



SPOOLER-PROJEKTI

Opinnäytetyö

Jani Heikkinen

**Tietotekniikan koulutusohjelma
Ohjelmistotekniikka - Multimedia**

Hyväksytty __ . __ . _____

SAVONIA-AMMATTIKORKEAKOULU TEKNIikka

Koulutusohjelma

Tietotekniikan koulutusohjelma

Tekijä

Jani Heikkinen

Työn nimi

Spooler -projekti

Työn laji

Päiväys

Sivut

Opinnäytetyö

19.6.2010

34+3

Työn valvoja

Yrityksen yhdyshenkilö

Lehtori

Suunnittelupäällikkö

Jussi Koistinen

Jani Karhunen

Yritys

Enfo Oyj

Tiivistelmä

Tämän insinööri työn aiheena oli luoda web-käyttöliittymä, jolla käyttäjä pystyi hallinnoimaan tulostustöitä ja niiden asetuksia. Työssä jouduttiin tekemään kaksi erillistä käyttöliittymää, koska käyttäjäkuntia oli tuotanto ja palveluneuvonta, joiden tarpeet olivat erilaiset.

Työn kannalta tärkein vaatimus oli luotettavuus. Koska kyseinen web-käyttöliittymä tuli tuotannolliseen käyttöön, olivat vaatimukset korkeat. Käyttöliittymän täytyi pystyä selviytymään eri virhetilanteista itsenäisesti ja myös siten, että käyttäjä sai niistä riittävästi tietoa. 2009 kesästä lähtien web-käyttöliittymä on ollut tuotannollisessa käytössä ja versio 2 on suunnitteilla.

Web-käyttöliittymän ulkoasu toteutettiin Adobe Flex -työkalulla, ja palvelinohjelmakoodi Perl-kielellä. Kaikki tieto tallennetaan ja luetaan MySQL-tietokannasta.

Avainsanat

www, web-käyttöliittymä, Perl, Adobe Flex

Luottamuksellisuus

julkinen

| | | |
|---|--|---------------|
| SAVONIA UNIVERSITY OF APPLIED SCIENCES | | |
| Degree Programme Information Technology | | |
| Author Jani Heikkinen | | |
| Title of Project Spooler-Project | | |
| Type of Project Final Year Project | Date 19 June, 2010 | Pages 34+3 |
| Academic Supervisor Mr Jussi Koistinen, Lecturer | Company Supervisor Mr Jani Karhunen, Design Manager | |
| Company Enfo Oyj | | |
| <p>Abstract</p> <p>The aim of this thesis was to create a web-based user interface, which can be used to handle the printing works and its settings. In this thesis, two different user interfaces were studied because there were also two user groups, whose needs differed very much from each other.</p> <p>Reliability was important for this project. Since this web user interface was created for productive use the requirements were high. The user interface had to solve different problematic situations by itself and the user has to gain enough information. Since summer 2009 the web user interface has been in productive use and version 2 is being designed.</p> <p>The outfit of the web user interface was created with Adobe Flex – development tool, and a server side code was created with the program Perl. All information was saved and read from MySQL database.</p> | | |
| Keywords Adobe Flex, MySQL, Perl, user interface | | |
| Confidentiality public | | |

ALKUSANAT

Haluan kiittää suunnittelupäällikkö Jani Karhusta insinööriyömahdollisuudesta ja työn ohjauksesta. Haluan myös kiittää kehityspäällikkö Samuli Räikköstä työn ohjaamisesta teknisissä asioissa ja ohjelmointityön aikana annetuista neuvoista ja lehtori Jussi Koistista työn kirjoitusvaiheen ohjaamisesta.

Kuopiossa 19.6.2010

Jani Heikkinen

SISÄLLYSLUETTELO

| | |
|---|----|
| LYHENTEET JA SELITTEET..... | 6 |
| 1 JOHDANTO..... | 7 |
| 2 SOVELLUS..... | 8 |
| 2.1 Tarkoitus | 8 |
| 2.2 Suunnittelu | 8 |
| 2.2.1 Tuotanto | 8 |
| 2.2.2 Asiakaspalvelu | 9 |
| 2.2.3 Työkalut | 10 |
| 2.3 Sovelluskehittimet..... | 10 |
| 2.3.1 Adobe Flex Builder | 10 |
| 2.3.2 Adobe Flex SDK..... | 11 |
| 2.3.3 ActionScrip3 | 11 |
| 2.3.4 Perl | 11 |
| 2.3.5 MySQL..... | 11 |
| 2.4 Rakenne..... | 13 |
| 2.4.1 Palvelinpuoli | 14 |
| 2.4.2 Selainpuoli..... | 16 |
| 2.5 Tietokanta..... | 19 |
| 2.5.1 Taulu Spooler_ajot..... | 19 |
| 2.5.2 Taulu Ajot | 19 |
| 2.5.3 Taulu Palveluneuvojat..... | 19 |
| 2.6 Käyttötapaukset..... | 21 |
| 2.6.1 Käyttötapaus # 1 – Tuotannon Spooler – Tulostaminen..... | 21 |
| 2.6.2 Käyttötapaus # 2 – Tuotannon Spooler – Testitulostus | 23 |
| 2.6.3 Käyttötapaus # 3 – Spooler – Huomioitava –kentän muokkaus | 25 |
| 2.6.4 Käyttötapaus #4 – Spooler – Ohje –sivun lukeminen..... | 27 |
| 2.6.5 Käyttötapaus # 5 – Asiakaspalvelun Spooler – Ajon vapautus..... | 28 |
| 2.6.6 Käyttötapaus # 6 – Asiakaspalvelun Spooler – Liitteiden ilmoittaminen.. | 30 |
| 2.7 Testaus..... | 32 |
| 3 YHTEENVETO | 33 |
| LÄHTEET | 34 |
| LIITE 1 - Tietokanta taulu: Spooler_ajot | A1 |
| LIITE 2 - Tietokanta taulu: ajot..... | A3 |

LYHENTEET JA SELITYKSET

| | |
|----------------------|--|
| MXML | <i>Magic eXtensible Markup Language, XML-pohjainen käyttöliittymä merkintäkieli</i> |
| RIA | <i>Rich Internet Applications</i> |
| Ajo | <i>Ajolla tarkoitetaan tulostettavaa työtä. Ajolla on tulostamiseen ja jälkikäsittelyyn liittyviä ohjeita.</i> |
| Vapaat ajot | <i>Tarkoitetaan ajoa, joka on vapaasti tulostettavissa. Ei tarvitse erillistä lupaa tulostamiseen enää.</i> |
| Lupaa odottavat ajot | <i>Tarkoittaa ajoa, joka odottaa palveluneuvojan hyväksyntää. Samainen ajo myös odottaa palveluneuvojan tekemää vapautusta, ennen kuin se siirtyy vapaisiin ajoihin.</i> |
| Ajetut ajot | <i>Ajo, joka on tulostettu. Kuitenkin ajo pidetään tallessa muutaman päivän verran, jotta voidaan tarpeen tullessa uudelleen tulostaa.</i> |
| Ohje-sivu | <i>Sivu, joka sisältää ajoa koskevat käsittelyohjeet.</i> |

1 JOHDANTO

Tämän työn tavoitteena on luoda käyttöliittymä, jonka tarkoitus on olla eräänlaisena putkena, johon tulee ajoja. Ajoilla tarkoitetaan tässä tapauksessa tulostettavia töitä, jotka tulevat digitaalisessa muodossa asiakkaalta ja jotka tulostetaan paperille. Tämä tarkoittaa, että raakadata yhdistellään suunnittelijan luomaan ulkoasuun. Käyttöliittymän avulla käynnistetään tulostusprosessi yksittäisillä ajoilla tai ajoryhmillä.

Käyttöliittymän tavoitteena on helpottaa operaattorin ja palveluneuvojien päivittäistä työtä. Tämä ohjelma tulee laajentamaan käyttömahdollisuuksia ja se tulee vähentämään puhelinviestintää näiden käyttäjäryhmien välillä, ja minkä myötä työtehokkuus nousisi.

Ajojen tiedot tallennetaan MySQL-tietokantaan ja käyttöliittymän selainpuolen toiminnallisuus toteutetaan Adobe Flex 3 -kehitystyökalulla. Adobe Flex 3 -kehitystyökalu käyttää MXML-merkintäkieltä, jolla luodaan käyttöliittymän rakenne. Selainpuolen toiminnallisuus tullaan ohjelmoimaan ActionScript3-kielellä. Kaikki tietokanta- ja palvelintoiminnallisuus toteutettiin Perl-ohjelmointikielellä ja palvelimella on Apache HTTP-palvelinsovellus, jonka päälle rakennetaan palvelintoiminta.

Koska kyseessä on tuotannollinen työkalu, on varmuus ja luotettavuus tärkeitä. Pienikin virhe jossain kohdin prosessia, voi aiheuttaa ison ongelman, ja minkä vuoksi testaaminen on erityisen tärkeää.

Ohjelma tehdään käyttäjäryhmien ajatusten ja toiveiden mukaan, jotta saadaan juuri sellainen ohjelma, joka on hyödyllinen ja myös helppo käyttää.

2 SOVELLUS

2.1 Tarkoitus

Sovelluksen tarkoitus on luoda työkalu tuotannon ja asiakaspalvelun käyttöön. Työkalun avulla pyritään parantamaan tiedonvälitystä ja vähentämään puhelimitse tapahtuvaa kanssakäymistä. Tarkoituksena on myös vähentää työvaiheita ja helpottaa tuotannon työtaakkaa. Ohjelmaan sisällytetään myös erilaisia toimintoja, joiden tarkoitus on olla informatiivisia ja työtä tukevia.

2.2 Suunnittelu

Aluksi haastateltiin tulevia käyttäjiä eli tuotantoa ja asiakaspalvelua. Haastattelulla selvitettiin tarpeet ja vaatimukset käyttöliittymille. Käyttäjärühmien tarpeet olivat erilaiset, joten kummallekin päädyttiin tekemään omat käyttöliittymänsä.

Kahden käyttöliittymän ajatusta pidettiin hyvänä, koska silloin käyttöliittymien kehitystyössä pystyttiin luomaan juuri oikeanlainen käyttäjärühmien tarpeisiin. Perusajatuksena on, että kumpikin käyttöliittymä käyttää samoja palvelinohjelmakomponentteja (Perl-koodia) ja samaa MySQL-tietokantaa.

2.2.1 Tuotanto

Haastatteluissa ilmeni, että toiveita on paljon, mutta kaikkia ei voida sisällyttää ensimmäiseen vaiheeseen. Rajaus oli näin ollen tarpeen, jotta näin saadaan ensimmäinen käyttöliittymä tuotannolliseen käyttöön mahdollisimman pian. Tärkeintä oli saada valmiiksi perustoiminnallisuus, jota jatkokehitetään. Esiin tulleita ideoita myös toteutetaan mahdollisuuksien mukaan.

Ensimmäiseen vaiheeseen otettiin seuraavat toiminnot:

- yksittäisten töiden käynnistäminen
- yksittäisten töiden uusinta ja testaaminen

- ohje-sivun katselu
- huomioita-tekstin lisääminen ja muokkaaminen.

Myöhemmin toteutettavat toiminnot:

- töiden niputus
- työn jakaminen osiin.

2.2.2 Asiakaspalvelu

Myös asiakaspalvelulla oli paljon toiveita. Suurin osa toiveista kuitenkin oli helposti lisättävissä ensimmäiseen vaiheeseen.

Ensimmäiseen vaiheeseen otettiin seuraavat toiminnot:

- luvan antaminen ajolle
- huomioitava-tekstin lisääminen ja muokkaaminen
- liitteiden ilmoitus
- ohje-sivun katselu
- PDF-malli.

Myöhemmin toteutettavat toiminnot:

- pois poimittavat sivut
- liitteen kuva.

PDF-mallin luonti ominaisuus oli alun perin tarkoitus tuoda vasta myöhempään vaiheeseen, mutta sen läheinen toiminnallisuus tulostuksen kanssa auttoi sen toteuttamista jo ensimmäiseen vaiheeseen.

2.2.3 Työkalut

Työkalut oli jo valmiiksi määritelty, joten seuraavassa luvussa esitellään ne lyhyesti. Alun perin oli tarkoitus tehdä tietokanta Microsoftin SQL Server -tuotteen avulla, mutta päädyin kuitenkin avoimen lähdekoodin tuotteeseen eli MySQL:ään, joka on ilmainen ja erittäin suosittu juuri web-sovelluksien tietokantana. MySQL sisältää myös mm. mahdollisuuden varmuuskopiointiin ja lokituksen.

2.3 Sovelluskehittimet

Käyttöliittymä toteutettiin seuraavilla kehittimillä:

- Adobe Flex Builder –ohjelmisto
 - o Versio 3
- Adobe Flex SDK
 - o Versio 3.1
- ActionScript –kieli
 - o Versio 3.0
- Perl –kieli
 - o CGI - Simple Common Gateway Interface Class
 - o DBI - Database independent interface for Perl
 - o Mod_perl
- MySQL –tietokanta.

Seuraavaksi kerrotaan lyhyesti näistä kehitystyökaluista.

2.3.1 Adobe Flex Builder

Adobe Flex Builder -ohjelmisto on Eclipsen pohjautuva RIA-sovellusten kehitystyökalu. Builder mahdollistaa älykkään koodin, vuorovaikutteisen askel-askel- läpimenevän ohjelman virheiden etsimisen ja visuaalisen käyttöliittymien ulkoasujen suunnittelun, toteuttamisen ja toiminnallisuuden toteuttamisen.

2.3.2 Adobe Flex

Adobe Flex on korkeasti tuottava, ilmainen avoimen koodin runko rakentaessa ja ylläpitäessä verkkosovelluksia, jotka käyttäytyy yhdenmukaisesti kaikilla suurimmilla selaimilla, työpöydillä ja käyttöjärjestelmillä. Seuraava lause kuvastaa hyvin Flexiä: Flash on aina ollut hieno työkalu lisätessä interaktiivisuutta nettisivuille. Mutta Flexin tuleminen toi insinööreille, suunnittelijoille ja harrastelijoille mahdollisuuden rakentaa suuren määrän erinlaisia Flash sovelluksia.[1]

2.3.3 ActionScript

ActionScript on ohjelmointikieli, joka pohjautuu ECMAScript:iin. Pääasiallisesti ActionScriptiä käytetään verkkosivujen kehittämiseen ja sellaisten sovellusten kehittämiseen, jotka käyttävät Adoben Flash Player -alustaa. Alun perin Macromedian kehittäminen kieli on Adoben omaisuutta. ActionScript oli alussa suunniteltu hallinnoimaan Adoben Flashilla tehtyjä yksinkertaista 2D-vektorianimaatioita. Myöhemmissä versioissa on lisätty toiminnallisuutta erityisesti Web-pohjaisiin ratkaisuihin. Tästä hyvänä esimerkkinä web-pohjaiset pelit ja mahdollisuus toistaa videota ja audiota rikkailla Internet-sovelluksilla. [2]

2.3.4 Perl

Perl on monimuotoinen ja helposti opittava skriptimäinen ohjelmointikieli. Se on ottanut vaikutteita syntaksiinsa mm. awkista (joka on tekstinkäsittelyyn kehitetty ohjelmointikieli), sedistä (tehokas ohjelmoitava vuonsuodatin tekstivirran käsittelyyn), shell-skripteistä, C-kielestä ja Lispistä. Perlän vahvuus on säännöllisten lausekkeiden helppous, mikä tekee siitä myös suosituksen kielen. ”Perlistä onkin sanottu, että se on ohjelmointikieli jolla helpot asiat ovat helppoja ja vaikeat asiat mahdollisia. Kielessä on pyritty suureen ilmaisuvapauteen, ja ongelman voi ratkaista monella tavalla.”[3]

2.3.5 MySQL

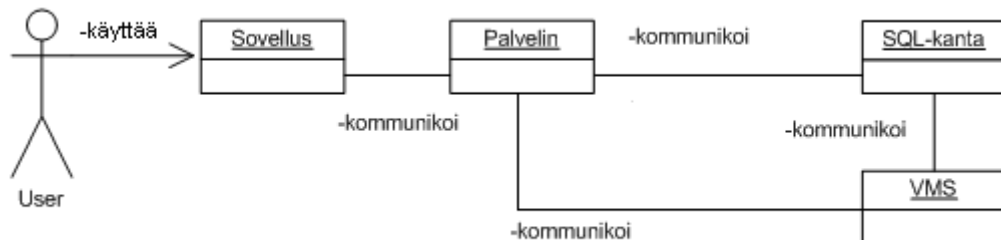
MySQL on maailman suosituin Open Source -tietokanta, joka sopii hyvin tämän projektin tarpeiden kaltaiseen käyttöön. Alun perin ajatuksena oli käyttää Microsoftin SQL Server -tuotetta, mutta päädyttiin MySQL:ään, koska se on ilmainen,

helppokäyttöinen eikä niin raskas kuin SQL Server. Vaikkakin MySQL ei ole aivan niin järeä työkalu kuin SQL, se pystyy hallitsemaan suuria määriä dataa ja yksinkertaiset kyselyt onnistuvat helposti, joten MySQL oli oiva valinta tietokannaksi. [4]

2.4 Rakenne

Kun kaikki osapuolet olivat hyväksyneet toiminnallisen määrittelyn, siirryttiin toteuttamaan määrittelyä. Alkuvaiheessa luotiin perusrakenne, jota ryhdyttiin laajentamaan. Näin varmistuttiin siitä, että kaikki toimii suunnitellusti ja yhteydet toimivat moitteettomasti. Matkan varrella ilmeni uusia ideoita, joita ei suunnitteluvaiheessa oltu otettu huomioon. Näillä ideoilla saatiin käyttöliittymästä enemmän käyttäjäystävällisempi ja käytettävyydeltään parempi. Käyttäjiltä pyydettiin palautetta eri vaiheissa, jolloin varmistettiin että ohjelmasta tuli sellainen kuin haluttiin.

Kuvassa 1 esitellään pelkistetty malli, kuinka komponentit ovat toisiinsa sidoksissa. Yksinkertaisesti systeemin periaate on seuraava: käyttäjä käynnistää sovelluksen, jolloin sovellus käynnistää perl –palvelinohjelman palvelimella, joka hakee MySQL-kannasta datan. Tämä data tulee käyttäjälle näkyviin. MySQL-kannan datan on hakenut palvelimella ollut toinen perl –palvelinohjelman sovellus, joka on käynyt tutkimassa VMS-palvelinta ja tallentanut tiedot MySQL-kantaan. Kun käyttäjä suorittaa toiminnon sovelluksessa, ottaa palvelin yhteyttä joko MySQL-kantaan tai/ja VMS-palvelimeen. Nämä tapahtumat tapahtuvat palvelimella sijaitsevien perl-palvelinohjelmien avulla.



Kuva 1. Pelkistetty datamalli systeemistä

Taulukossa 1 on selitetty datamallissa käytetyt termit.

Taulukko 1. Datamallin selitykset termeille.

| Nimi | Selitys |
|------------------|---|
| <i>Käyttäjä</i> | <i>Sovelluksen käyttäjä</i> |
| <i>Sovellus</i> | <i>Ohjelma, joka toimii käyttäjän työvälineenä toimintoja suorittaessa.</i> |
| <i>Palvelin</i> | <i>Paikka, jossa sijaitsee perl-palvelinohjelmakomponentit. Mahdollisesti myös MySQL-kanta sijaitsee tällä samaisella palvelimella.</i> |
| <i>SQL-kanta</i> | <i>Tietopankki, jossa on kaikkien ajojen tiedot. Tiedot haettu VMS-palvelimelta. SQL-kantana on MySQL-tietokanta.</i> |
| <i>VMS</i> | <i>Palvelin, jolla ajojen tiedostot sijaitsevat. Erinäiset prosessit siirtävät tiedostot tänne sen jälkeen, kun ne on esikäsitelty.</i> |

2.4.1 Palvelinpuoli

Palvelinohjelmakomponentit toteutettiin Perlillä, jolla käsiteltiin tieto ja tietokantakyselyt ja myös erilaiset tiedonhaut eri järjestelmistä.

Palvelinpuolen koodi koostui kolmesta tiedostosta, jotka olivat seuraavat:

- TuotannonSpooler.pl
- Request.pm
- Tulostus.pm.

Tuotannospooler.pl toimi ns. välittäjänä. Tälle ohjelmalle Spooler-ohjelma lähettää käyttöliittymästä tulleet tiedot, ja Tuotannospooler.pl toimittaa tiedon eteenpäin annettujen parametrien perusteella. Eli se ei sisällä toiminnallisuutta, vaan ainoastaan välittää dataa tai tulostaa XML-muotoista dataa, jota sovellus käyttää. Tällä ns. välittäjällä haluttiin eriyttää tiedon vastaanotto ja lähetys pois datan käsittelyn seasta. Tällä tavalla haluttiin selkeyttää palvelinkomponenttien rakennetta.

Request.pm sisältää tietokantakyselyt ja esitettävän datan käsittelyprosessit. Jokaisella selaimen toiminnolla on oma aliohjelmansa, jota kutsutaan tarvittaessa tai jos jokin toinen aliohjelma tarvitsee aliohjelman apua. Aliohjelma palauttaa dataa tai nollan sen mukaan, onko suoritus onnistunut tai kaatunut johonkin virheeseen. Alla on esimerkki

yksinkertaisesta tietokantakyselystä, jossa haetaan tietokannasta palveluneuvojien nimet.

```
## Haetaan palveluneuvojat
sub haePalveluneuvojat
{
    ## esitellään
    my @palveluneuvojat;

    ## Luodaan tietokantakysely
    my $query = qq{ SELECT pname FROM $$SQL_TABLE_PALVELUNEUVOJAT};
    my $sth = $DBH->prepare( $query );
    $sth->execute();

    ## Käydään läpi tietokantakyselyn tulos
    while (my $rivi = $sth->fetchrow_hashref())
    {
        push @palveluneuvojat, $rivi->{name};
    }
    $sth->finish();

    ## Palautetaan data
    return @palveluneuvojat if(wantarray);
    return \@palveluneuvojat;
}
```

Tulostus.pm sisältää kaiken tulostusprosesseja koskevan toiminnallisuuden. Alla on ote Tulostus.pm:stä. Toiminnallisuus on yrityksen salaista tietoa, mutta perusrakenne on alla näkyvä. Jos tulostusprosessin jossain vaiheessa tapahtuu virhe, se palautetaan käyttöliittymälle ja kyseinen ajo merkitään virheelliseksi. Näin ollen tulostusoperaattori näkee virheen ja pystyy ottamaan suunnittelijaan yhteyttä selvittääkseen virheen. Muissa tapauksissa palautetaan nolla tulostuksen onnistumisen merkiksi ja merkitään ajon tila ajetuksi.

```
## Tuotantotulostus
sub tuotanto
{
    ## Parametrien talteen otto
    my ($ajo_nimi, $ajo_aikaleima, $tulostin_nimi) = @_;

    ## toiminnallisuus
    #
    # SALAINEN
    #
    # Jos suorituksessa tapahtui virhe niin palautetaan se.
    If($virhe)
    {
        return $virhe;
    }
    ## Palautetaan 0 onnistuneesta suorittamista
    return 0;
}
```

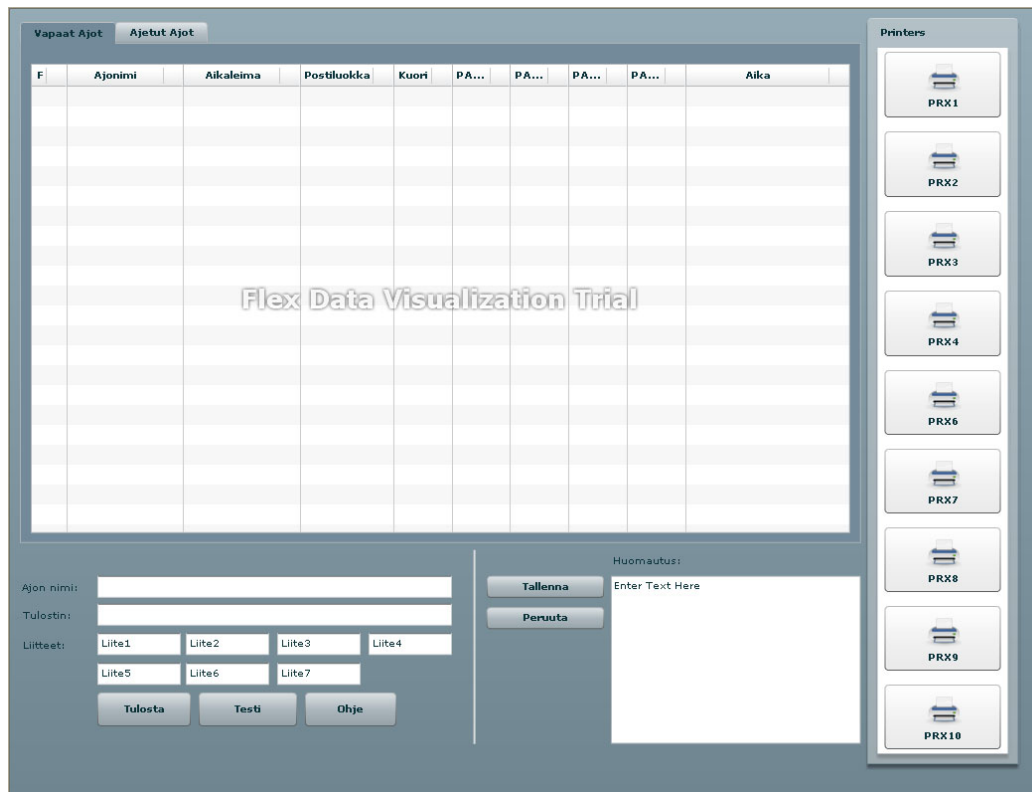
Tulostus.pm käyttää Request.pm:n aliohjelmia, koska tulostusprosessissa tarvitaan mm. tietopäivitystä tietokantaan ja tiedon hakua tietokannasta. Juuri virhetilanteiden

tapahtuessa käytetään Request.pm:ssä olevaa aliohjelmia hyödyksi, kun halutaan tallentaa virhetieto kyseisen ajon tietoihin.

2.4.2 Selainpuoli

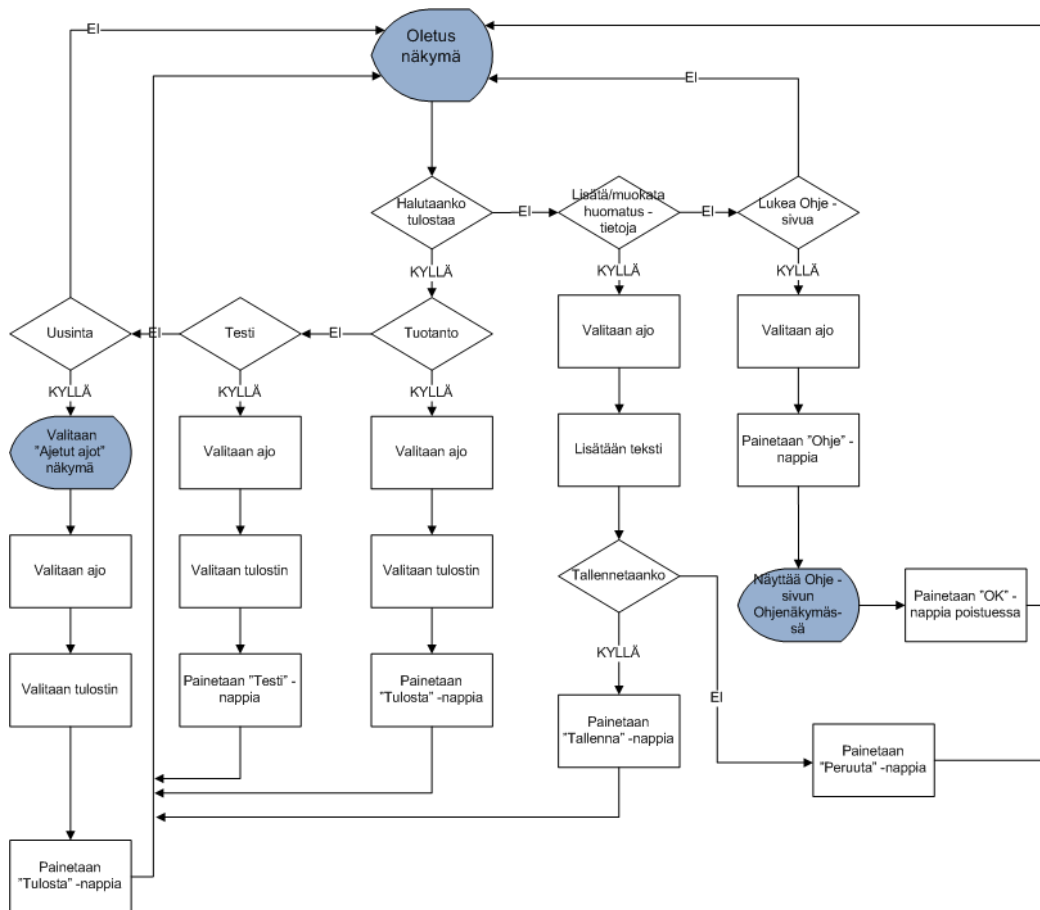
Selainpuolen koodi toteutettiin Adobe Flexin merkintäkielellä (MXML) ja ActionScript3:llä. Rakenne, graafinen ulkoasu ja palvelinkyselyt tuotettiin MXML-merkintäkielellä. ActionScriptillä tuotettiin kaikki toiminnallisuus ja tiedonkäsittely. Myös ActionScriptillä luotiin interaktiivisia toimintoja, esim. virheilmoitukset.

Alla on kuva Tuotannon Spoolerin (kuva 2) selainnäkömystä. Kuva on otettu projektin alkuvaiheesta ja kyseessä oli prototyyppi.



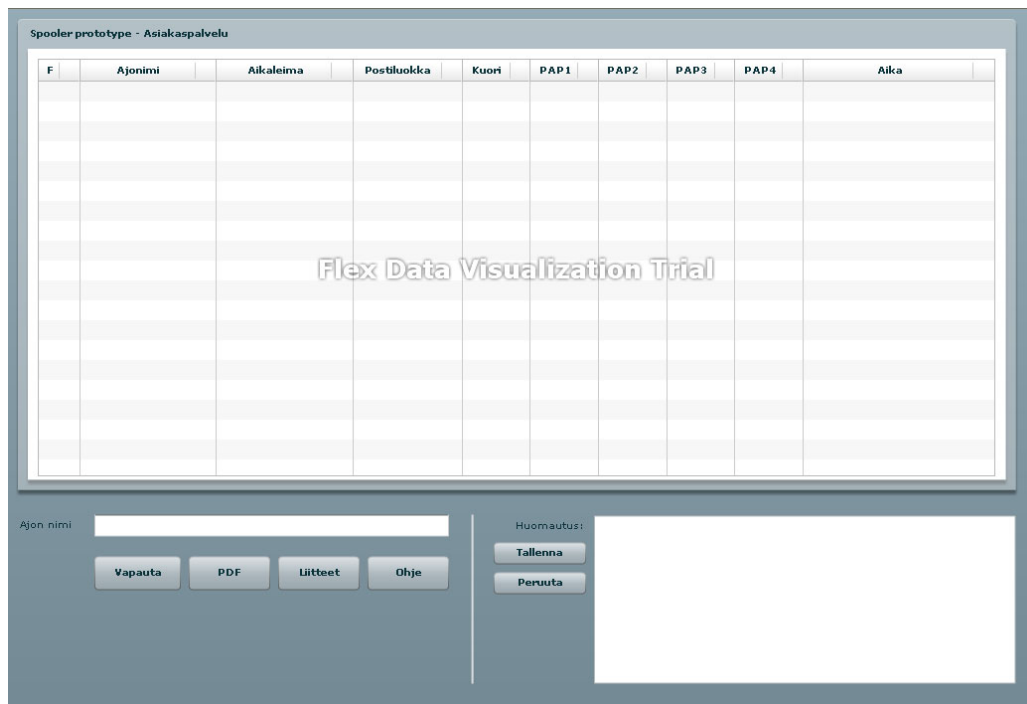
Kuva 2. Selainnäkömää Tuotannon Spoolerista.

Seuraavaksi avataan Tuotannon Spoolerin toimintaa kaavion (kuva 3) avulla josta selviää, kuinka tietyt toiminnot suoritetaan. Tässä (kuva 3) kaaviossa on esitetty selaimen toiminnallisuutta. Kaaviossa ei ole esitetty, mitä ”Tulosta” –napin painallukseen jälkeen tapahtuu palvelimella.



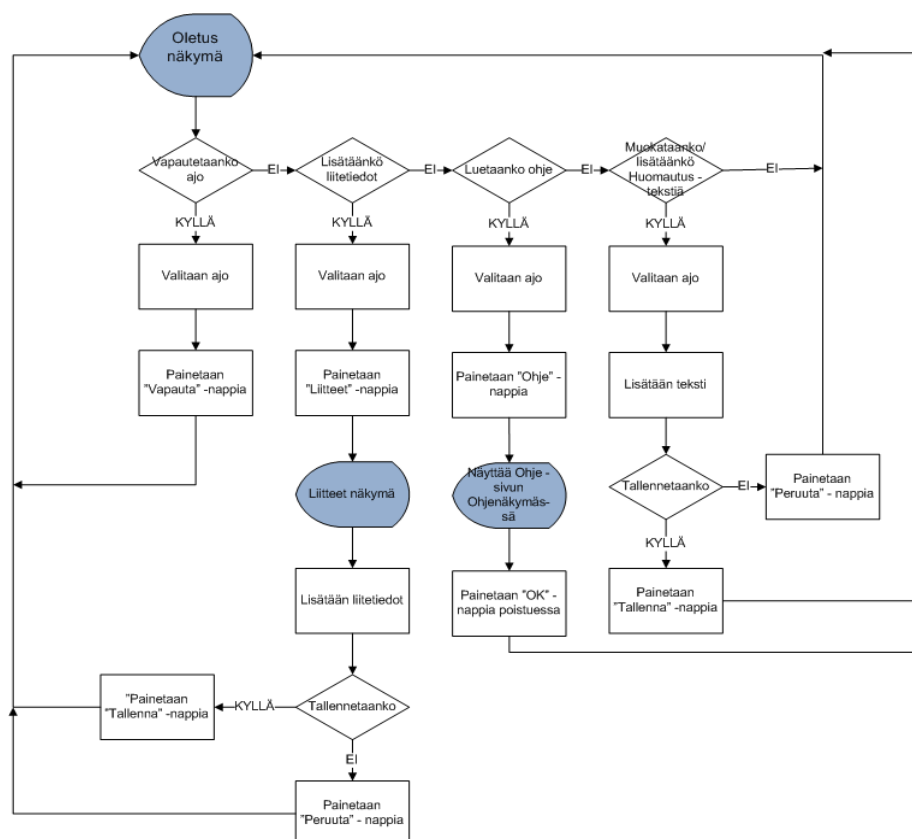
Kuva 3. Kaavionäkymä Tuotannon Spoolerista.

Seuraava kuva (kuva 4) esittää Asiakaspalvelun selainnäkömää. Kuva on otettu projektin alkuvaiheesta ja kyseessä oli prototyyppi.



Kuva 4. Selainnäkö Asiakaspalvelun Spoolerista.

Seuraava kuva (kuva 5) esittelee Asiakaspalvelun Spoolerin toimintaa kaavion muodossa.



Kuva 5. Kaavionäkymä Tuotannon Spoolerista.

2.5 Tietokanta

Sovelluksen kaikki tiedot tallennetaan MySQL-tietokantaan. Alkuvaiheessa sovelluksella oli vain yksi taulu, mistä kaikki tieto löytyi. Mutta projektin alussa tuli selväksi, että tarvitaan useampi taulu. Seuraavissa kappaleissa kerrotaan enemmän tauluista ja niiden sisällöistä.

2.5.1 Taulu: Spooler_ajot

Taulu on kuvattu opinnäytetyön liitteessä (liite 1). Tämä taulu sisältää kaikki ns. käytössä olevat ajot ja niiden tiedot. Tästä taulusta haetaan käyttöliittymän näytölle data ja halutut tiedot, mitkä halutaan näyttää käyttäjälle. Aina kun järjestelmään tulee uusi ajo, se lisätään tähän tauluun. Lisäyksen yhteydessä haetaan Ajot -taulusta vakiotietoja, jotka lisätään ajon tietoihin. Vakiotietoihin kuuluu esim. palveluneuvojen nimi, paperien mallikoodi ja kuoren mallikoodi. Tätä taulua siivotaan 14 päivän välein, jotta taulun koko pysyy kohtuullisena.

2.5.2 Taulu: Ajot

Taulu on kuvattu opinnäytetyön liitteessä (liite 2). Tämä taulu sisältää töiden ns. vakiotietoja, jotka kopioidaan uusille ajoille. Aina kun uusi ajo tulee järjestelmään, niin se näkyy punaisella värillä. Tämä kertoo siitä, että sitä ei löydy vielä tästä taulusta, ja palveluneuvojan täytyy syöttää vakiotiedot ja tallentaa ne. Näitä tietoja palveluneuvoja pystyy muokkaamaan, mutta ne eivät vaikuta jo järjestelmässä oleviin ajoihin. Vaikutukset tulevat voimaan, kun uusi ajo tulee järjestelmään ja sille haetaan vakiotiedot tästä taulusta.

2.5.3 Taulu: Palveluneuvojat

Tämä taulu (taulukko 2) sisältää palveluneuvojat. Tätä taulua käytetään käyttöliittymän näkymän suodatustyökalun tietolähteenä ja työn vakiotietojen lisäämisessä käytettävän drop-down-valikon tietolähteenä.

Taulukko 2 selittää taulun koostumuksen ja termit.

Taulukko 2. Taulun termien selitykset ja koostumus.

| Nimi | Selitys | Tyyppi |
|-------------|-----------------------------|------------------|
| <i>id</i> | <i>Uniikki tunnus</i> | <i>BIGINT</i> |
| <i>nimi</i> | <i>Palveluneuvojan nimi</i> | <i>VARCHAR20</i> |

2.6 Käyttötapaukset

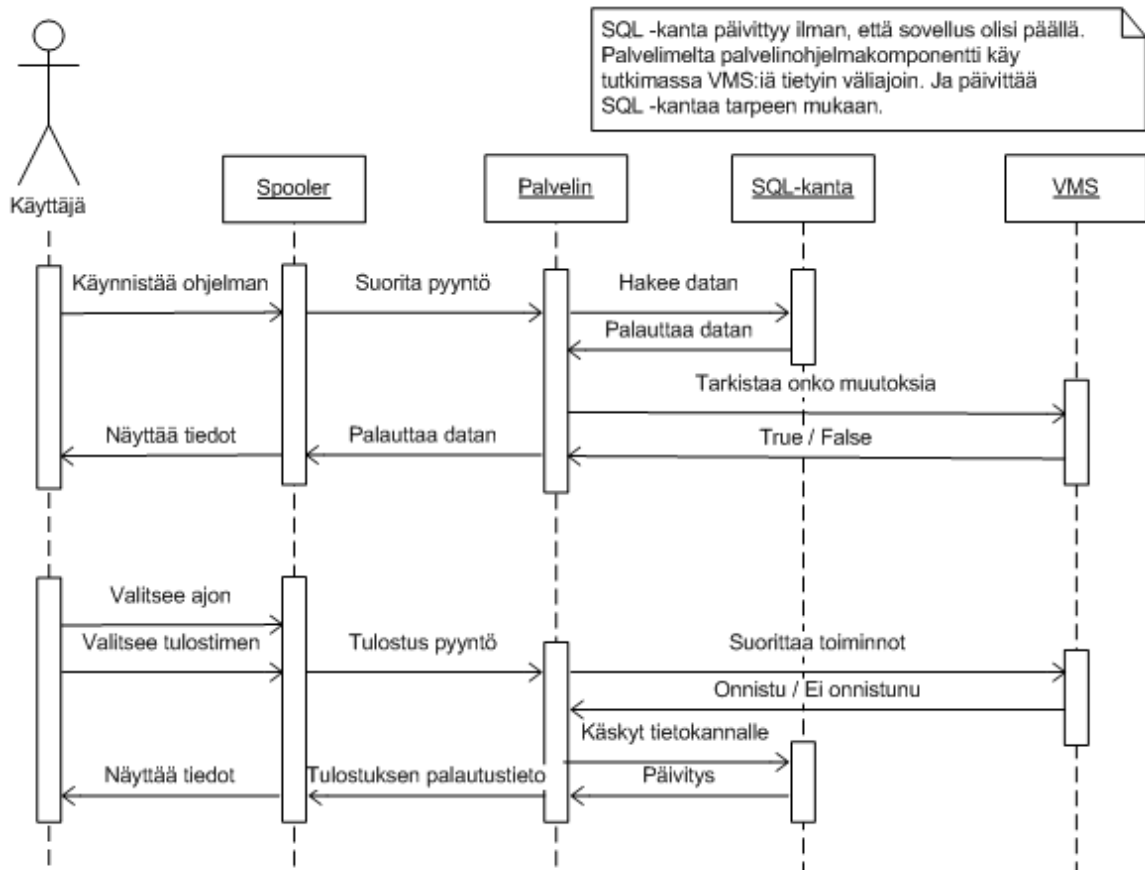
Tässä kappaleessa esittelen käyttötapauksen muodossa käyttöliittymän toiminnallisuutta.

2.6.1 Käyttötapaus # 1 – Tuotannon Spooler – Tulostaminen

Taulukko 3 tuo esille tulostamisen käyttötapauksen taulukon muodossa ja kuva 6 esittää käyttötapauksen sekvenssikaavion muodossa.

Taulukko 3. Käyttötapaus # 1 – Tuotannon Spooler – Tulostaminen.

| | |
|--------------------------|--|
| Käyttötapaus | <i>Tuotannon Spoolerilla ajon tulostaminen</i> |
| Yhteenveto | |
| Tarkoitus | <i>Käyttäjä haluaa tulostaa ajon</i> |
| Lähtökohta | <i>Käyttäjä ei ole vielä käynnistänyt ohjelmaa, mutta taustalla perl-ohjelma on päivittänyt tietokantaan ajettavia ajoja lukemalla niitä VMS-tiedostopalvelimelta.</i> |
| Kuvaus | <i>Käyttäjä tulee töihin ja rupeaa purkamaan listalla olevia ajoja ohjelman avulla. Hän käynnistää ohjelman ja saa kuvaruudulleen vapaat ajot.</i> <i>Valittuaan haluamansa ajon ja tulostimen jolla hän haluaa työn tulostuvan. Käyttäjä käynnistää prosessin painamalla ”Tulosta” – nappia.</i> |
| Poikkeustapaukset | <i>Virhetilanteet</i> |
| Lopputulos | <i>Haluttu ajo tulostuu paperille</i> |



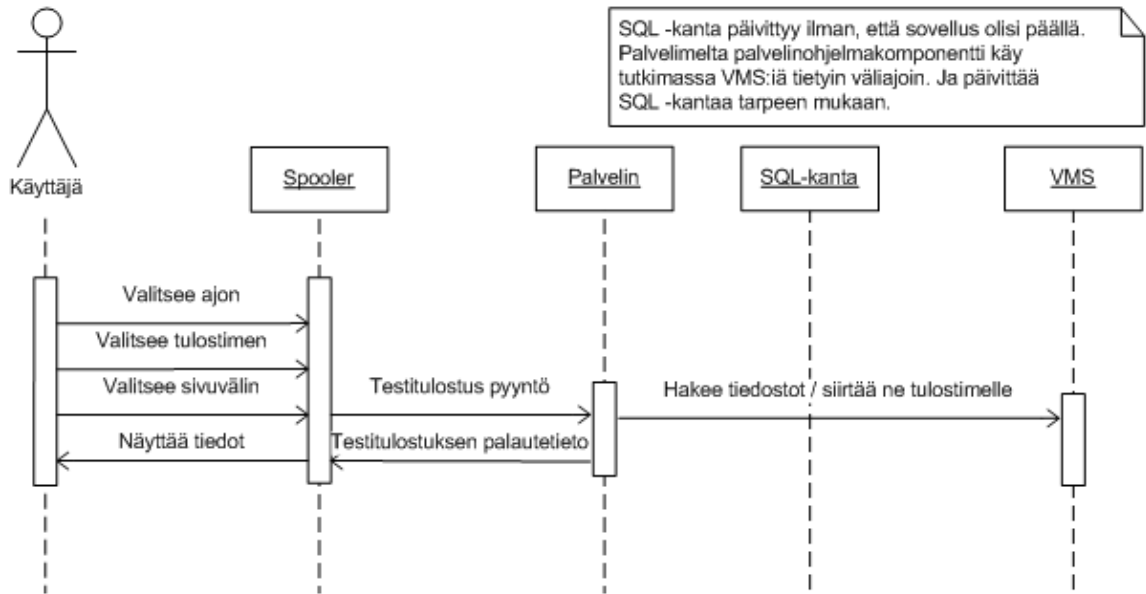
Kuva 6. Sekvenssikaavio käyttötapauksesta #1.

2.6.2 Käyttötapaus # 2 – Tuotannon Spooler – Testitulostus

Taulukko 4 tuo esille testitulostamisen käyttötapausten taulukon muodossa ja kuva 7 esittää käyttötapausten sekvenssikaavion muodossa.

Taulukko 4. Käyttötapaus # 2 – Tuotannon Spooler – Testitulostus.

| | |
|--------------------------|--|
| Käyttötapaus | <i>Tuotannon Spoolerilla testitulostaminen</i> |
| Yhteenveto | |
| Tarkoitus | <i>Käyttäjä haluaa tulostaa ajosti testitulosteen tarkistusta varten.</i> |
| Lähtökohta | <i>Käyttäjä ei ole vielä käynnistänyt ohjelmaa, mutta taustalla perl-ohjelma on päivittänyt tietokantaan ajettavia ajoja lukemalla niitä VMS-tiedostopalvelimelta.</i> |
| Kuvaus | <i>Käyttäjä haluaa tulostaa testiaineiston tietyistä ajoista. Testiaineisto tarkoittaa normaalia tulostamista, mutta vain osittain tai kokonaan ilman että esim. laskutustiedot siirtyvät järjestelmään.</i> <i>Valittuaan haluamansa ajon ja tulostimen jolla hän haluaa työn tulostuvan. Käyttäjä käynnistää prosessin painamalla ”Testi” -nappia. Tällöin avautuu ikkuna, johon hän voi syöttää sivuvälin, jolta testituloste tulostuu. Jättämällä syöttökentän tyhjäksi, tulostuu koko ajo.</i> |
| Poikkeustapaukset | <i>Virhetilanteet</i> |
| Lopputulokset | <i>Haluttu ajo tulostuu paperille</i> |



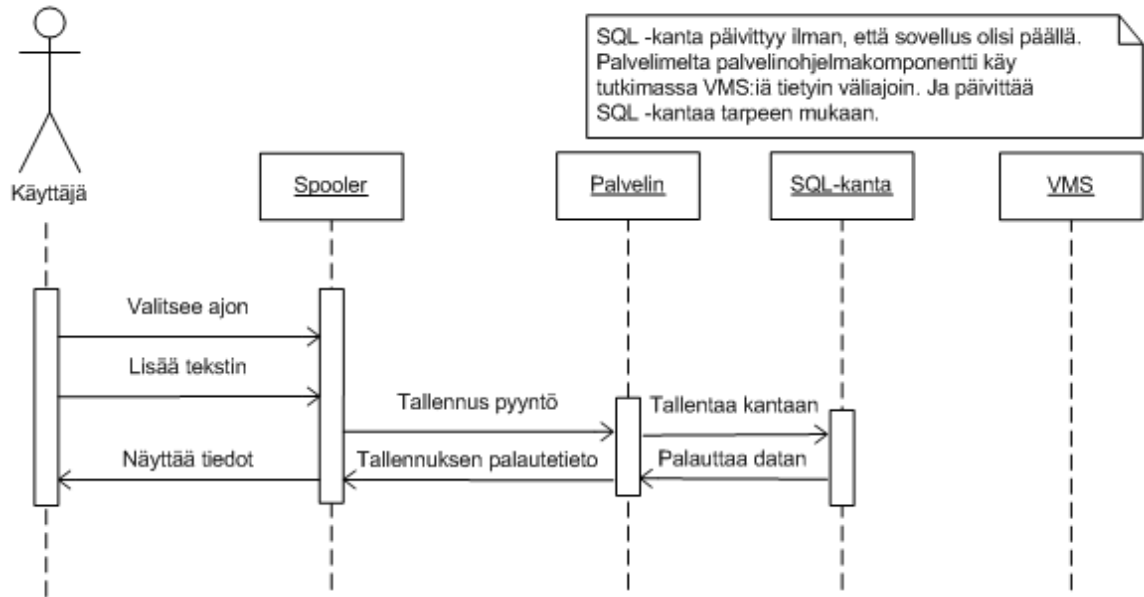
Kuva 7. Sekvenssikaavio Käyttötapaus #2.

2.6.3 Käyttötapaus # 3 – Spooler – Huomioitava-kentän muokkaus

Taulukko 5 tuo esille Huomioitava-kentän käyttötapausten taulukon muodossa ja kuva 8 esittää käyttötapausten sekvenssikaavion muodossa.

Taulukko 5. Käyttötapaus # 3 – Spooler – Huomioitava-kentän muokkaus.

| | |
|--------------------------|---|
| Käyttötapaus | <i>Spoolerilla Ohje-sivun muokkaaminen Huomioitava-kentän avulla.</i> |
| Yhteenveto | |
| Tarkoitus | <i>Asiakaspalvelu pystyy ilmoittamaan Huomioitava-kentän avulla tuotannolle, jos ajossa on jotain huomioitavaa.</i> |
| Lähtökohta | <i>Asiakaspalvelija on saanut asiakkaalta pyynnön, että tietyssä ajossa käytetään erilaista liitettä kuin yleensä.</i> |
| Kuvaus | <i>Asiakaspalvelija avaa ohjelman. Seuraavaksi hän valitsee ajon, johon hän haluaa huomautuksen lisätä. Lisää huomioitavan asian, ”Huomautus:” -tekstikenttään ja painaa tallenna. Tämän jälkeen tieto on tallentunut tietokantaan ja se on heti näkyvässä tuotannon Spoolerissa. Tämä tieto myös tulostuu ohje-sivulle.</i> |
| Poikkeustapaukset | <i>Virhetilanteet</i> |
| Lopputulos | <i>Haluttu tieto tallentuu Huomioitava-kenttään</i> |



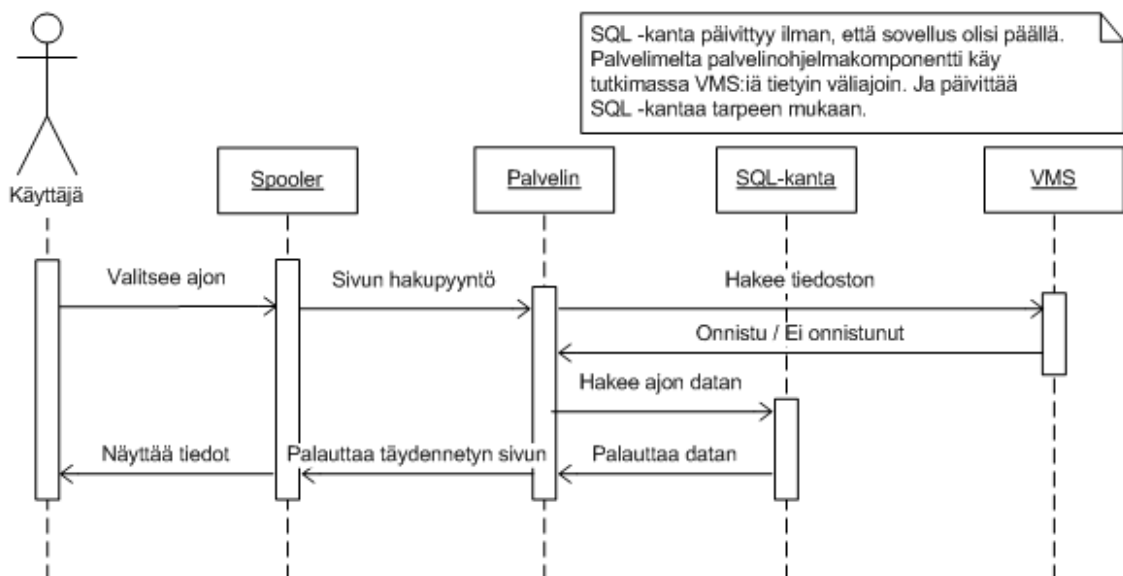
Kuva 8. Sekvenssikaavio Käyttötapaus #3.

2.6.4 Käyttötapaus #4 – Spooler – Ohje –sivun lukeminen

Taulukko 6 tuo esille Ohje-sivun lukemisen käyttötapausten taulukon muodossa ja kuva 9 esittää käyttötapausten sekvenssikaavion muodossa.

Taulukko 6. Käyttötapaus #4 – Spooler – Ohje-sivun lukeminen.

| | |
|--------------------------|--|
| Käyttötapaus | <i>Spoolerilla Ohje-sivun lukeminen</i> |
| Yhteenveto | |
| Tarkoitus | <i>Käyttäjä pääsee lukemaan Ohje-sivua.</i> |
| Lähtökohta | <i>Käyttäjä haluaa tarkistaa, onko Ohje-sivulla kaikki tarvittava tieto.</i> |
| Kuvaus | <i>Käyttäjä avaa ohjelman. Seuraavaksi hän valitsee ajon, jonka Ohje-sivua hän haluaa tarkastella. Painamalla ”Ohje” -nappia aukeaa ohje ikkunaan.</i> |
| Poikkeustapaukset | <i>Virhetilanteet</i> |
| Lopputulos | <i>Käyttäjälle avautuu Ohje-sivu ruudulle.</i> |



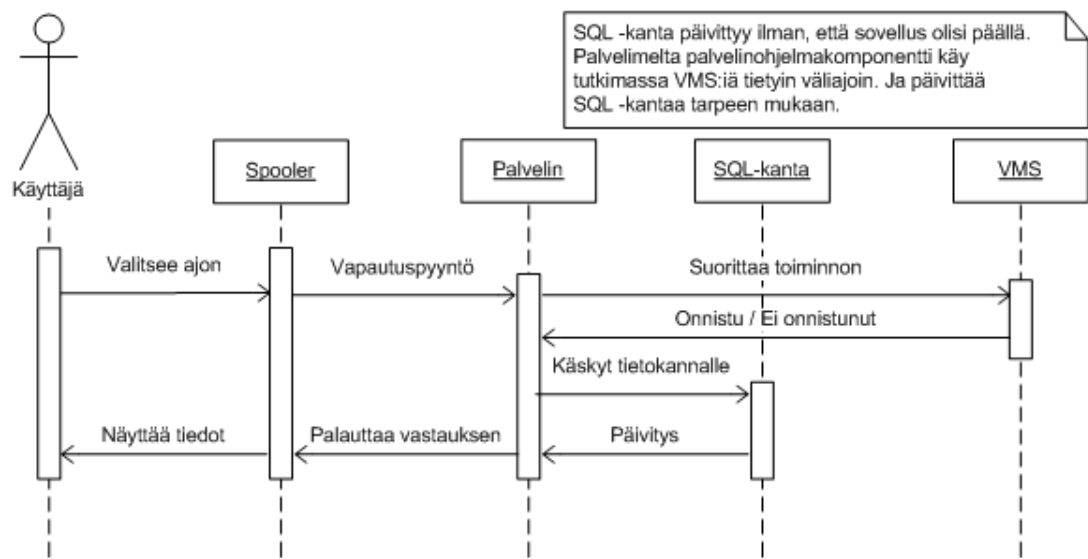
Kuva 9. Sekvenssikaavio Käyttötapaus #4.

2.6.5 Käyttötapaus # 5 – Asiakaspalvelun Spooler – Ajon vapautus

Taulukko 7 tuo esille ajon vapauttamisen käyttötapausten taulukon muodossa ja kuva 10 esittää käyttötapausten sekvenssikaavion muodossa.

Taulukko 7. Käyttötapaus # 5 – Asiakaspalvelun Spooler – Ajon vapautus.

| | |
|--------------------------|--|
| Käyttötapaus | <i>Asiakaspalvelun Spoolerilla ajon vapauttaminen</i> |
| Yhteenveto | |
| Tarkoitus | <i>Asiakaspalvelija antaa luvan tuotannolle tulostaa ajon.</i> |
| Lähtökohta | <i>Asiakaspalvelija on saanut asiakkaalta luvan tulostaa ajon, joka on ollut odottamassa ohjelmassa lupaa.</i> |
| Kuvaus | <i>Asiakaspalvelija avaa ohjelman, jolla hän voi vapauttaa ajoja ja antaa lisäohjeita tuotannolle.</i> <i>Valittuaan ajon listalta, hän tarkistaa ajon tiedot ja painaa ”Vapauta ajo” -nappia. Ajontila muuttuu ”lupaa odottava”-tilasta ”vapaa”-tilaan. Näin Spooler osaa näyttää kyseisen työn tuotannossa.</i> |
| Poikkeustapaukset | <i>Virhetilanteet</i> |
| Lopputulokset | <i>Ajon tila muuttuu ja se siirtyy vapaisiin ajoihin.</i> |



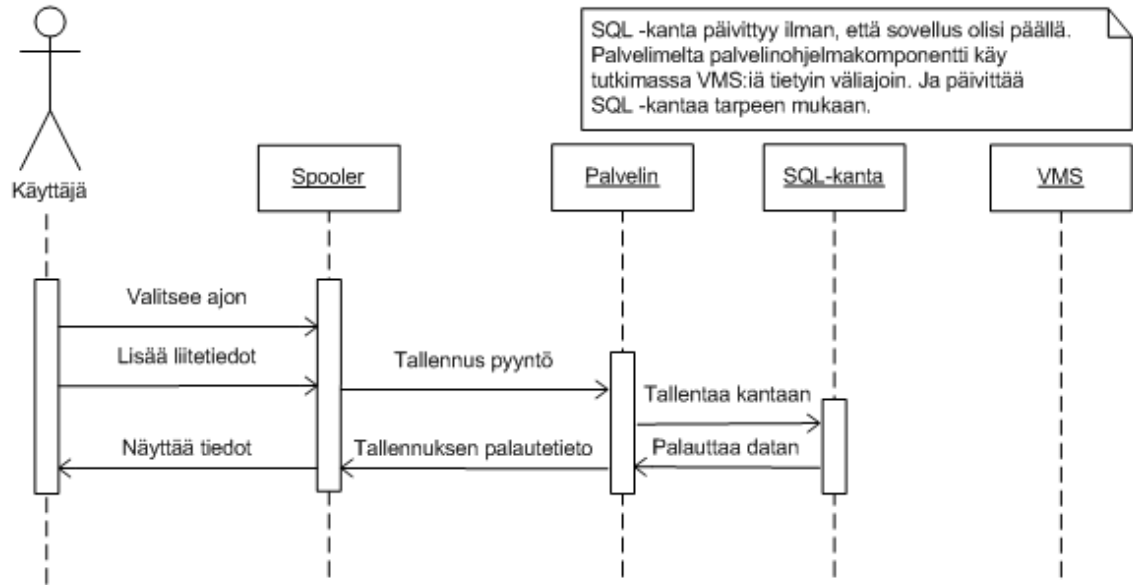
Kuva 10. Sekvenssikaavio Käyttötapaus #5.

2.6.6 Käyttötapaus # 6 – Asiakaspalvelun Spooler – Liitteiden ilmoittaminen

Taulukko 8 tuo esille liitteiden ilmoittamisen käyttötapausten taulukon muodossa ja kuva 11 esittää käyttötapausten sekvenssikaavion muodossa.

Taulukko 8. Käyttötapaus # 6 – Asiakaspalvelun Spooler – Liitteiden ilmoittaminen.

| | |
|--------------------------|---|
| Käyttötapaus | <i>Asiakaspalvelun Spoolerilla liitteiden ilmoittaminen</i> |
| Yhteenveto | |
| Tarkoitus | <i>Asiakaspalvelija lisää ajolle liitteet.</i> |
| Lähtökohta | <i>Asiakaspalvelija on saanut asiakkaalta tiedoksi, että kyseiseen ajoon liitetään tiettyjä liitteitä.</i> |
| Kuvaus | <i>Asiakaspalvelija avaa ohjelman, jolla hän voi lisätä liitetiedot</i> <i>Valittuaan ajon listalta, käyttäjä painaa ”Liitteet”-nappia. Jolloin aukeaa ikkuna, johon voi lisätä 10 kpl liitteitä. Liitteiden nimen lisäksi, käyttäjä valitsee onko liite pakkoliite vai valinnainen liite. Sen jälkeen hän painaa ”Tallenna”-nappia.</i> |
| Poikkeustapaukset | <i>Virhetilanteet</i> |
| Lopputulos | <i>Liitteet tallentuvat ajon tietoihin.</i> |



Kuva 11. Sekvenssikaavio Käyttötapaus #6.

2.7 Testaus

Käyttöliittymää testattiin paikallisesti testiympäristössä, minkä jälkeen siirryttiin tuotannolliseen testaamiseen. Tämä testaaminen tapahtui vanhan käyttöliittymän rinnalla. Alkuvaiheessa käyttöliittymään siirrettiin yksinkertaisia töitä, joitten läpimenoa oli helppoa valvoa. Näin virheet havaittiin helpommin. Pikku hiljaa siirrettiin töitä vanhasta järjestelmästä Spooleriin, ja kun oltiin varmoja käyttöliittymän luotettavuudesta, siirrettiin kaikki ajot käyttöliittymän alaisuuteen.

Testausvaiheessa käyttäjiltä kerättiin palautetta, jonka perusteella pyrittiin mahdollisimman paljon parantamaan sovellusta. Jotkin kehitystoiveet siirtyivät seuraaviin vaiheisiin, mutta suurin osa saatiin sisällytyksi ensimmäiseen.

Suurin ongelma oli palvelimen riittämätön teho käsitellä monta Perl-prosessia kerrallaan. Tämä ongelma korjattiin parantamalla Perl-koodia Mod_perl-moduulilla. Tämän ansiosta käynnistyy vain yksi prosessi. Tällöin palvelin ei tarvitse priorisoida prosesseja vaan pystyy antamaan yhdelle prosessille tarvittavan määrän resursseja toimiakseen mahdollisimman hyvin.

3 YHTEENVETO

Projektin tarkoitus oli luoda käyttöliittymä tuotannolliseen käyttöön. Käyttäjryhmiä oli tuotanto ja asiakaspalvelu, joille kummallekin tehtiin oma web-sovellus.

Käyttöliittymien selainpuolen osuus tehtiin Adobe Flex-tuotantotyökalulla ja palvelinohjelmistokoodi tehtiin Perl-kielellä. Kaikki data tallennettiin MySQL-tietokantaan.

Projekti alussa tehdyllä käyttäjäryhmien haastattelulla kerättiin vaatimuksia ja tarpeita. Tämän pohjalta suunniteltiin toiminnallinen määrittely, joka hyväksytettiin kaikilla osapuolilla. Seuraavaksi aloitettiin ohjelmointityö joka on projektin pisin vaihe. Ohjelmointityön eri vaiheissa kysyttiin käyttäjien mielipiteitä. Näin pyrin varmistamaan, että suunta oli oikea. Työn edetessä testasin myös kaikki toiminnallisuudet virheiden välttämiseksi.

Seuraavaksi siirryttiin tuotannolliseen testaamiseen, jolla haluttiin poistaa viimeiset virheet. Tämä tapahtui käyttämällä sovellusta tuotannollisesti vanhan käyttöliittymän rinnalla. Tämän vaiheen jälkeen sovellus otettiin tuotannolliseen käyttöön.

Käyttöliittymälle on olemassa jo jatkokehitysmäärittely, jossa luodaan enemmän toiminnallisuuksia, esimerkiksi hakutoiminto, ajojen rajausta palveluneuvojen nimellä ja virheiden tallentaminen ajon tietoihin ja näiden esittäminen. Nämä kehityspiirteet toteutetaan seuraavaksi.

Mielestä projekti onnistui hyvin, ja käyttäjäryhmien palaute on ollut hyvää. Sovellus on auttanut päivittäistä työntekoa ja lisännyt näin tehokkuutta.

LÄHTEET

- 1 Jack D . Herrington & Emily Kim. (2008). Getting Started with Flex 3. Canada. O'Reilly Media, Inc.
- 2 Colin Mook. (2007). Essential ActionScript 3.0. United States of America. O'Reilly Media, Inc.
- 3 Perl. Wikipedia, vapaa tietosanakirja. Luettu 31.5.2010. [verkkodokumentti]
<http://fi.wikipedia.org/wiki/Perl>
- 4 MySQL. Wikipedia, vapaa tietosanakirja. Luettu 31.5.2010. [verkkodokumentti]
<http://fi.wikipedia.org/wiki/Mysql>

| Nimi | Selitys | Tyyppi |
|--------------------------------|--|-------------------|
| <i>Data_id</i> | <i>Uniikki tunnus ajolla</i> | <i>BIGINT</i> |
| <i>Data_name</i> | <i>Ajon nimi</i> | <i>VARCHAR 20</i> |
| <i>Data_timestamp</i> | <i>Ajon aikaleima</i> | <i>VARCHAR 20</i> |
| <i>Data_actiontimestamp</i> | <i>Ajon aikaleima, milloin ajo on siirretty kyseiseen tauluun. Toimii myös samalla aikaleimana, milloin ajo on ajettu.</i> | <i>TIMESTAMP</i> |
| <i>Data_postclass</i> | <i>Ajon postiluokka</i> | <i>VARCHAR 10</i> |
| <i>Data_resource_envelope</i> | <i>Ajon kuorikoodi</i> | <i>VARCHAR 10</i> |
| <i>Data_resource_paper1</i> | <i>Ajon pap1 koodi</i> | <i>VARCHAR 10</i> |
| <i>Data_resource_paper2</i> | <i>Ajon pap2 koodi</i> | <i>VARCHAR 10</i> |
| <i>Data_resource_paper3</i> | <i>Ajon pap3 koodi</i> | <i>VARCHAR 10</i> |
| <i>Data_resource_paper4</i> | <i>Ajon pap4 koodi</i> | <i>VARCHAR 10</i> |
| <i>Data_envelope_count</i> | <i>Kuorien määrä</i> | <i>INT</i> |
| <i>Data_paper1_count</i> | <i>Pap1 määrä</i> | <i>INT</i> |
| <i>Data_paper2_count</i> | <i>Pap2 määrä</i> | <i>INT</i> |
| <i>Data_paper3_count</i> | <i>Pap3 määrä</i> | <i>INT</i> |
| <i>Data_paper4_count</i> | <i>Pap4 määrä</i> | <i>INT</i> |
| <i>Data_notice</i> | <i>Ajon huomautus teksti</i> | <i>TEXT</i> |
| <i>Data_attachment1</i> | <i>Ajon liitteen1 tieto</i> | <i>VARCHAR 20</i> |
| <i>Data_attachment1_type</i> | <i>Liitteen1 tyyppi</i> | <i>VARCHAR 10</i> |
| <i>Data_attachment1_weight</i> | <i>Liitteen paino</i> | <i>INT</i> |
| <i>Data_attachment2</i> | <i>Ajon liitteen2 tieto</i> | <i>VARCHAR 20</i> |
| <i>Data_attachment2_type</i> | <i>Liitteen2 tyyppi</i> | <i>VARCHAR 10</i> |
| <i>Data_attachment2_weight</i> | <i>Liitteen paino</i> | <i>INT</i> |
| <i>Data_attachment3</i> | <i>Ajon liitteen3 tieto</i> | <i>VARCHAR 20</i> |
| <i>Data_attachment3_type</i> | <i>Liitteen3 tyyppi</i> | <i>VARCHAR 10</i> |
| <i>Data_attachment3_weight</i> | <i>Liitteen paino</i> | <i>INT</i> |
| <i>Data_attachment4</i> | <i>Ajon liitteen4 tieto</i> | <i>VARCHAR 20</i> |
| <i>Data_attachment4_type</i> | <i>Liitteen4 tyyppi</i> | <i>VARCHAR 10</i> |
| <i>Data_attachment4_weight</i> | <i>Liitteen paino</i> | <i>INT</i> |
| <i>Data_attachment5</i> | <i>Ajon liitteen5 tieto</i> | <i>VARCHAR 20</i> |

| | | |
|---------------------------------|--|-------------------|
| <i>Data_attachment5_type</i> | <i>Liitteen5 tyyppi</i> | <i>VARCHAR 10</i> |
| <i>Data_attachment5_weight</i> | <i>Liitteen paino</i> | <i>INT</i> |
| <i>Data_attachment6</i> | <i>Ajon liitteen6 tieto</i> | <i>VARCHAR 20</i> |
| <i>Data_attachment6_type</i> | <i>Liitteen6 tyyppi</i> | <i>VARCHAR 10</i> |
| <i>Data_attachment6_weight</i> | <i>Liitteen paino</i> | <i>INT</i> |
| <i>Data_attachment7</i> | <i>Ajon liitteen7 tieto</i> | <i>VARCHAR 20</i> |
| <i>Data_attachment7_type</i> | <i>Liitteen7 tyyppi</i> | <i>VARCHAR 10</i> |
| <i>Data_attachment7_weight</i> | <i>Liitteen paino</i> | <i>INT</i> |
| <i>Data_attachment8</i> | <i>Ajon liitteen8 tieto</i> | <i>VARCHAR 20</i> |
| <i>Data_attachment8_type</i> | <i>Liitteen8 tyyppi</i> | <i>VARCHAR 10</i> |
| <i>Data_attachment8_weight</i> | <i>Liitteen paino</i> | <i>INT</i> |
| <i>Data_attachment9</i> | <i>Ajon liitteen9 tieto</i> | <i>VARCHAR 20</i> |
| <i>Data_attachment9_type</i> | <i>Liitteen9 tyyppi</i> | <i>VARCHAR 10</i> |
| <i>Data_attachment9_weight</i> | <i>Liitteen paino</i> | <i>INT</i> |
| <i>Data_attachment10</i> | <i>Ajon liitteen10 tieto</i> | <i>VARCHAR 20</i> |
| <i>Data_attachment10_type</i> | <i>Liitteen10 tyyppi</i> | <i>VARCHAR 10</i> |
| <i>Data_attachment10_weight</i> | <i>Liitteen paino</i> | <i>INT</i> |
| <i>Data_signature</i> | <i>Ajoa viimeksi muokanneen tunnistetieto</i> | <i>VARCHAR 20</i> |
| <i>Data_condition</i> | <i>Ajon tilanne (vapaa, lupaa odottava, tulostettu, siirto, käsitettely/virhe)</i> | <i>VARCHAR 20</i> |
| <i>Data_last_action</i> | <i>Ajolle viimeksi suoritettu toimenpide (testi/tuotanto/uusinta)</i> | <i>VARCHAR 20</i> |
| <i>Data_new_job</i> | <i>Ilmoittaa, onko ajo uusi vai tuttu Spoolerille</i> | <i>VARCHAR 10</i> |
| <i>Data_color</i> | <i>Ajon väritieto(ei vielä ensimmäisessä versiossa käytössä)</i> | <i>VARCHAR10</i> |
| <i>Data_palveluneuvoja</i> | <i>Ajon lajittelu tieto, jossa kerrotaan ajoa hallinnoivan henkilön nimi.</i> | <i>VARCHAR20</i> |
| <i>Data_modify_timestamp</i> | <i>Ajon lisäyspäivämäärä. Käytetään tietokanta puhdistuksen tietueena.</i> | <i>TIMESTAMP</i> |
| <i>Data_niputus_lukko</i> | <i>0 tai 1. Kertoo onko ajolla niputusta estävä lukitus päällä (ei vielä ensimmäisessä versiossa käytössä)</i> | <i>VARCHAR10</i> |
| <i>Data_error_message</i> | <i>Virhetilanteiden viesti</i> | <i>TEXT</i> |

| Nimi | Selitys | Tyyppi |
|--------------------------|---|------------------|
| <i>id</i> | <i>Uniikki tunnus</i> | <i>BIGINT</i> |
| <i>name</i> | <i>nimi</i> | <i>VARCHAR20</i> |
| <i>Attachment1</i> | <i>Liitteen 1 tieto</i> | <i>VARCHAR20</i> |
| <i>Attachment2</i> | <i>Liitteen 2 tieto</i> | <i>VARCHAR20</i> |
| <i>Attachment3</i> | <i>Liitteen 3 tieto</i> | <i>VARCHAR20</i> |
| <i>Attachment4</i> | <i>Liitteen 4 tieto</i> | <i>VARCHAR20</i> |
| <i>Attachment5</i> | <i>Liitteen 5 tieto</i> | <i>VARCHAR20</i> |
| <i>Attachment6</i> | <i>Liitteen 6 tieto</i> | <i>VARCHAR20</i> |
| <i>Attachment7</i> | <i>Liitteen 7 tieto</i> | <i>VARCHAR20</i> |
| <i>Attachment8</i> | <i>Liitteen 8 tieto</i> | <i>VARCHAR20</i> |
| <i>Attachment9</i> | <i>Liitteen 9 tieto</i> | <i>VARCHAR20</i> |
| <i>Attachment10</i> | <i>Liitteen 10 tieto</i> | <i>VARCHAR20</i> |
| <i>Attachment1_type</i> | <i>Liitteen 1 tyyppi</i> | <i>VARCHAR10</i> |
| <i>Attachment2_type</i> | <i>Liitteen 2 tyyppi</i> | <i>VARCHAR10</i> |
| <i>Attachment3_type</i> | <i>Liitteen 3 tyyppi</i> | <i>VARCHAR10</i> |
| <i>Attachment4_type</i> | <i>Liitteen 4 tyyppi</i> | <i>VARCHAR10</i> |
| <i>Attachment5_type</i> | <i>Liitteen 5 tyyppi</i> | <i>VARCHAR10</i> |
| <i>Attachment6_type</i> | <i>Liitteen 6 tyyppi</i> | <i>VARCHAR10</i> |
| <i>Attachment7_type</i> | <i>Liitteen 7 tyyppi</i> | <i>VARCHAR10</i> |
| <i>Attachment8_type</i> | <i>Liitteen 8 tyyppi</i> | <i>VARCHAR10</i> |
| <i>Attachment9_type</i> | <i>Liitteen 9 tyyppi</i> | <i>VARCHAR10</i> |
| <i>Attachment10_type</i> | <i>Liitteen 10 tyyppi</i> | <i>VARCHAR10</i> |
| <i>Notice</i> | <i>Huomautusteksti</i> | <i>TEXT</i> |
| <i>Palveluneuvoja</i> | <i>Palveluneuvojan nimi</i> | <i>VARCHAR20</i> |
| <i>Color</i> | <i>Väritieto(ei käytössä ensimmäisessä versiossa)</i> | <i>VARCHAR10</i> |
| <i>Niputus_lukko</i> | <i>0 tai 1, kertoo onko niputusta estävä lukitus päällä (ei käytössä ensimmäisessä versiossa)</i> | <i>VARCHAR10</i> |