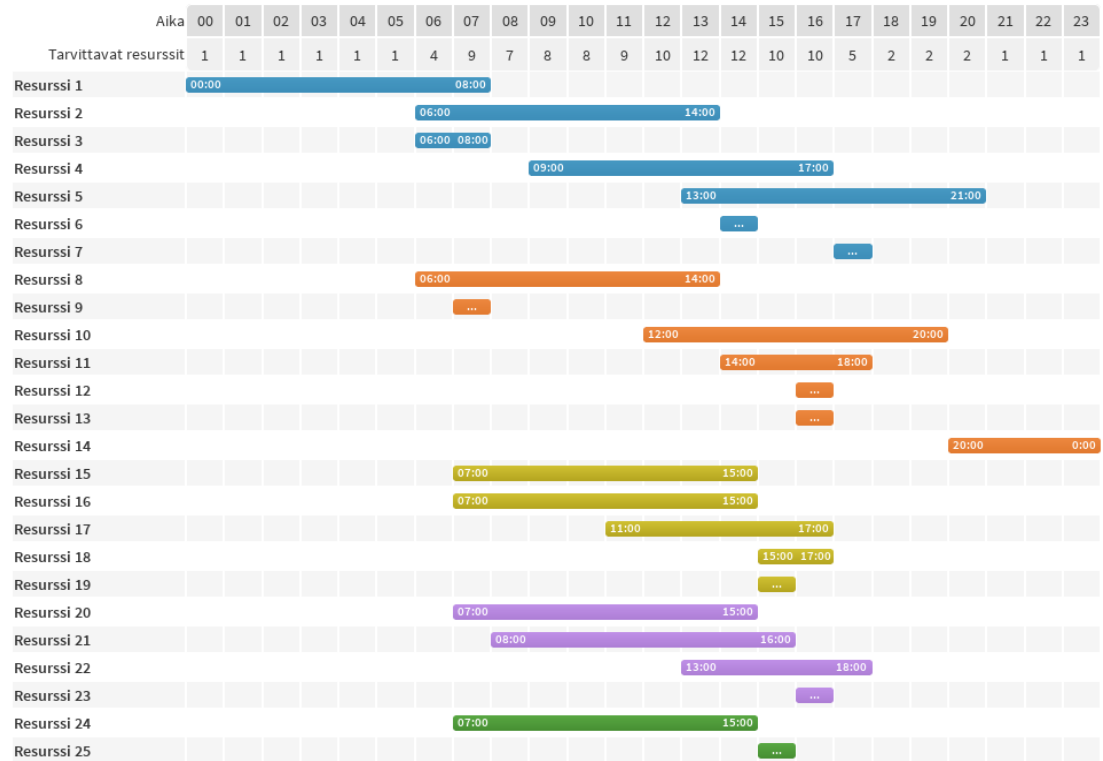


KUVA 17. Työvuorosunnittelu -näköymän valitsimet

Valitsinvalikossa on myös Kaikki resurssit -painike, jolla saadaan näkyviin koko päiväkodin resurssitarve, jonka perusteella hoitajat pystyvät luomaan väliaikaisia varauk-

sia, jotta resurssitarvetta saataisiin pienemmäksi. Mitä pienempi resurssitarve päiväkodilla on, sitä vähemmän hoitajia tarvitsee olla töissä. (Kuva 18.)

Pe 2.5.



KUVA 18. Kaikki resurssit

Tulosta tämä -valitsinta painamalla selain aukaisee uuden ikkunan, johon sovellus piirtää tulostettavan version näkymästä. Tulosteen ja sovelluksen näkymän välillä on hieman eroja, johtuen tulosteiden tyylimäärityksistä (kaikkia tyylejä ja asetteluja ei voida tulostaa). Tulosteen tarkoitus on, että hoitajat voivat sovelluksen käytön rinnalla suorittaa työvuorosuunnittelua käsin piirtämällä vuoronsa tyhjiin työvuorosuunnittelu- taulukkoon. Tämä tehdään siksi, että sovellus ei pysty täysin arvioimaan minkälaisia vuoroja hoitajat voivat/haluavat tehdä.

4.4 REST -palvelin

REST -palvelin kuuntelee HTTP -pyyntöjä, joita REST -asiakasohjelmisto lähettää. Työvuorosuunnitteluovelluksen tapauksessa Apache Camel käyttää *restlet* -komponenttia kuunnellakseen asiakasohjelmiston lähettämiä HTTP -pyyntöjä. Työvuorosuunnittelun tapauksessa Camel -reitit ovat konfiguroitu käyttämällä Spring XML -konfiguraatiota. Tämä tarkoittaa, että reitityssäännöt kirjoitetaan XML -

tiedostoon, josta Camel:n käynnistettäessä käy lukemassa asetettujen reittien tiedot. Nämä reitityssäännöt sijaitsevat *CamelContext* -tiedostossa. Yksi reitti on konfiguroitu yhtä toiminnallisuutta varten. Esim. lapsen lisäys menee eri reittiä kuin lasten tietojen hakeminen. (Kuva 19.)

```

<route id="getChild">
  <from ref="getChildRest" />
  <bean ref="DCPrest" method="getChild" />
  <marshal ref="child" />
</route>

<route id="addChild">
  <from ref="addChildRest" />
  <doTry>
    <unmarshal ref="child" />
    <bean ref="DCPrest" method="addChild" />
    <marshal ref="childResponse" />
    <doCatch>
      <exception>java.lang.Exception</exception>
      <setHeader headerName="subject">
        <constant>Add child error</constant>
      </setHeader>
      <setBody>
        <simple>Virhe viesti: ${exception.message} XML: ${bodyAs(String)}
        </simple>
      </setBody>
      <to uri="mock:finally" />
      <log message="${exception.message} " />
    </doCatch>
  </doTry>
</route>

```

KUVA 19. Kaksi Apache Camel -reittiä työvuorosuunnittelusovelluksesta

Apache Camel *marshaloi* JSON -tyyppisen viestin POJO –olioon (Plain Old Java Object), jolloin lähetetty data on REST -pavun käytössä (kuva 20). REST -papu suorittaa pyynnön mukaiset toiminnot tietokantaan (lisäys, luku, päivitys tai poisto) ja lähettää vastauksen REST -asiakasohjelmistolle. Sen jälkeen REST -asiakasohjelmisto vastaanottaa vastauksen ja suorittaa tarvittavat toimenpiteet näkymien ja muun *front-end* -logiikan päivittämiseksi.

```

1 package fi.smarttime.dcp.pojo.child;
2
3 import java.math.BigDecimal;
4 import java.sql.Timestamp;
5 import java.util.Date;
6
7 public class Child {
8     protected int id;
9     protected int organisationId;
10    protected String firstName;
11    protected String lastName;
12    protected String screenName;
13    protected String diet;
14    protected BigDecimal multiplier;
15    protected Timestamp createTime;
16    protected Timestamp editTime;
17    protected Boolean disabled;
18    protected int groupId;
19    protected int organisationalUnitId;
20    protected Date dateOfBirth;
21
22    public int getId() {
23        return id;
24    }
25    public void setId(int id) {
26        this.id = id;
27    }
28    public int getOrganisationId() {
29        return organisationId;
30    }
31    public void setOrganisationId(int organisationId) {
32        this.organisationId = organisationId;
33    }
34    public String getFirstName() {
35        return firstName;
36    }
37    public void setFirstName(String firstName) {
38        this.firstName = firstName;
39    }
40    public String getLastName() {
41        return lastName;
42    }
43    public void setLastName(String lastName) {
44        this.lastName = lastName;
45    }

```

KUVA 20. Osa POJO -luokasta (Child.java)

Työvuorosuunnittelusovellukseen käyttöön tehty ESB -toteutus on turhan järeä implementaatio, mutta tarkoitus on myöhemmin integroida työvuorosuunnittelusovelluksen tarvitsemat ominaisuudet toisiin ESB -toteutuksiin, jolloin järjestelmien määrä ESB:ssä kasvaa. Silloin vasta ESB -arkkitehtuurin hyödyt tulevat esiin. REST -palvelimen sijoituspaikka on vielä toistaiseksi epävarma. Se voi sijaita joko Smarttime:n tai Mikkelin kaupungin tiloissa. On myös mahdollista, että palvelin sijaitsee sa-

massa koneessa kuin REST -asiakasohjelmisto. Tämä ei ole kuitenkaan pakollista, sillä REST -palvelimeen implementoidaan myöhemmin tunnistautumisominaisuus, jottei ei-toivottuja pyyntöjä pääsisi läpi.

5 PÄÄTÄNTÖ

Työvuorosuunnittelusovelluksen julkaisu hieman myöhästyi tavoiteaikataulusta osittain johtuen ketterästä kehitystavasta, koska asiakkaan toiveita sovelluksen suhteen on kuunneltu ja asiakkaan haluamia ominaisuuksia on toteutettu melko anteliaasti. Toisaalta ominaisuuksien lennosta muuttaminen on ketterän sovelluskehittämisen parhaita puolia. Silloin saadaan pidettyä asiakas ja toteuttaja tyytyväisenä. Vaikka ensimmäisen vaiheen julkaisu viivästyikin, niin mielestäni projektin on sujunut tähän asti hyvin. Pilottiprojektin luonteeseen kuuluu, että vastoinkäymisiäkin on. Työvuorosuunnittelusovellus on tällä hetkellä käytössä vuoropäiväkoti Vilttihatussa ja kehitys jatkuu vaiheilla kaksi ja kolme. Vilttihatun työvuorosuunnittelu prosessia on siis saatu lyhennettyä, joten projektin tavoite on saavutettu. Siitä huolimatta projekti saatetaan loppuun, sillä sovellukset eivät vielä tee täysin sitä mitä niiden olisi tarkoitus tehdä.

ESB -arkkitehtuuri on auttanut Smart Time Oy:tä kehittämään integraatiopalveluita luotettavammin, nopeammin ja siten myös tehokkaammin. Pienelle yritykselle on erityisen tärkeää, että työvälineet ja ns. pohja toteutukselle ovat kunnossa. Pienen yrityksen resurssitkin ovat pienemmät, joten maksullisten työvälineiden ja työtapojen kokeileminen ei aina ole järkevää. Helposti menetetään enemmän rahaa kuin tienataan.

Ilmeisiä jatkokehityskohteita ovat projektin vaiheet kaksi ja kolme. Niiden onnistuminen, varsinkin vaiheen kolme, riippuu paljon Mikkelin kaupungista ja siitä miten he suostuvat yhteistyöhön projektin osalta. Ongelmakohtia ovat mm. tietokantojen fyysinen ja looginen sijoitus, verkkosivujen integrointi, REST -pyyntöjen salliminen ja palveluiden ylläpito.

LÄHTEET

- Hunter, Lee 2008. The Drupal overview. WWW-dokumentti. <https://drupal.org/getting-started/before/overview>. Päivitetty 28.4.2014. Luettu 29.5.2014.
- Jansen, Geert 2011. Methods. WWW-dokumentti. <http://restful-api-design.readthedocs.org/en/latest/methods.html>. Päivitetty 22.5.2014. Luettu 1.6.2014.
- Kennedy, Giles 2003. System requirements. WWW-dokumentti. <https://drupal.org/requirements>. Päivitetty 30.5.2014. Luettu 31.5.2014.
- Kress, Jürgen, Maier, Berthold, Normann, Hajo, Schmeidel, Danilo, Schmutz, Guido, Trops, Bernd, Utschig-Utschig, Clemens & Winterberg, Torsten 2013. Enterprise Service Bus. WWW-dokumentti. <http://www.oracle.com/technetwork/articles/soa/ind-soa-esb-1967705.html>. Ei päivitystietoa. Luettu 1.6.2014.
- Mason, Ross 2009. To ESB or not to ESB. WWW-dokumentti. <http://blogs.mulesoft.org/to-esb-or-not-to-esb/>. Päivitetty 6.7.2009. Luettu 1.6.2014.
- Mason, Ross 2011a. ESB or not to ESB Revisited – Part 1. WWW-dokumentti. <http://blogs.mulesoft.org/esb-or-not-to-esb-revisited-part-1/>. Päivitetty 8.6.2011. Luettu 1.6.2014.
- Mason, Ross 2011b. ESB or not to ESB Revisited – Part 2. WWW-dokumentti. <http://blogs.mulesoft.org/esb-or-not-to-esb-revisited-%E2%80%93-part/>. Päivitetty 28.6.2011. Luettu 1.6.2014.
- Mielke, Torsten 2013. Tutorial: Managing Apache ServiceMix clusters with Fuse Fabric. WWW-dokumentti. <http://jaxenter.com/tutorial-managing-apache-servicemix-clusters-with-fuse-fabric-46885.html>. Päivitetty 3.5.2013. Luettu 26.5.2014.
- Mikkelin kaupunki 2013. Päiväkoti Vilttihattu automatisoi toimintojaan uudella sovelluksella. WWW-dokumentti. <http://newsread.hosting.fi/1383644761.30261006/>. Päivitetty 5.11.2013. Luettu 29.5.2014.
- MuleSoft 2013. Understanding Enterprise Application Integration - The Benefits of ESB for EAI. WWW-dokumentti. <http://www.mulesoft.com/resources/esb/enterprise-application-integration-eai-and-esb>. Ei päivitystietoa. Luettu 1.6.2014.
- Oracle Corporation 2010. JMS as a MOM standard. WWW-dokumentti. <http://docs.oracle.com/cd/E19340-01/820-6424/aerar/index.html>. Päivitetty 26.1.2011. Luettu 28.5.2014.
- Singh, Tarandeep 2009. REST vs. SOAP – The Right WebService. WWW-dokumentti. <http://geeknizer.com/rest-vs-soap-using-http-choosing-the-right-webservice-protocol/>. Päivitetty 29.5.2014. Luettu 1.6.2014.
- The Apache Software Foundation 2002. What is Maven?. WWW-dokumentti. <http://maven.apache.org/what-is-maven.html>. Päivitetty 22.5.2014. Luettu 1.6.2014.

The Apache Software Foundation 2004a. What is ActiveMQ. WWW-dokumentti. <http://activemq.apache.org/what-is-activemq.html>. Päivitetty 19.2.2014. Luettu 28.5.2014.

The Apache Software Foundation 2004b. What is Camel. WWW-dokumentti. <http://camel.apache.org/what-is-camel.html>. Päivitetty 16.3.2014. Luettu 1.6.2014.

The Apache Software Foundation 2004c. Is Camel an ESB?. WWW-dokumentti. <http://camel.apache.org/is-camel-an-esb.html>. Päivitetty 16.3.2014. Luettu 30.5.2014.

The Apache Software Foundation 2004d. Components Included. WWW-dokumentti. <http://camel.apache.org/components.html>. Päivitetty 22.4.2014. Luettu 1.6.2014.

The Apache Software Foundation 2008a. Apache ServiceMix. WWW-dokumentti. <http://servicemix.apache.org/index.html>. Päivitetty 20.5.2014. Luettu 26.5.2014.

The Apache Software Foundation 2008b. What is ServiceMix 4?. WWW-dokumentti. <http://servicemix.apache.org/docs/5.0.x/user/what-is-smx4.html>. Päivitetty 28.5.2014. Luettu 30.5.2014.

The Apache Software Foundation 2008c. Apache CXF: An Open-Source Services Framework. WWW-dokumentti. <http://cxf.apache.org/>. Päivitetty 20.5.2014. Luettu 1.6.2014.

The Apache Software Foundation 2008d. Apache Karaf. WWW-dokumentti. <http://karaf.apache.org/>. Päivitetty 14.4.2014. Luettu 28.5.2014.

The Apache Software Foundation 2008e. ServiceMix Installation. WWW-dokumentti. <http://servicemix.apache.org/docs/5.0.x/quickstart/installation.html>. Päivitetty 28.5.2014. Luettu 28.5.2014.

The Apache Software Foundation 2008f. Optional features. WWW-dokumentti. <http://servicemix.apache.org/docs/5.0.x/quickstart/features.html>. Päivitetty 28.5.2014. Luettu 1.6.2014.

The Apache Software Foundation 2008g. Using Camel. WWW-dokumentti. <http://servicemix.apache.org/docs/5.0.x/quickstart/camel.html>. Päivitetty 28.5.2014. Luettu 1.6.2014.

Wähner, Kai 2013. Choosing the Right ESB for Your Integration Needs. WWW-dokumentti. <http://www.infoq.com/articles/ESB-Integration>. Päivitetty 2.4.2013. Luettu 25.5.2014.