

Dynaamisen WEB-lomakkeen toteutus Fivaldi WorkFlow - sovelluksessa

Teemu Anttila

Opinnäytetyö

Tietojenkäsittelyn koulutusohjelma

2014



Tekijä tai tekijät Teemu Anttila	Ryhmätunnus tai aloitusvuosi 2008
Raportin nimi Dynaamisen WEB-lomakkeen toteutus Fivaldi WorkFlow – sovelluksessa	Sivu- ja liitesivumäärä 23+1
Opettajat tai ohjaajat Amir Dirin	
<p>Opinnäytetyö tehtiin toimeksiantona Oy Finnvalli Finland Ab:lle ja sen tarkoituksena oli toteuttaa dynaaminen WEB-lomake. Toteutus tuli osaksi suurempaa sovellusta ja sen tarkoituksena oli mahdollistaa erilaisten lomakkeiden tulostus eri käyttötarkoituksiin.</p> <p>Tietoperustana on eri ohjelmointikieliä, joiden avulla olisi mahdollista toteuttaa tietokannan kanssa kommunikoiva WEB-sovellus. Lisäksi on esitetty projektin pohjana olleet tekniset- ja käyttäjävaatimukset.</p> <p>Ratkaisu esitellään selkeästi askel kerrallaan ja selvennetään ohjelman toimintaa. Toteutustapoja on esitelty esimerkki koodein.</p> <p>Viimeisenä esitetään saavutetut tulokset, kohdatut haasteet, pohditaan projektin onnistumista ja jatkokehitys- ja parannusehdotuksia.</p>	
Asiasanat PL/SQL, HTML, dynaaminen, JavaScript, ohjelmistokehitys	

Degree programme

<p>Authors Teemu Anttila</p>	<p>Group or year of entry 2008</p>
<p>The title of thesis Dynamic web form implementation in Fivaldi Workflow application</p>	<p>Number of report pages and attachment pages 23+1</p>
<p>Advisor(s) Amir Dirin</p>	
<p>This thesis was assignment from Oy Finnvalli Finland Ab and its purpose was to develop dynamic web form. Implementation became part of a larger application and its purpose was to enable the printing of different kind of forms.</p> <p>The theoretical part of the thesis consists of different programming languages. which can be used to develop web application that communicates with a database. In addition, technical and user requirements of this project have been represented.</p> <p>The solution is represented distinctly step-by-step and actions of the application are clarified. Example codes are used to represent the implementation.</p> <p>At last the achieved results and the faced challenges are represented. Success of the project and further development and improvement suggestions are discussed in the end.</p>	
<p>Key words PL/SQL, HTML, dynamic, JavaScript, software development</p>	

Sisällys

1	Johdanto	1
1.1	Toimeksiantaja	1
2	Sovellusympäristö ja vaatimukset.....	2
2.1	Fivaldi-sovellus	2
2.2	Fivaldi WorkFlow.....	2
2.3	Tekniset vaatimukset	3
2.4	Käyttjävaatimukset	4
3	Teoria.....	5
3.1	Java	5
3.2	Microsoft .NET Framework.....	5
3.3	PL/SQL.....	6
3.4	Javascript ja jQuery	7
3.5	PHP	7
4	Määrittely ja suunnittelu	9
4.1	Käyttötapaukset.....	9
4.2	Käytettävät menetelmät.....	10
5	Ratkaisu	11
5.1	Tulostettavan lomakkeen tietojen kerääminen	11
5.2	Lomakkeen tulostus	12
5.3	Tapahtuman prosessi.....	15
5.4	Dokumenttien lisäys tehtävälle.....	15
5.5	Lomakkeen tallentaminen	15
5.6	Kenttätarkistukset	16
5.7	Viestien lähetys	17
5.8	Käytettävyys	18
5.9	Tietoturva	18
6	Testaus.....	19
7	Yhteenveto	20
7.1	Haasteet ja pohdinnat	20
7.2	Jatkokehitys	20

7.3 Oman oppimisen arviointi	20
Lähteet	22
Liitteet.....	24

1 Johdanto

Tämän opinnäytetyön aiheena on toteuttaa dynaaminen HTML-pohjainen tehtävänsyöttölomake, joka soveltuu erilaisten lomakkeiden tulostukseen ja tallennukseen. Toimeksiantajana projektissa toimi Oy Finnvalli Finland Ab, myöhemmin Finnvalli, ja lopputuloksena syntynyt ratkaisu tuli osaksi Fivaldi WorkFlow, myöhemmin FWF, – sovellusta. Ensisijaisesti FWF-sovellusta käytetään kiinteistöhallinnan huoltotehtävien ja vikailmoitusten hallintaan.

Opinnäytetyö pitää sisällään lomakkeen tulostuksen suunnittelun ja toteutuksen. Tuloksen testaus on rajattu opinnäytetyön ulkopuolelle, koska testauksesta vastaa erikseen Finnvallin testaustiimi. Kuitenkin toteutuksen yhteydessä suoritettiin luonnollisesti jatkuvaa käyttöttestausta.

1.1 Toimeksiantaja

Opinnäytetyön toimeksiantajana toimi Oy Finnvalli Finland Ab, joka tuottaa pääasiallisesti tilitoimistoille ja kiinteistöhallinnan isännöitsijöille kokonaisratkaisua SaaS-palveluna Fivaldi-sovellusta. Yritys on vahvasti kasvava ja sen liikevaihto oli vuonna 2012 noin 4 miljoonaa euroa. Finnvallissa työskentelee noin 30 henkilöä myynti, sovellusasiantuntija, sovelluskehittäjä ja järjestelmäasiantuntija tehtävissä.

2 Sovellusympäristö ja vaatimukset

2.1 Fivaldi-sovellus

Fivaldin eri sovellukset on toteutettu HTML-sivuina käyttäen avuksi CSS-kieltä ja JavaScriptiä tai Oraclen Formsin avulla. Molemmissa tapauksissa käytetään avuksi Oracle-tietokannassa olevia PL/SQL –paketteja, jotka sisältävät proseduureja ja funktioita. Käyttöliittymät kutsuvat tietokannasta proseduureja ja funktioita, jotka hoitavat käyttöoikeuksien, istuntojen voimassaolon, parametrien tarkistuksen ja virhetilanteiden käsittelyn, sekä kutsuvat edelleen muita proseduureja ja funktioita, jotka suorittavat vaaditut toimenpiteet, kuten tietojen tallennuksen tai poiston tietokannasta. Kutsutut proseduurit ja funktiot voivat palauttaa tietoa toisilleen aina käyttöliittymälle asti. Käyttöliittymälle palautettava tieto voi olla eri muodoissa, kuten HTML, XML tai JSON.

2.2 Fivaldi WorkFlow

Fivaldi WorkFlow on uudiskehityksessä oleva sovellusosa Fivaldi-sovelluksessa. Se tulee korvaamaan entisen Työmääräinkäsittely – sovelluksen, joka on toteutettu Oraclen Forms – tekniikalla. FWF monipuolistaa haluttua käyttötarkoitusta. Sen ominaisuuksia ovat erilaisten tehtävien ja tapahtumien hallinta ja niiden prosessien hallinta. Uusi selainpohjainen HTML-lomake ratkaisu mahdollistaa tehtävä- ja tapahtumakohtaiset lomakkeet erilaisin syöttökentin. Tehtävällä voi olla rajaton määrä kohteita, sekä osapuolia ja tarpeen vaatiessa osapuolia voidaan laskuttaa.

FWF:n perusta on eri tehtävätyypit. Niihin perustuu sovelluksen muu toiminnallisuus. Tehtävätyypistä riippuen, sillä voi olla erilaisia kohteita, kenttiä, tapahtumia, osapuolia ja prosesseja. Nämä kaikki ovat käyttäjän itse määriteltävissä.

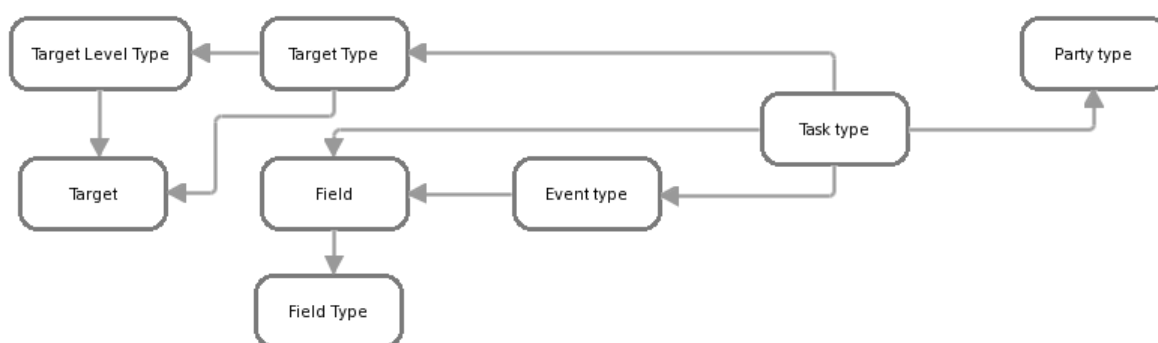
Tehtävä voi liittyä erilaisiin kohteisiin. Kohde voi olla mitä tahansa mihin suoritettava tehtävä liittyy, kuten esimerkiksi asuinhuoneisto. Kohdetyypit ovat usea tasoisia puurakenteita, jotka ovat käyttäjän vapaasti määriteltävissä. Kun tehtävälle valitaan jokin kohde, sen takaa voi tehtävälle tulla osapuoleksi esimerkiksi asuinhuoneiston asukas.

Tehtävään liittyvät osapuolet voivat olla henkilöitä tai yrityksiä, kuten tehtävän suorittava huoltoyhtiö. Osapuolen liittyvyyden tehtävään kertoo sen tyyppi. Osapuolia voidaan laskuttaa tapahtumakohtaisesti tai osapuoli voi olla vastuullisena tapahtumassa.

Tapahtumat ovat tehtävään liittyviä toimenpiteitä ja niiden avulla seurataan tehtävän suorittamisen etenemistä. Tehtävä on valmis, kun sen kaikki tapahtumat ovat merkitty valmiiksi. Tapahtumia voi luoda käsin tai automaattisesti prosessin edetessä.

Tehtävä- ja tapahtumatyypeillä voi olla erilaisia kenttiä. Sovelluksessa on tiettyjä ennalta määriteltyjä kenttiä tehtäville ja tapahtumille, mutta käyttäjä voi myös itse luoda uusia kenttiä. Kentät voivat olla esimerkiksi teksti-, päivämäärä- tai listakenttiä.

Lisäksi sovelluksen perustoiminnallisuuteen kuuluu prosessi, joka voi olla erilainen tehtävä- ja tapahtumatyypeillä. Prosessin avulla tehtävän suoritusta voidaan seurata ja mahdollistetaan viestien lähetys tapahtumista halutuille osapuolille.



Kuva 1. FWF – rakenteiden riippuvuudet

2.3 Tekniset vaatimukset

Lomakkeen tulostus toteutetaan käyttäen HTML- ja CSS-kieltä. Toiminnallisuuksia luodaan JavaScriptillä käyttäen apuna jQuery-kirjastoa. Kullakin kerralla tulostettavan lomakkeen tiedot ja kentät haetaan käyttöliittymälle PL/SQL- ja SQL-ohjelmointikielien avulla. Tietokantana on Oracle ja se pitää sisällään tallennetut lomakemäärittelyt ja kenttien tallennetut arvot tauluissa.

Lomakkeen tulee toimia selainriippumattomasti yleisimmillä selaimilla, esimerkiksi Internet Explorer, Mozilla Firefox, Google Chrome ja Safari. Toteutuksessa noudatetaan Finnvallin toimintamalleja, käytäntöjä ja ohjeita

2.4 Käyttäjävaatimukset

Toteutettavalla ratkaisulla tulee pystyä tulostamaan erilaiset lomakkeet ja kentät. Jokaisen tulostettavan lomakkeen tulee tulostua, kuten se on käyttäjän toimesta määritelty. Lomakkeilla voidaan luoda uusia, sekä muokata vanhoja tehtäviä ja tapahtumia.

Käyttäjälle tulostetaan vain lomakkeita, joihin hänellä on oikeudet. Oletuksena tulostetaan tehtävä- tai tapahtumatyypille määritelty oletuslomake. Jos käyttäjällä oikeudet useampaan lomakkeeseen, luodaan mahdollisuus vaihtaa lomakkeiden välillä.

Kentissä tulee olla perustarkistukset, että niihin ei voida syöttää virheellisiä arvoja. Virheellisiä arvoja voivat olla liian pitkät syötteet tekstikentissä tai virheelliset päivämäärät ja ajat. Lisäksi kenttien pakollisuus tallennusvaiheessa tulee tarkistaa. Virheellisyyksistä pitää ilmoittaa selkeästi käyttäjälle.

Tapahtumat seuraavat niille ennalta määriteltyä prosessia erilaisten toimintojen avulla. Toimintojen avulla prosessi etenee halutulla tavalla ja tehtävät saadaan valmiiksi suunnitellusti.

Käyttäjille halutaan lähettää viestejä, kun tehtäviä ja tapahtumia luodaan, muutetaan tai kun ne tulevat valmiiksi. Viestejä voidaan lähettää tehtävän osapuolille ja kullekin osapuolelle mahdollisesti erilainen viesti.

3 Teoria

Ohjelma olisi mahdollista toteuttaa eri ohjelmointikielillä ja alustoilla. Tässä kappaleessa esitetään näistä yleisimpiä.

3.1 Java

Java on kehitetty C-ohjelmointikielen pohjalta. Javalla on kuitenkin oma rakenne, syntaksisäännöt ja oma ohjelmointi ajatusmalli. Javan ajatusmallin perustana on olio painotteinen ohjelmointi. Tätä tukee myös ohjelmointikielen toiminnot. (Perry. 2010.)

Java ohjelmat käännetään väliaikaisesti Java tavukoodiksi, jonka Java Virtual Machine (JVM) kääntää toimivaksi ohjelmistoksi käytettävällä käyttöjärjestelmällä ja laitteistolla. Käännetty Java ohjelma voidaan suorittaa millä tahansa alustalla, jossa on jokin JVM:n versio. (DeveloperWorks.)

Web-ohjelmointii käytetään Java Enterprise Edition (Java EE) versiota. Kehittäjät käyttävät apunaan myös laajaa valmista Java kirjastoa nimeltään Java Application Programming Interface (Javav API). Se sisältää perus luokkia laajalta alalta tietoturvallisuudesta XML:n käsittelyyn ja moneen muuhun. (DeveloperWorks.)

Javan parhaita puolia on sen skaalautuvuus ja siirrettävyys. Java ohjelmat ovat monitasoisia, joten samaa Java-koodia on helppo käyttää eri alustoille. Java ohjelmat ovat helppo siirtää alustalta toiselle ja eri käyttöjärjestelmään JVM:n takia, koska se ymmärtää aina samalla tavalla Java tavukoodia, johon kaikki Java ohjelmat käännetään. (DeveloperWorks.)

3.2 Microsoft .NET Framework

Microsoft .NET Framework (myöhemmin .NET) on käytön aikainen suoritusympäristö, joka hallitsee ohjelmia, joiden kohde on .NET. .NET koostuu kahdesta pääkomponentista: the common language runtime (CLR), joka on suorittaa käytettävät ohjelmat; ja .NET luokka kirjasto, josta kehittäjät voivat käyttää testattua ja uudelleenkäytettävää koodia. (MSDN 2014a.)

.NET:n yhtenä etuna on, että kehittäjä voi itse valita ohjelmointikielen, jota käyttää. Sen täytyy vain tukea .NET:a. Koska .NET tarjoaa ohjelmointikieli riippumattomuuden ja yhteentoimivuuden, voidaan olla vuorovaikutuksessa muiden .NET ohjelmien kanssa riippumatta siitä mitä ohjelmointikieltä niissä on käytetty. Lisäksi .NET:a on kehitetty yhteensopivuus periaatteella, että aiemmin luodut ohjelmat toimivat myöhemmissäkin .NET-versioissa. (MSDN 2014a.)

CLR hoitaa muistin käsittelyn, säiesuorituksen, koodin suorituksen, turvallisuuden tarkistuksen ja kääntämisen, sekä muut järjestelmä palvelut. Nämä ominaisuudet ovat olennainen osa hallittua koodia, joka suoritetaan CLR:ssa. CLR poistaa monet yleiset ohjelmisto ongelmat. Esimerkiksi se hoitaa viittaukset objekteihin ja vapauttaa ne, kun niitä ei enää käytetä. Tämä automaattinen muistin hallinta selvittää kaksi yleisintä ohjelmavirhetä, muistivuodot ja virheelliset muistiviittaukset. (MSDN 2014b.)

.NET luokkakirjasto on kokoelma uudelleen käytettäviä tyyppejä, jotka integroituvat tiukasti CLR:n kanssa. Luokkakirjasto on olioperustainen ja tarjoaa tyyppejä, joista kehittäjä voi johtaa toimintoja. Tämä nopeuttaa uusien .NET-toimintojen oppimista. Kuten tyyppillistä olioperustaisille luokkakirjastolle, .NET tyyppit mahdollistavat muun muassa merkkijono käsittelyn, tiedon keruun, tietokantayhteyden ja tiedostojen käsittelyn. (MSDN 2014b.)

3.3 PL/SQL

PL/SQL tulee sanoista Procedural Language extensions to SQL. Eli vapaasti suomennettuna se lisää normaaliin SQL-kieleen ohjelmallisia ominaisuuksia. PL/SQL toimii Oracle-tietokannassa, jolloin erillistä tietokantayhteyttä ei tarvitse määrittää ohjelman ja tietokannan välille. (Oracle 2012.)

PL/SQL on lohkorakenteinen ohjelmointikieli. Tarkoittaen, että perus yksiköt (proseduurit ja funktiot), joista PL/SQL ohjelma muodostuu, ovat loogisia lohkoja. Nämä lohkot voivat sisältää rajattoman määrän sisäkkäisiä alalohkoja. Tyypillisesti yksi looginen lohko vastaa yhtä ongelmaa. (Oracle 2002.)

PL/SQL muuttujat voivat olla tietotyybiltään mikä SQL-tietotyyppi tai jokin PL/SQL:n oma tietotyyppi, kuten *BOOLEAN*. Muuttujia voidaan käyttää SQL-lauseissa, esimerkiksi rajausehtoina. Tietokannasta voidaan myös hakea arvo jollekin muuttujalle. (Oracle.2012.)

PL/SQL:n erityisominaisuus on kursorit, joka toimii nimettynä työalueena. Sen sisältö on SQL-lauseen tulosjoukko tietokannasta. Työaluetta voi käydä läpi rivi kerrallaan, esimerkiksi tulostaa tietoa rivi kerrallaan näkyville. (Oracle.2012.)

3.4 Javascript ja jQuery

JavaScript on dynaaminen ohjelmointikieli, jota käytetään miljoonilla verkkosivuilla. Perussyntaksi on samantapainen kuin C++- ja Java-ohjelmointikielissä, jotta se olisi helppo oppia. JavaScript voi toimia proseduuralisesti tai olio suuntautuneena kielenä. Oliot luodaan käytön aikana toisin kuin käännettävissä ohjelmointikielissä. Luotuja olioita voidaan käyttää alustamaan uusia samanlaisia olioita. (Smith 2013.)

JavaScriptin ohessa käytetään apuna yleisesti kirjastoja. Kirjastoissa on valmiita objekteja, joita kehittäjä voi hyödyntää omilla verkkosivuillaan. Noin kolmanneksella verkkosivuista ei ole mitään JavaScript-kirjastoa käytössä. Jäljelle jäävistä verkkosivuista reilulla 90% on käytössä jQuery JavaScript-kirjasto, joka tekee siitä selkeästi suosituimman ja yleisimmän käytössä olevan kirjaston. (W3Techs 2014.)

jQuery on vapaakäyttöinen aina kaupallisissa tuotteissa asti. Se tarjoaa valmiita objekteja verkkosivun manipuloimiseen, tapahtumakäsittelyyn, animaatioihin ja mahdollistaa yksinkertaiset AJAX (Asynchronous JavaScript And XML) -kutsut. Sen avulla verkkosivuista voi myös tehdä useampaa selainta tukevan, kun jokaiselle selaimelle ei tarvitse tehdä itse sen tarvitsemia erikoisvaatimuksia. (jQuery 2014.)

3.5 PHP

PHP on akronyymi sanoille ”PHP Hypertext Preprocessor”. PHP on avoimen lähdekoodin ohjelmointikieli, jota käytetään laajalti WEB-ohjelmoinnissa sulautettuna

HTML-kielen sisälle. Toisin kuin JavaScript, joka suoritetaan käyttäjän selaimessa, PHP suoritetaan serverillä, jossa HTML-sivu luodaan ja lähetetään takaisin käyttäjän selaimelle nähtäväksi. (PHP.NET 2014.)

PHP-kielen avulla voi käyttää eri tietokanta vaihtoehtoja, kuten MySQL, PostgreSQL tai Oracle. Niihin on mahdollista suorittaa erilaisia tietokantalauseita. Myös tiedostojen käsittely, kuten avaaminen, lukeminen ja kirjoittaminen on mahdollista PHP-kielillä. Selaimen keksimuuttujien käyttäminen ja asettaminen on tuettu PHP-kielissä, kuten on käyttäjien pääsyn estäminen joillekin sivuille WEB-sivustossa. (Tutorialspoint 2014.)

PHP-kielen etuna on sen helppokäyttöisyys aloittelijalle, mutta se tarjoaa kuitenkin lukuisia monipuolisempia ominaisuuksia vaativimmille käyttäjille. PHP-kieltä tukee kaikki yleisimmät käyttöjärjestelmät, kuten Linux, Microsoft Windows tai Mac OS X. Useiden tuettujen protokollien myötä PHP-kielen avulla voi kommunikoida eri palveluiden kanssa. Tuettuja protokollia ovat esimerkiksi LDAP, HTTP, ja POP3. (PHP.NET 2014.)

4 Määrittely ja suunnittelu

4.1 Käyttötapaukset

Tehtäviä luo, tarkastelee ja suorittaa eri henkilöt, joten lomakkeilla on useita eri käyttötapauksia. Pääasiallisesti lomakkeilla luodaan uusia tehtäviä ja kirjataan tapahtumat suoritetuiksi. Nämä toiminnot on kuvattu alempana käyttötapauksina

Taulukko 1. Tehtävän luonti.

Käyttäjä	Järjestelmä
1. Saapuu syöttämään uutta tehtävää.	
	1. Käyttäjälle tulostetaan yhtiö- ja tehtävätyyppi valinnat.
2. Valitsee yhtiön ja tehtävätyypin.	
	2. Haetaan tietokannasta tehtävätyypin oletuslomake ja tulostetaan se käyttäjän ruudulle.
3. Täyttää lomakkeen tiedot.	
	3. Tarkistetaan syötetyt tiedot ja annetaan virheilmoitukset
4. Korjataan puuttelliset tiedot.	
	4. Tallentaa annetut tiedot tietokantaan.
	5. Luodaan automaattisesti ensimmäinen tapahtuma tehtävälle.

Taulukko 2. Tapahtuman suorittaminen.

Käyttäjä	Järjestelmä
1. Avaa suoritettavan tapahtuman.	
	1. Haetaan tietokannasta tapahtumaan liittyvä tehtävä, tehtävä- ja tapahtumatyyppin oikeat lomakkeet käyttäjän osapuolen mukaan.
2. Kirjaa tapahtumaan liittyviä tietoja ja tallentaa tehtävän valmiiksi toiminnolla.	
	2. Tarkistetaan tallennettavat tiedot ja tallennetaan ne tietokantaan.
	3. Haetaan suoritettujen toiminnon jatkotoimenpiteet. Vaihdetaan uusi tila tapahtumalle ja luodan uudet tapahtumat tehtävälle.

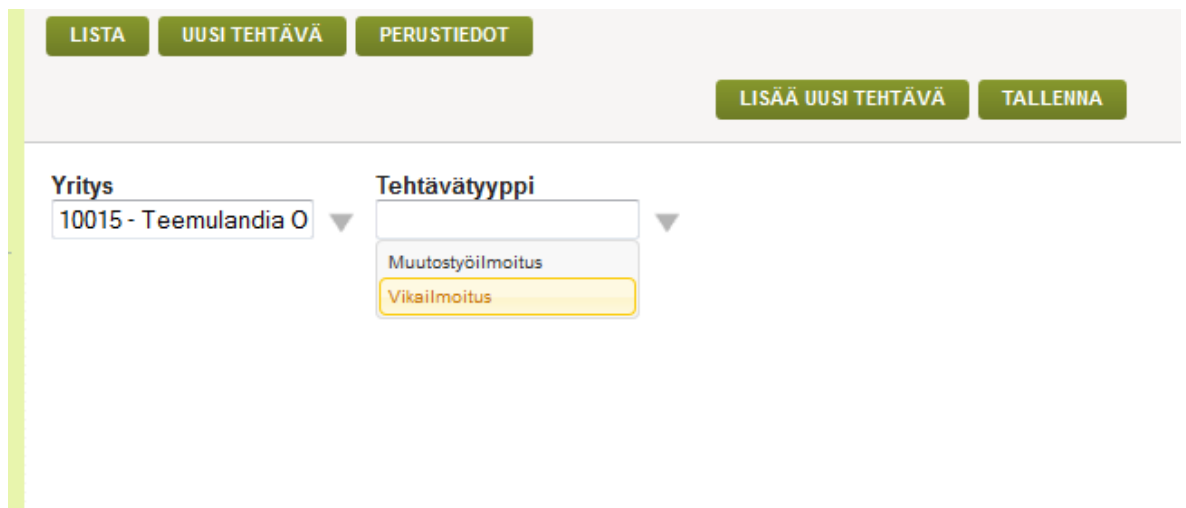
4.2 Käytettävät menetelmät

Fivaldi-sovellus on kehitetty käyttäen alustana Oracle-tietokantaa. Tämän takia luonnollinen valinta on toteuttaa tämä opinnäytetyöprojekti käyttäen ohjelmointikielänä PL/SQL:aa. Lisäksi edellisissä Fivaldi-sovelluksen web-sovellusosissa on käytetty jQuery-kirjastoa, joten sen takia on järkevää käyttää tässäkin projektissa samaa kirjastoa JavaScriptin tukena.

5 Ratkaisu

5.1 Tulostettavan lomakkeen tietojen kerääminen

Lomakkeen tulostavaa proseduuria ”tmar_web.tehtava” kutsutaan kahdella eri tavalla, joko kokonaan ilman parametreja, jolloin kyseessä on uuden tehtävän luonti, tai parametrina annetaan avattavan tapahtuman tunniste. Proseduuritarkastaa käyttöoikeudet ja istunnon voimassaolon. Uuden tehtävän luonti aloitetaan valitsemalla tehtävätyyppi ja yritys, johon tehtävä kohdistuu. Nämä valintalistat ovat toteutettu jQuery UI:n Autocomplete – ominaisuudella (Kuva 2). Kun nämä tiedot ovat valittu, tallennetaan tässä vaiheessa tehtävästä nämä tiedot tietokantaan, mutta merkitään kuitenkin, että tehtävää ei ole vielä varsinaisesti käyttäjä tallentanut. Tämä tehdään siksi, että on mahdollista lisätä dynaamisesti tehtävälle tapahtumia, osapuolia ja liittyviä dokumentteja, ennen varsinaista tallennusta.



Kuva 2. Uuden tehtävän luonti.

Kun proseduuria kutsutaan tapahtuman id parametrina, haetaan sen avulla tiedot siihen liittyvästä tehtävästä. Tällöin on tiedossa tapahtumaan liittyvä tehtävätyyppi, kuten luotaisiin uutta tehtävää. Tehtävätyypin perusteella haetaan tulostettava lomake. Tulostettava lomake riippuu käyttäjän oikeuksista. Mahdollisesti käyttäjällä ei ole oikeuksia mihinkään näkymiin, josta ilmoitetaan punaisella virhetekstillä. Jos käyttäjällä on oikeudet useampaan saman tehtävätyypin näkymään, tulostetaan lisäksi mahdollisuus vaihtaa käytettävää lomaketta.

Jos kyseessä on vanhan tehtävän tulostus, haetaan tehtävän kenttien tallennetut arvot tietokannasta. Tiedot tallennetaan PL/SQL-tauluun (Kuva 3), josta ne voidaan hakea oikeisiin kenttiin tulostusvaiheessa. Samaa PL/SQL-tauluun tallennetaan myöhemmässä vaiheessa kenttien tarkistuksessa ilmenevät virheilmoitukset.

```
20 type r_field is record(value varchar2(2048), error varchar2(256), value_array t_value_array);
21
22 type t_fields is table of r_field
23 index by varchar2(40); --Index is field_id
```

Kuva 3. Kenttien arvojen PL/SQL-taulu.

5.2 Lomakkeen tulostus

Lomakkeen tulostus on erillisessä proseduurissa, jotta sitä voidaan käyttää hyväksi myös tulostettaessa tapahtumien lomakkeita. Tulostus alustetaan muodostamalla PL/SQL-kursori (Kuva 4), johon kerätään tiedot lomakkeen kentistä ja niiden tiedoista (Taulukko 1). Lomakkeen kentillä on tiedot sen sijainnista lomakkeella, rivi- ja sarakenumerot, sekä sen koko tiedot, leveys ja korkeus. Leveys ja korkeus määrittelevät kuinka monta saraketta ja riviä kenttä vie lomakkeella, ei niiden fyysisiä arvoja. Lomakkeella kentän tietona on myös onko kenttä vain näyttökenttä vai voiko sen tietoja muokata lomakkeella.

```
1116 • cursor c_field is
1117 • select a.layout_field_id, a.cols_place,a.rows_place,a.cols_size,a.rows_size, nv(a.display_field,'0') display_field, a.css_id
1118 from tmar_layout_place_t a
1119 where a.yt = v_maint_yt and a.layout_id = v_layout_id
1120 order by a.rows_place, a.cols_place;
1121
```

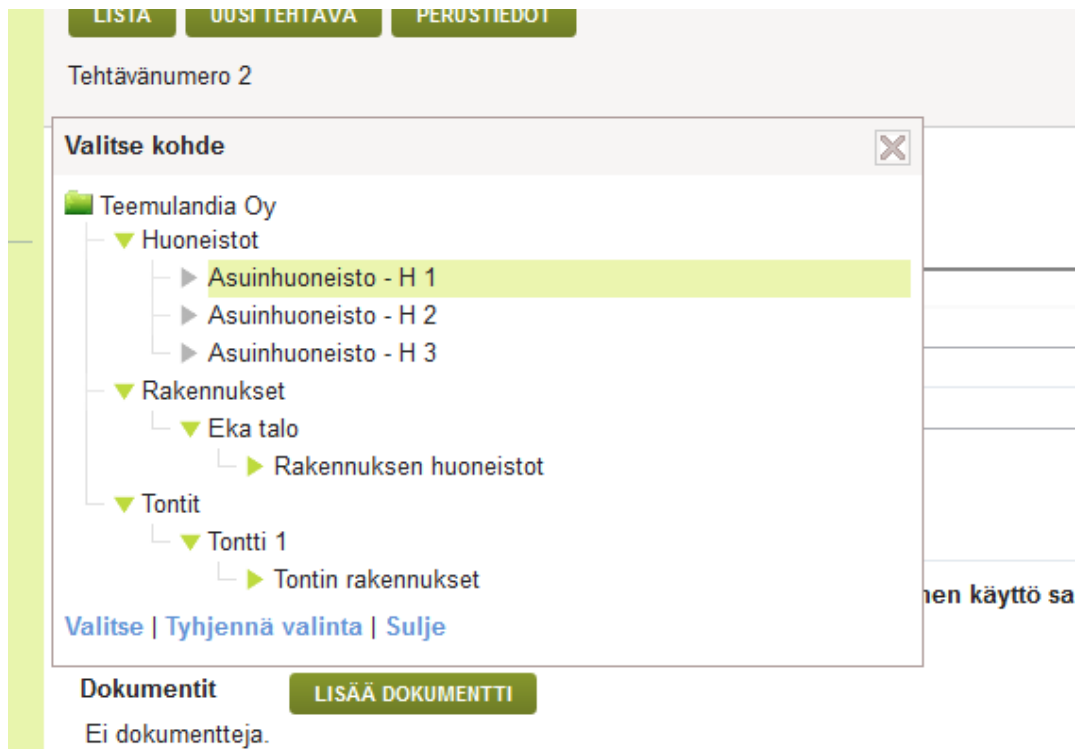
Kuva 4. PL/SQL-kursori lomakkeen kentistä

Taulukko 3. Esimerkki tehtävän lomakkeen kenttien tiedoista. (Vastaa liitettä 1.)

FIELD_ID	ROWS_PLACE	ROWS_SIZE	COLS_PLACE	COLS_SIZE	DISPLAY_FIELD
TARG-KIHA_TREE	1	1	1	2	0
FWF_PARTYS	1	2	3	2	0
TASK_DESC	2	1	1	2	0
3	3	1	1	4	0
1	4	1	1	1	0
2	4	1	2	1	0

Kenttien tulostus päätettiin toteuttaa HTML-kielen taulukko elementtiin, koska se vastaa lähes suoraan, miten kentät on määritelty lomakkeelle. Kentät tulostetaan taulukon soluihin yksi kerrallaan. Solun leveys määritellään prosentteina. Oikea prosenttiluku saadaan lomakkeen tiedoista, koska lomakkeelle on määritelty sen sarakemäärä. Tällä sarakemäärällä voidaan jakaa sadan prosentin maksimileveys, jolloin saadaan yhden sarakkeen prosentuaalinen leveys kullakin lomakkeella. Kentän solulle lasketaan leveys kertomalla yhden sarakkeen leveys kentän määritellyllä leveydellä.

Varsinaisen kentän tulostus on omassa proseduurissaan, jolla tulostetaan solun sisällä kentän tiedot. Kentälle haetaan sen otsikko kielikoodin perusteella, sekä kentän tyyppi. Kentän tyyppin mukaan tulostetaan haluttu kenttä. Erityyppisille kentille asetetaan maksimimerkkimääriä ja tulostetaan eri kenttiä, kuten HTML:n input-, textarea- ja select-elementtejä. Käytössä jQuery UI:n Autocomplete-ominaisuus, jota käytetään listoissa, joista voidaan valita vain yksi arvo. Tehtävillä on mahdollista olla myös kohdekenttiä sekä erillinen osapuolikenttä, joka määrittelee mihin tehtävän osapuolet tulostetaan lomakkeella. Kohteet ovat puurakenteisia valintoja ja valitseminen tulostetaan jQuery:n Ajax kutsulla erilliseen HTML:n div-elementtiin (Kuva 5). Tapahtumilla ainoana erikoiskenttänä voi olla tapahtumalistauskenttä, johon tulostetaan lista tehtävään liittyvistä tapahtumista.



Kuva 5. Kohteen valitseminen

Kenttiä tulostettaessa kentän mahdollinen tallennettu arvo haetaan PL/SQL –taulusta, jonka indeksinä kentän id on. Luonnollisesti uutta tehtävää luodessa tämä PL/SQL –taulu on tyhjä, eikä millekään kentälle löydy tallennettua arvoa. Tämä mahdollistaa saman kentän tulostuksen käytön uutta tehtävää luodessa ja vanhaa näyttäessä.

Käytäessä läpi lomakkeen kenttien kursoria, otetaan jokaisen kentän kohdalla sen rivinumero talteen. Kursorin alussa vertaillaan yksinkertaisella if-lauseella onko vuorossa oleva kenttä tulossa uudelle riville, jolloin on tarpeen luoda uusi rivi taulukkoon.

Kun kaikki kentät tehtävän lomakkeesta on tulostettu, taulukko suljetaan ja seuraavaksi on vuorossa tapahtuman lomakkeen tulostus. Tapahtuman lomake tulostetaan samalla proseduurilla kuin tehtävänkin lomake, kun on haettu tapahtuman tyyppi, tulostettava lomake ja tapahtuman kenttien tallennetut arvot. Tapahtuman erillinen lomake näkyy harmaalla pohjalla liitteessä 1.

5.3 Tapahtuman prosessi

Tapahtumat etenevät ennalta määritettyä prosessia, ja tapahtuman eri vaiheissa tapahtumalla on erilaisia toimintoja, joilla voi esimerkiksi merkitä tapahtuman valmiiksi tai avata uudestaan. Tapahtumalle näkyvät toiminnot riippuvat tapahtuman tilasta, onko se avoin, työn alla vai valmis. Toiminnosta voi myös generoitua uusia tapahtumia tehtävälle, jolloin mahdollisesti siirretään olemassa olevia arvoja uudelle tapahtumalle.

5.4 Dokumenttien lisäys tehtävälle

Dokumentit lisätään tehtäväkohtaisesti. Dokumentit tallennetaan Fivaldin Arkistoverso-ovellukseen, josta ne voidaan hakea ja näyttää listaus tehtävään liittyvistä dokumenteista. Tiedostojen lisäämiseen on valmiiksi olemassa oleva proseduurin ”ark_api_util.sov_dok_lisays”, jota kutsumalla oikeilla parametreilla tallennus onnistuu vaivattomasti. Vastaavasti tehtävään liittyvien dokumenttien haku onnistuu yhdellä funktiolla ”ark_api_util.sov_dok_haku”, joka palauttaa haun id:n, jonka avulla hakutuloksen voi hakea arkiston hakutulostaulusta.

5.5 Lomakkeen tallentaminen

Tehtävän ja tapahtuman lomake tallennetaan aina samaan aikaan. Tehtävää ei voi tallentaa lisäämättä yhtään tapahtumaa, koska lomakkeille tullaan aina tapahtuman id:n perusteella. Tallennuksen voi tehdä tallennuspainikkeella, jolloin tehdään pelkästään kenttien arvojen oikeellisuuden tarkistukset. Lisäksi tallennus tapahtuu tapahtuman toimintonappeja painaessa. Näitä käyttäessä tarkastetaan tehtävän kenttien pakollisuuksia prosessin eri vaiheissa. Tapahtuman kenttien pakollisuus riippuu tapahtuman tilasta.

Tehtävä- ja tapahtumalomakkeen kentät kerätään arvoineen omiin kenttien PL/SQL – tauluihin. Jos tallennetut arvot menevät kenttätarkastuksista läpi tallennetaan tiedot tietokantaan. Muutoin lomakkeet tulostetaan uudestaan, määriteltynä kuitenkin siten, että vanhojen tietojen haku ohitetaan ja tulostettavat arvot tulevat lomakkeilta kerätystä arvoista. Virheellisestä tallennuksesta ilmoitetaan lomakkeen yläpuolella

punaisella virhetekstillä ja virheelliset kentät kehystetään punaisiksi, sekä kentän virheteksti tulostetaan kentän alapuolelle.

5.6 Kenttätarkistukset

Oikeellisuustarkistukset tehdään päivämäärä- ja aikakentille, jolloin tallennetut arvot yritetään yksinkertaisesti asettaa PL/SQL:n date-tyyppiseen muuttujaan halutun päivämäärämuodon avulla (Kuva 6). Muuttujan asettamisen ympärille tehdyn virhekäsittelyn avulla voidaan selvittää, onko päivämäärä annettu oikeassa muodossa. Jos ei ole, tallennetaan virhe kenttien PL/SQL-tauluun virhekenttään, josta se voidaan hakea tulostettavaksi.

```
2308 • elsif v_field_type_id = 'DATE' then
2309
2310 •     v_date_error := false;
2311
2312 •     begin
2313 •         v_date := to_date(v_fields(v_field_id).value,g_date_fm);
2314 •     exception when others then
2315 •         v_fields(v_field_id).error := k('date_error');
2316 •         v_date_error := true;
2317 •         v_virhe := true;
2318 •     end;
2319
2320 •     if not v_date_error then
2321 •         v_fields(v_field_id).value := to_char(v_date,g_date_fm);
2322 •     end if;
```

Kuva 6. Lomakkeen arvon asettaminen date-tyyppiseen muuttujaan.

Tehtävän lomakkeen pakolliset kentät tarkastetaan tapahtuman toiminnon perusteella. Kenttä kerrallaan käydään läpi kenttien PL/SQL –taulu, ja haetaan tieto onko kenttä pakollinen tämän tapahtuman kohdalla. Tämän jälkeen tarkastetaan, että kentälle on asetettu arvo. Jos kentällä ei ole arvoa asetetaan kentälle PL/SQL –tauluun virheilmoitus (Kuva 7).

```

2230 • v_field_id := v_fields.first;
2231
2232 • while v_field_id is not null loop
2233
2234     --Tehtävän pakolliset kentät action_id:n perusteella
2235 • if v_action_id is not null and v_ref_type = 'TASK_ID' then
2236
2237 •     begin
2238 •         select 'xxx' into v_mandatory
2239             from tmar_action_task_field_manda_t a, tmar_task_t b
2240             where a.yt = v_yt and a.action_id = v_action_id
2241                 and field_id = v_field_id
2242                 and a.yt = b.yt and b.task_id = v_ref_id and a.task_type_id = b.task_type_id;
2243 •     exception when no_data_found then
2244 •         v_mandatory := null;
2245 •     end;
2246
2247 •     if v_mandatory is not null and v_fields(v_field_id).value is null then
2248 •         v_fields(v_field_id).error := k('mandatory_field');
2249 •         v_virhe := true;
2250 •     end if;
2251
2252 •     end if;
2253
2254 •     v_field_id := v_fields.next(v_field_id);
2255 • end loop;

```

Kuva 7. Tehtävän pakollisten kenttien läpikäynti.

Tapahtuman lomakkeen pakolliset kentät tarkastetaan tapahtuman tulevan tilan mukaan. Kentät ovat pakollisia joko tulevassa tilassa tai edeltäneissä tiloissa. Pakolliset kentät haetaan samalla tavalla PL/SQL –kursoriin, josta yksi kerrallaan tarkastetaan, että kaikilla kentillä on arvo.

5.7 Viestien lähetys

Tallennusvaiheessa halutaan myös mahdollisesti lähettää viestejä tapahtuman tai tehtävän luonnista, muutoksesta tai valmistumisesta. Tallennuksen yhteydessä kutsutaan viestejä luovaa proseduuria ”tmar_util.create_messages”. Parametreilla proseduurille kerrotaan mitä tehtävälle tai tapahtumalle on tapahtumassa, luodaanko uutta, muutetaanko vanhaa vai tuleeko valmiiksi. Tämä proseduurin hakee tiedot lähetettävistä viesteistä ja osapuolista kenelle ne halutaan lähettää. Viestit tallennetaan tauluun odottamaan niiden lähetystä.

Kun kaikki halutut viestit on luotu, niin ne lähetetään kerralla yhdestä tallennuksesta kutsumalla ”tmar_util.send_messages” proseduuria, joka lähettää kaikki tähän tallennukseen kuuluneet viestit keskitetysti, koska mahdollisista tapahtumista saattaa lähteä samalle osapuolelle sama viesti, eikä yhdestä tallennuksesta haluta lähettää montaa kertaa samaa viestiä. Viestien lähettämisen hoitaa Fivaldin Postitusjono-sovellus. FWF:stä voi lähettää sähköposteja ja tekstiviestejä.

5.8 Käytettävyys

Käytettävyys ei ole suuressa roolissa sovelluksen toteutusvaiheessa. Sovelluksen käytettävyys tulee suoraan sen dynaamisuudesta. Käyttäjä itse suunnittelee lomakkeet ja niiden kenttien paikat. Näin lomakkeiden luoja on vastuussa eri lomakkeiden käytettyydestä. Eri käyttöalustoille on mahdollista luoda omat lomakkeensa ja tämäkin itsessään lisää sovelluksen käytettyyttä.

5.9 Tietoturva

Käyttäjien käyttöoikeudet tarkistetaan ennen jokaista toimintaa selaimen keksin perusteella. Tällöin käyttäjä ei pääse suorittamaan hänelle kuulumattomia toimenpiteitä, eikä näe hänelle kuulumattomia tietoja. Käyttäjillä voi olla erikseen oikeudet nähdä kaikki tehtävät ja oikeudet käyttää kaikkia lomakkeita, mutta peruskäyttäjät näkevät vain omat tehtävänsä, joista ovat vastuussa ja pystyvät käyttämään vain heille määriteltyjä lomakkeita, joiden käyttötarkoitukset voivat olla rajattuja.

6 Testaus

FWF-sovelluksen testauksesta vastasi kehitysvaiheessa Finnvallin erillinen testaustiimi ja lisäksi sovellus on otettu pilottikäyttöön muutamalla yrityksellä, joiden avulla on saatu varmistettua toteuksen toimivuus.

Oleellinen osa testausta on erillaisten lomakkeiden luominen ja niiden testaaminen, että käyttäjät voivat jatkossa luoda vapaasti täysin haluamiaan lomakemalleja ilman minkäänlaisia rajoituksia. Mahdollisuus luoda erilaisia lomakkeita lisää sovelluksen käytettävyyttä, kun esimerkiksi eri alustoille voidaan luoda omat lomakkeensa.

7 Yhteenveto

Projektin tuloksena syntyi ratkaisu, jonka avulla FWF-sovelluksessa pystytään tulostamaan käyttäjälle erilaisia lomakkeita. Toimiva lomakkeen tulostus toimii ratkaisevassa roolissa FWF-sovelluksessa, jolloin käyttäjät voivat luoda monipuolisesti erilaisia lomakkeita eri käyttötarkoituksiin.

7.1 Haasteet ja pohdinnat

Ainoaksi haasteeksi lomakkeiden tulostuksessa muodostui tyhjien kenttien puuttuminen lomakkeiden määrittelyistä, jolloin HTML-taulukko meni sekaisin, kun kaikkia soluja ei ollut määriteltynä HTML-kieleen. Tämä ratkaistiin, niin että lomakkeita määrittelevä työkalu tallentaa tyhjätkin kentät omana tyyppinään, jolloin lomakkeiden tulostuksen ei tarvitse huolehtia näistä. Tämä todettiin helpommaksi ratkaisuksi.

HTML-taulukkoa pidettiin kehitysvaiheessa selkeimpänä ratkaisuna, koska lomakkeiden määrittelyt ovat lähes suoraan, kuten HTML-taulukon rakenne. Muitakin ratkaisuja mietittiin, kuten HTML:n div-elementtien käyttöä, mutta tässä vaiheessa todettiin, että taulukkorakenteella saadaan yksinkertaisemmin aikaiseksi haluttu ratkaisu.

7.2 Jatkokehitys

Jatkokehittävää FWF-sovellukseen jäi, koska osa ominaisuuksista oli selkeästi rajattu tämän projektin ulkopuolelle. Lisäksi varsinaista testaustakaan ei projektin aikana toteutettu tarkemmin.

7.3 Oman oppimisen arviointi

Projektin aikana kehittyi entisestään web-ohjelmointi osaaminen, jota oli jo entuudestaan muista projekteista. Ensimmäistä kertaa luotiin kuitenkin varsinaisesti kokonaan uutta sovellusosaa.

Projektinhallinnallisesti toteutus sujui aikataulun mukaisesti, mutta kirjallinen osuus samassa ajassa tuotti haasteita. Mahdollisesti tulevissa projekteissa siihen panostamiseen ja ajanhallintaan pitää ottaa tarkempi ote.

Lähteet

DeveloperWorks. New to Java programming. Luettavissa:

<http://www.ibm.com/developerworks/java/newto/>. Luettu 1.5.2014.

jQuery. 2014. jQuery website. Luettavissa: <http://jquery.com>. Luettu: 2.5.2014.

Microsoft Developer Network (MSDN). 2014a. Getting Started with the .NET Framework. Luettavissa: <http://msdn.microsoft.com/en-us/library/hh425099.aspx>.

Luettu: 16.4.2014.

Microsoft Developer Network (MSDN). 2014b. Overview of the .NET Framework.

Luettavissa: <http://msdn.microsoft.com/en-us/library/zw4w595w.aspx>. Luettu:

16.4.2014.

Oracle. 2002. Overview of PL/SQL. Luettavissa:

http://docs.oracle.com/cd/B10501_01/appdev.920/a96624/01_oview.htm. Luettu.

16.4.2014.

Perry, Steven. 2010. Introduction to Java programming. Luettavissa:

<https://www.ibm.com/developerworks/java/tutorials/j-introtojava1/>. Luettu:

2.5.2014.

PHP.NET. 2014. What is PHP?. Luettavissa: [http://php.net/manual/en/intro-](http://php.net/manual/en/intro-what-is.php)

[what-is.php](http://php.net/manual/en/intro-what-is.php). Luettu: 7.11.2014.

Smith, Dan ja muut. 2013. About JavaScript. Luettavissa:

https://developer.mozilla.org/en-US/docs/Web/JavaScript/About_JavaScript.

Luettu: 2.5.2014.

Tutorialspoint. 2014. PHP Introduction. Luettavissa:

http://www.tutorialspoint.com/php/php_introduction.htm. Luettu: 7.11.2014.

W3Techs. 2014. Usage of JavaScript libraries for websites. Luettavissa:
http://w3techs.com/technologies/overview/javascript_library/all. Luettu: 2.5.2014.

Liitteet

Liite 1. Kuva valmiista FWF-lomakkeesta.

LISTA **UUSI TEHTÄVÄ** **PERUSTIEDOT**

Tehtävännumero 2 **LISÄÄ UUSI TEHTÄVÄ** **POISTA TÄMÄ TEHTÄVÄ** **TALLENNA**

Yritys
10015 - Teemulandia Oy

Tehtävätyyppi
Vikailmoitus

Lomake
Vikailmoitus

KH rekisteri
Asuinhuoneisto - H 1

Osapuolet **LISÄÄ OSAPUOLI**

Asukas
Juhani Sappula **AVAA** **POISTA**

Huoltoyhtiö
Kurvin huoltoliike Oy **AVAA** **POISTA**

Isännöitsijä
Hippolan Isännöitsijät Oy **AVAA** **POISTA**

Otsikko
Kylpyhuoneen hana vuotaa

Lisätietoa
Tiputtelee vettä lattialle

Huoneistossa kotieläimiä Yleisavaimen käyttö sallittu

Dokumentit **LISÄÄ DOKUMENTTI**
Ei dokumentteja.

Tapahtumat **UUSI TAPAHTUMA**

Tyyppi	Aika	Valmis
Puhelu	17:25 07.05.2014	Valmis
Huoltokäynti	08.05.2014	Avoin

Arvioitu suoritusajankohta
08.05.2014

Vastuullinen
Kurvin huoltoliike Oy - Huoltoyhtiö

Valmistumisaika

Laskutus
Hippolan Isännöitsijät Oy - Isännöi

Lisätietoa
Korjataan vuotava hana kylpyhuoneesta.

Huoltokäynti **POISTA TAPAHTUMA** **VALMIS** **TALLENNA**