



SAVONIA

OPINNÄYTETYÖ - AMMATTIKORKEAKOULUTUTKINTO
TEKNIIKAN JA LIIKENTEEN ALA

JÄSENREKISTERIN MIG- RAATIO KONGRESSI-JÄR- JESTELMÄÄN

TEKIJÄ/T: Mikko Väätäinen

Koulutusala Tekniikan ja liikenteen ala			
Koulutusohjelma/Tutkinto-ohjelma Tietotekniikan koulutusohjelma			
Työn tekijä(t) Mikko Väätäinen			
Työn nimi Jäsenrekisterin migraatio Kongressi järjestelmään			
Päiväys	8. toukokuuta 2018	Sivumäärä/Liitteet	23
Ohjaaja(t) Lehtorit Jussi Koistinen ja Keijo Kuosmanen			
Toimeksiantaja/Yhteistyökumppani(t) Data Prisma Oy			
Tiivistelmä <p>Opinnäytetyö toteutettiin yhteistyössä Data Prisma Oy:n kanssa. Tavoitteena oli päivittää vanhentuneella teknikalla toteutettu Jäsenrekisteri ja sen ominaisuudet osaksi nykyistä Kongressi-järjestelmää, jotta kyseiset ominaisuudet saadaan helpommin ylläpidettäviksi sekä nykyisen kehityksen pariin.</p> <p>Opinnäytetyö aloitettiin suunnittelemalla sekä toteuttamalla Jäsenrekisterin perusominaisuuksien, kuten jäsenrekisterin luonnin mahdollistaminen, migraation Kongressi-järjestelmään. Hyvin nopeasti tämän jälkeen huomattiin, että on tärkeämpiä ominaisuuksia, jotka täytyi tuoda osaksi Kongressia, esimerkiksi erilaisten lasku- ja reskontra-aineistojen tulostamisen mahdollistaminen sekä GDPR tietosuojauudistuksen vaatimia muutoksia. Työn tekemiseen käytettiin Visual Studio 2017 sekä ASP.Net MVC ja .Net frameworkia. Nämä menetelmät valikoituivat sen takia, koska ne olivat käytössä jo muualla kehityksessä.</p> <p>Lopputuloksena saatiin asiakkaille tärkeimmät ominaisuudet valmiiksi laskutuksen sekä reskontra-aineistojen osalta sekä tuotiin Kongressiin erilaisten tietosuoja-asetusten ylläpito-ominaisuus. Jäsenrekisterin migraatiosta jäi toteuttamatta ominaisuuksia, jotka tullaan toteuttamaan jatkokehitysvaiheessa.</p>			
Avainsanat ASP.Net MVC, .NET Framework, C#			

Field of Study Technology, Communication and Transport			
Degree Programme Degree Programme in Information Technology			
Author(s) Mikko Väätäinen			
Title of Thesis Member Register Migration to Kongress			
Date	8 May 2018	Pages/Appendices	23
Supervisor(s) Mr. Jussi Koistinen, Senior Lecturer and Mr. Keijo Kuosmanen, Senior Lecturer			
Client Organisation /Partners Data Prisma Oy			
<p>Abstract</p> <p>The purpose of this thesis was to migrate Member register as part of a more modern Kongress system for Data Prisma Oy. The aim was to transfer the features of the Member register to a technologically more modern system. That would allow the features to be much easier to maintain and they would be part of the current development chain.</p> <p>The work began by planning and creating the most basic features into the Kongress system; for example, making it possible to create a member register. Very quickly it was noticed that there were far more important features that needed to be integrated in the Kongress; for example, printing invoice and ledger data in different formats. The upcoming General Data Protection Regulations also had to be addressed at a fast pace. It was decided that the development will be made by using Visual Studio 2017 and the ASP.Net MVC with .Net framework; C# as the main programming language. These techniques were selected mainly since they were already used in other parts of the development chain.</p> <p>As a result, the main features for the customer were finished including the printing of invoices and ledger records. Also, for the data protection regulations a maintenance feature was implemented in the users' privacy settings. Some of the Member register's features were not migrated at this stage and they will be migrated in the upcoming development stages.</p>			
<p>Keywords</p> <p>ASP.Net MVC, .NET Framework, C#</p>			

ESIPUHE

Tämä opinnäytetyö tehtiin yhteistyössä Data Prisma Oy:n kanssa kevään 2018 aikana Kuopiossa. Haluan kiittää työnantajaani Data Prisma Oy:tä toimeksiannosta sekä erityisesti Jukka-Pekka Kurkea toimimisesta opinnäytetyön ohjaajana sekä Jussi Tuovista toimimisesta opinnäytetyön valvojana.

Suuri kiitos kuuluu myös opinnäytetyön ohjaavalla opettajalla lehtori Jussi Koistiselle.

Kuopiossa 10.6.2018

Mikko Väättäinen

SISÄLTÖ

1	JOHDANTO	6
1.1	Työn tavoite	6
1.2	Lyhenteet ja määritelmät.....	7
2	SUUNNITTELU JA KEHITYSMENETELMÄT	8
2.1	Kehitysmenetelmät	8
2.1.1	ASP.Net MVC	8
2.1.2	MVC-arkkitehtuuri	8
2.1.3	Kehitysympäristö.....	9
2.1.4	Tietokanta	9
3	TOTEUTUS.....	10
3.1	Jäsenrekisteri.....	10
3.2	Laskutusaineisto	12
3.3	Reskontra-aineisto	16
3.3.1	AX-aineisto	16
3.3.2	Reskontra-aineiston tulostuksen automatisointi	17
3.4	Pankkitunnistautuminen	18
3.5	GDPR tietosuojan muutoksia	20
4	YHTEENVETO.....	22
	LÄHTEET JA TUOTETUT AINEISTOT	23

1 JOHDANTO

Jäsenrekisteri on vanha yhdistysten- ja jäsentenhallintajärjestelmä, jonka ylläpidettävyys oli tullut vaikeaksi vanhentuneen teknologian myötä. Kongressi on selainpohjainen tapahtumanhallintajärjestelmä, jonka avulla voidaan hallinnoida kaikkea tapahtumien järjestämiseen liittyviä asioita kuten esimerkiksi ilmoittautumisia, myyntiä, laskutusta, asiakastietojen ylläpitoa ja raportointia.

Opinnäytetyön aiheena oli siirtää Jäsenrekisterin ominaisuudet osaksi uutta Kongressi-järjestelmää. Jäsenrekisteri on vanhalla tekniikalla tehty Data Prisman tuote, jonka tuki päättyy palvelintekniikan päivittyessä. Tästä syystä on syntynyt tarve siirtää Jäsenrekisterin ominaisuudet osaksi uutta Kongressi-järjestelmää. Opinnäytetyö sai alkunsa asiakkaan tarpeesta saada käyttöönsä Jäsenrekisterin ominaisuudet uudessa Kongressissa. Työn tilaajana toimi Data Prisma Oy ja työ oli tarkoitus suorittaa työnohella scrum -viitekehyksen mukaan viikon mittaisissa sprinteissä.

Tässä raportissa käydään läpi työnsuorituksen vaiheet.

1.1 Työn tavoite

Opinnäytetyön keskeisimpinä tavoitteina on saada Jäsenrekisteri nykyaikaiseen teknologiaan, jotta myös uusien ominaisuuksien kehittäminen tulisi jatkossa onnistumaan uuden Kongressin kehityksen yhteydessä. Näin saataisiin Jäsenrekisteri helpommin saataville asiakkaille, joilla Kongressi on jo käytössä. Samalla päästäisiin hyödyntämään Kongressissa jo olevia ominaisuuksia Jäsenrekisterin kanssa sekä mahdollistetaan Jäsenrekisterin uusmyynti uusille asiakkaille. Tämän lisäksi myös kehitystyöstä tulisi jouhevampaa, kun tarvittavat ominaisuudet on niputettu yhden projektin alle.

1.2 Lyhenteet ja määritelmät

MVC = Model – View – Controller, ohjelmistosuunnittelumalli

ORM = Object-relational mapping, datan käsittelytekniikka jossa luodaan tietokannan datasta olioita

EF = Entity Framework, Object-relational mapper. Luo tietokannasta virtuaalisen olion

LINQ = Language Integrated Query, komponentti jolla on mahdollista tehdä kyselyjä esimerkiksi olioista

HTML = Hypertext Markup Language, verkkosivujen rakenteenkuvauskieli

SCRUM = Ketterässä ohjelmistokehityksessä käytetty projektihallinnan kehys

JavaScript = Websivuilla käytettävä komentosarjakieli

Sftp = SSH File Transfer Protocol, tiedostonsiirtoprotokolla

Code First = Ohjelmointityyli, jossa tietokantaa ei tarvitse olla ensin taustalla

2 SUUNNITTELU JA KEHITYSMENETELMÄT

Työhön liittyvistä vaatimuksista oli jo laadittu vaatimusmäärittelydokumentti, jonka perusteella jo olemassa oleva Jäsenrekisteri oli luotu. Opinnäytetyön suunnittelussa käytettiin samoja vaatimuksia. Projektista laadittiin projektisuunnitelma, jonka mukaan itse työtä lähdettiin tekemään.

2.1 Kehitysmenetelmät

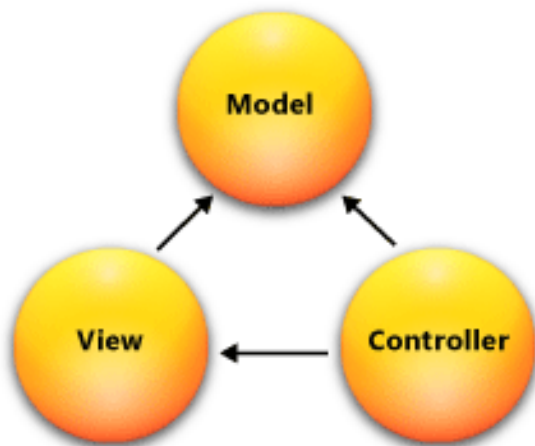
Kehityksessä käytettäviksi menetelmiksi sekä alustoiksi valikoitui luonnollisesti Data Prismassa jo käytettävät uusimmat menetelmät. Näin varmistettiin ohjelmiston helpompi ylläpidettävyyys sekä kehitys jatkossa.

2.1.1 ASP.Net MVC

Kehitys tehtiin ASP.Net MVC:llä. ASP.Net MVC on Microsoftin kehittämä websovellus framework, joka toteuttaa MVC suunnittelumallia. MVC-mallin käyttäminen mahdollistaa muun muassa suuremman kehitystiimin käytön samassa projektissa yhtäaikaaisesti, koska sovellus on jaettu eri osioihin. Näin ollen kehitystä voidaan tehdä samanaikaisesti sovelluksen eri tasoilla. (Microsoft, 2018)

2.1.2 MVC-arkkitehtuuri

MVC-suunnittelumalli käytännössä erottaa ohjelmiston käyttöliittymälogiikan ja businesslogiikan toisistaan. MVC on lyhenne sanoista Model, View ja Controller. Model käsitteellä tarkoitetaan sovelluksen datan käsittelyyn tarvittavia objekteja. Näiden objektien avulla esimerkiksi luetaan, kirjoitetaan sekä päivitetään dataa tietokannassa. View kuvaa sovelluksen käyttöliittymää, jolla esitetään data loppukäyttäjälle. Käyttöliittymä, eli näkymät, luodaan model -tasolta saadun datan avulla. Näkymä voi olla tyypillisesti HTML-sivu, joka sisältää esimerkiksi tekstilaatikoita, alasetoistoja tai valintalaatikoita datan näyttämiseen. Lyhenteessä sana Controller tarkoittaa komponenttia model -tason sekä näkymien välillä. Nämä komponentit toimivat yhteydessä modelien sekä näkymien kanssa.



Kuva 1 MVC-suunnittelumallin rakenne (Microsoft 2018.)

2.1.3 Kehitysympäristö

Kehitysympäristönä käytettiin Microsoft Visual Studio 2017 -työkalua, joka mahdollisti yhdessä Entity Frameworkin kanssa sujuvan ohjelmoinnin. EF on ORM framework, jolla voidaan muodostaa tietokannasta virtuaalinen malli (Microsoft, 2018). Tällöin tietokannan objektien käsittelystä tulee helpompaa suoraan koodissa eikä tarvitse huolehtia itse tietokantataulun rakenteesta. Tämä mahdollistaa muun muassa tietokantakyselyiden teon käyttämällä LINQ:a, jolloin kysely tehdään EF luomista tietokannan malleista eikä esimerkiksi suoralla SQL kyselyllä. Osa projektista toteutettiin Code First -tekniikalla, jossa mappaus tietokannan ja halutun objektin mallin välillä tehtiin käsin. Ohjelmointikielenä käytettiin C# sekä käyttöliittymän pieniin toiminnallisuuksiin Javascript kieliä.

2.1.4 Tietokanta

Projektin tietokantapalvelimena toimi Microsoft SQL Server, joka myös oli jo Data Prismalla käytössä ja hyväksi havaittu.

3 TOTEUTUS

Toteutus aloitettiin projektisuunnitelman mukaan, mutta kuitenkin heti alkumetreillä huomattiin, että laaditussa suunnitelmassa ei voida pysyä. Näin ollen suunnitelmasta jouduttiin poikkeamaan erinäisistä aikataulullisista seikoista johtuen. Suunnitelma laajeni ja jouduttiin ottamaan projektiin mukaan myös erilaisten lasku- ja reskontra-aineistojen luonti sekä GDPR tietosuoja uudistuksen vaatimia muutoksia.

3.1 Jäsenrekisteri

Kun lähdettiin yhdistämään Jäsenrekisteriä Kongressiin, täytyi ensimmäisenä miettiä, kuinka järkevimmin saadaan yhdistettyä molempien ominaisuudet. Kongressi-tapahtumahallintajärjestelmän ollessa jo pitkälle kehittynyt ja monipuolinen tultiin siihen tulokseen, että uusi jäsenrekisteri tulisi olemaan tapahtuma, jolle voidaan asettaa tyyppiä ”jäsenrekisteri”. Tämä salli myös Kongressissa olevien ominaisuuksien kuten asiakastietojen ylläpidon käytön ilman täysin uudelleen koodaamista. Tapahtumahallintaan lisättiin uusi tyyppi tapahtumalle nimeltä ”jäsenrekisteri”. Tämä valinta tulisi jatkossa muokkaamaan tapahtuman toimintaa muista tapahtumista.

KONGRESSI Jäsenrekisteri 28.2.2018 - 28.2.2018

Tapahtuma - Rekisterit - Contio - Asetukset

Peruuta Kumoa Tallenna

Perustiedot

Tapahtuman nimi: * Jäsenrekisteri

Ajankohta: * 28.02.2018 00:00 - 28.02.2018 23:59

Tunniste: * T_1895

Ulkoinen tunniste:

Kesto (päivää):

Max. henkilö määrä:

Tapahtumapaikka:

Osoite:

Postinumero:

Postitoimipaikka:

Tapahtuman tila: Päättynyt

Näytä lisäasetukset

Tapahtuman tyyppi: Jäsenrekisteri

Tapahtuman laji: Valitse

Maa: Jäsenrekisteri

WWW-osoite: Kokous

Email: Koulutusilaisuus

Koulutussuunnittelija: Organisaation sisäinen koulutus

Koulutussihteeri: Seminaari

Vastuhenkilö: Vapaa-aika

Tapahtuma on ulkoinen koulutus Verkkokoulutus

+Lisää uusi

Tavoitteet:

Kohderyhmä:

Keskeinen sisältö:

Annointimenetelmät:

Toteuttamiseen osallistuvat henkilöt:

Yhteyshenkilö/ Lisätietoja antaa:

Järjestäjän tiedot

Laskuttajan tiedot

Kuva 2 Tapahtuman ylläpidossa voi valita tyyppiä jäsenrekisteri

Ilmoittautuessa jäsenrekisteriin, jäsenelle annetaan automaattisesti jäsennumero, jota voidaan käyttää jäsenten identifiointiin toisistaan. Asiakastietojen ylläpitoon lisättiin Työosoitteet-lohko, jossa voidaan ylläpitää asiakkaan työosoitteita. Lisää -painiketta klikkaamalla avautuu modaali-ikkuna, jossa kysytään asiakkaan työosoitetietoja. Tallentamalla tiedot asiakkaalle tallentuu uusi työosoite, jota voidaan muokata painamalla listalla olevasta osoitteesta. Tällöin modaali-ikkuna avautuu ja kentät ovat valmiiksi täytettyinä valitulla osoitetiedolla. Muokkaamalla näitä tietoja ja painamalla tallenna, tiedot tallentuvat järjestelmään. Myös osoitetietojen poistaminen on mahdollista, valitsemalla valintaruudun osoitteen vierestä ja painamalla Poista-painiketta.

<input type="checkbox"/>	Yritys	Osoite	Postinumero	Postitoimipaikka	Oletus
<input type="checkbox"/>	Data Prisma	Kauppakatu 20	70100	Kuopio	<input checked="" type="radio"/>

Kuva 3 Työosoitteiden ylläpito

Olet lisäämässä työosoitetta henkilölle: **Matti Mielikäinen**

Nimi: *
Data Prisma

Sähköposti: *

Hetu/Y-Tunnus: *

Oletustyöosoite

Osoite: *
Kauppakatu 20

Postinumero: *
70100

Postitoimipaikka: *
Kuopio

Maa: *
Suomi

Peruda Tallenna

Kuva 4 Työosoitteen lisäys/muokkaus modaali-ikkuna

3.2 Laskutusaineisto

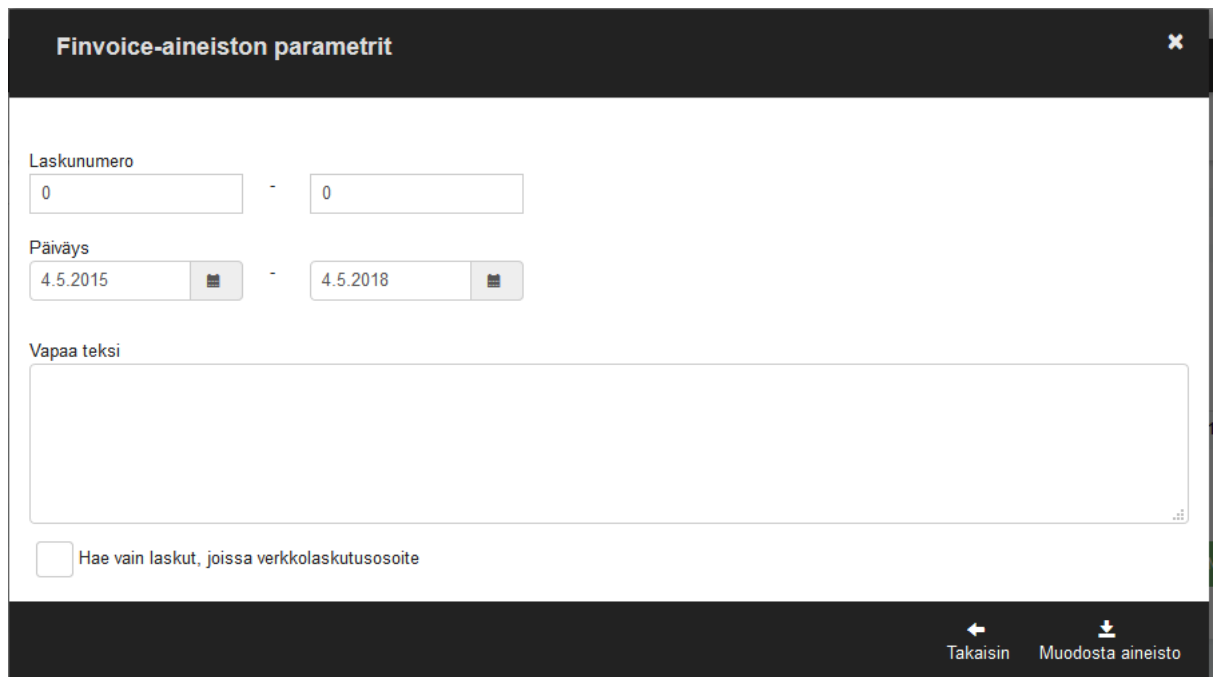
Suunnitelmasta poiketen projektiin otettiin mukaan myös laskutusaineiston generointi, joka aluksi toteutettiin vain Finvoice-aineistolle. Finvoice on suomalaisten pankkien määrittelemä ja Suomessa yleisimmin käytetty verkkolaskukuvauksen muoto. (Finanssiala ry, 27.4.2018) Tämä ominaisuus tarvittiin Kongressiin, jotta Kongressin käyttäjät voivat laskuttaa asiakkaitaan. Laskutusaineiston generointia lähdettiin suunnittelemaan siten, että sitä olisi mahdollista hyödyntää myös myöhemmin tuleville laskuaineistoille. Näin ollen oli järkevää tehdä toteutus, jota pystyy käyttämään myös muille aineistoille esimerkiksi Lindorffille ja vain itse aineiston luonti erottaa omaksi osakseen.

Laskuaineistojen datan haku toteutettiin yhteiseksi, koska se ei tule eroamaan aineistojen välillä millään tavalla. Datan haussa käytettiin Code First -tekniikkaa ja luotiin omia geneerisiä käsittelijöitä, joilla haettiin tarvittavat datat kannassa. Geneeriset käsittelijät toimivat samaan tapaan kuin Entity Framework repository, mutta niillä pystyttiin luomaan omanlainen entiteetti, johon yhdistettiin tietokannasta data itseluomiin objekteihin contextin luonnin yhteydessä.

Hyötynä tässä tekniikassa oli se, että pystyi itse luomaan sen tyyppisen objektin kuin tarvitsi ja mappamaan sen tietokantaan. Tällöin hakuja tehdessä ei tarvinnut palauttaa taulun kaikkia kenttiä vaan ne joita tarvitsi. Huonoksi puoleksi havaittiin se, että mikäli kyselystä tuli kauhean monimutkainen ja se sisälsi monta eri taulurakennetta, niin käytännössä täytyi tehdä monta erillistä hakuja ja yhdistellä hakutulokset haluttuun lopputulokseen muistinvaraisesti. Mikäli monimutkaisempia hakuja olisi tarvinnut tehdä olisi se tietysti onnistunut myös suorilla SQL-lauseilla.

Laskuaineistoihin ei kuitenkaan tarvittu kovin monimutkaisia hakuja, joten tämä osoittautui hyvin vaihtoehtoiseksi vaihtoehdoksi sen takia, ettei ylimääräistä dataa tarvitse kuljettaa prosessin mukana hidastamassa. Laskumassat voivat olla niin suuria niin että on turhaa hakea ylimääräistä dataa. Kun kaikki tarvittava data on haettu niin ja suodatettu parametrien mukaan, niin se mapataan yhteiseen laskuainesto-objektiin, josta voidaan muodostaa muun muassa Finvoice-aineisto.

Kongressin Talous -näkyvässä valitaan tapahtuman valintalaatikoita klikkaamalla ja painetaan Muodosta Finvoice-painiketta. Painikkeesta avautuu modaali-ikkuna, jossa käyttäjä voi syöttää haluamansa aineiston parametrit esimerkiksi millä laskunumero välillä haluaa aineiston olevan ja minkä tyyppisen aineiston hän haluaa tulostaa, tässä tapauksessa Finvoice. Kun parametrit on valittu ja painetaan Muodosta aineisto -painiketta, niin avautuu yhteenvetosivu, jossa näytetään yhteenveto tulostettavasta aineistosta.



Finvoice-aineiston parametrit

Laskunumero
0 - 0

Päiväys
4.5.2015 - 4.5.2018

Vapaa teksti

Hae vain laskut, joissa verkkolaskutusosoite

Takaisin Muodosta aineisto

Kuva 5 Laskuaineiston tulostuksen parametrinäyttö

Tällä näkymällä käyttäjä voi vielä tarkastaa aineiston ja esimerkiksi muuttaa henkilön verkkolaskuosoitetta tai operaattoria, mikäli havaitsee niissä virheitä. Käyttäjän ollessa valmis tulostamaan aineisto painetaan Tallenna -painiketta, jolloin laskuaineiston muodostus päätellään parametrina valitun aineistomuodon perusteella, joka tässä tapauksessa on Finvoice.

Yhteenveto							
Laskunumero	Asiakasnumero	Nimi	Verkkolaskuosoite	Operaattori	Laskun päiväys	Eräpäivä	Summa
7	8798	Henni Haakola		print	16.8.2017 0:00:00	30.8.2017 0:00:00	8,95
781	8798	Henni Haakola		print	23.8.2017 0:00:00	6.9.2017 0:00:00	50
11	8874	Majjalisa Mättänen		print	16.8.2017 0:00:00	30.8.2017 0:00:00	50
13	8874	Majjalisa Mättänen		print	16.8.2017 0:00:00	30.8.2017 0:00:00	4,95
782	8874	Majjalisa Mättänen		print	23.8.2017 0:00:00	6.9.2017 0:00:00	4,95
8	9510	Liisa Jaakkola		print	16.8.2017 0:00:00	30.8.2017 0:00:00	38,71
2587	9510	Liisa Jaakkola		print	15.9.2017 0:00:00	29.9.2017 0:00:00	50
676	9703	Jussi Tapaninen		print	22.8.2017 0:00:00	5.9.2017 0:00:00	8,95
17	9807	marjainen marjaana		print	16.8.2017 0:00:00	30.8.2017 0:00:00	20
783	9807	marjainen marjaana		print	23.8.2017 0:00:00	6.9.2017 0:00:00	30
9	12404	Kajja Karmaiseva		print	16.8.2017 0:00:00	30.8.2017 0:00:00	11
3063	12482	Tanja Testaaja		DABAFIHH	11.12.2017 0:00:00	11.12.2017 0:00:00	50
10	12628	Maisa Maksaja		print	16.8.2017 0:00:00	30.8.2017 0:00:00	40
2588	12628	Maisa Maksaja		print	15.9.2017 0:00:00	29.9.2017 0:00:00	68
11	12630	Niina Näpsäkkä		print	16.8.2017 0:00:00	30.8.2017 0:00:00	25,69
784	12630	Niina Näpsäkkä		print	23.8.2017 0:00:00	6.9.2017 0:00:00	186
1068	12630	Niina Näpsäkkä		print	25.8.2017 0:00:00	8.9.2017 0:00:00	68
1070	12630	Niina Näpsäkkä		print	25.8.2017 0:00:00	8.9.2017 0:00:00	30,65

Kuva 6 Tulostettavan laskuaineiston yhteenveto

Laskuaineiston muodostuksen käsittelijässä ohjataan aineisto oikeaan käsittelijään, jossa itse aineisto luodaan. Näin ollen vain aineiston luonnin toteutus eroaa muista, joten koodin ylläpito ja mahdollisten virheiden paikantaminen ja korjaaminen ovat huomattavasti helpompaa. Kun aineisto on muodostettu, se palautetaan käyttöliittymään ladattavaksi, jolloin käyttäjä voi ladata sen omalle koneelleen ja lähettää sen eteenpäin pankkiin.

```

<InvoiceRow>
  <ArticleName>Osallistuminen ma 100e</ArticleName>
  <DeliveredQuantity QuantityUnitCode="kpl">1</DeliveredQuantity>
  <StartDate Format="CCYYMMDD">20171229</StartDate>
  <EndDate Format="CCYYMMDD">20171230</EndDate>
  <UnitPriceAmount AmountCurrencyIdentifier="EUR">96,15</UnitPriceAmount>
  <RowIdentifier> </RowIdentifier>
  <RowOrdererName> </RowOrdererName>
  <RowVatRatePercent>4,00</RowVatRatePercent>
  <RowVatAmount AmountCurrencyIdentifier="EUR">3,85</RowVatAmount>
  <RowVatExcludedAmount AmountCurrencyIdentifier="EUR">96,15</RowVatExcludedAmount>
  <RowAmount AmountCurrencyIdentifier="EUR">100,00</RowAmount>
</InvoiceRow>
<InvoiceRow>
  <ArticleName>Osallistuminen ti 111e</ArticleName>
  <DeliveredQuantity QuantityUnitCode="kpl">1</DeliveredQuantity>
  <StartDate Format="CCYYMMDD">20170622</StartDate>
  <EndDate Format="CCYYMMDD">20170623</EndDate>
  <UnitPriceAmount AmountCurrencyIdentifier="EUR">99,11</UnitPriceAmount>
  <RowIdentifier> </RowIdentifier>
  <RowOrdererName> </RowOrdererName>
  <RowVatRatePercent>12,00</RowVatRatePercent>
  <RowVatAmount AmountCurrencyIdentifier="EUR">11,89</RowVatAmount>
  <RowVatExcludedAmount AmountCurrencyIdentifier="EUR">99,11</RowVatExcludedAmount>
  <RowAmount AmountCurrencyIdentifier="EUR">111,00</RowAmount>
</InvoiceRow>
<EpiDetails>
  <EpiIdentificationDetails>
    <EpiDate Format="CCYYMMDD">20171229</EpiDate>
    <EpiReference>01764000842900031559</EpiReference>
  </EpiIdentificationDetails>
  <EpiPartyDetails>
    <EpiBfiPartyDetails>
      <EpiBfiIdentifier IdentificationSchemeName="BIC">NDEAFIHH</EpiBfiIdentifier>
      <EpiBfiName>Nordea</EpiBfiName>
    </EpiBfiPartyDetails>
    <EpiBeneficiaryPartyDetails>
      <EpiNameAddressDetails>Data Prisma testaa</EpiNameAddressDetails>
      <EpiAccountID IdentificationSchemeName="IBAN">FI9115893000050196</EpiAccountID>
    </EpiBeneficiaryPartyDetails>
  </EpiPartyDetails>
  <EpiPaymentInstructionDetails>
    <EpiPaymentInstructionId>Tapahtuma</EpiPaymentInstructionId>
    <EpiRemittanceInfoIdentifier IdentificationSchemeName="SPY">01764000842900031559</EpiRemittanceInfoIdentifier>
    <EpiInstructedAmount AmountCurrencyIdentifier="EUR">211,00</EpiInstructedAmount>
    <EpiCharge ChargeOption="SHA">SHA</EpiCharge>
    <EpiDateOptionDate Format="CCYYMMDD">20180112</EpiDateOptionDate>
  </EpiPaymentInstructionDetails>
</EpiDetails>

```

Kuva 7 Esimerkki Finvoice-aineiston laskuriveistä, joiden mukaan pankissa muodostetaan laskut

3.3 Reskontra-aineisto

Toisena suurena lisäyksenä projektiin tuli erilaisten reskontra-aineistojen tulostus. Tätä lähdettiin suunnittelemaan samalta kannalta kuin laskuaineistoakin eli tavoitteena oli saada mahdollisimman yleiskäyttöinen toteutus, jotta myöhemmin on mahdollista lisätä myös muidenkin aineistojen tulostusominaisuus Kongressiin mahdollisimman helposti.

3.3.1 AX-aineisto

Microsoft Dynamics AX on toiminnanohjausjärjestelmä. Toiminnanohjausjärjestelmällä yritys voi tehostaa toimintaansa liiketoimintansa eri osa-alueilla. Microsoft Dynamics toiminnanohjausjärjestelmä on paljon käytetty järjestelmä erilaisten yritysten keskuudessa, joten oli luonnollista, että myös Kongressiin tuli tuki kyseiselle alustalle helpottamaan asiakkaan kirjanpitoa. (Microsoft, 2018)

Asiakkaan toiveiden mukaan lähdettiin suunnittelemaan AX-integraatiota. Suurimmat tarpeet asiakkaalla oli, että aineisto tulisi saada joko tulostaa itse Kongressi-järjestelmästä tai se tulisi automatisoida. Aluksi lähdettiin liikkeelle siitä, että mahdollistetaan aineiston tulostus manuaalisesti. Tämä oli helppo päätös, sillä meillä oli jo aiemmin mahdollistettu muun muassa Fivaldi, Raindance ja Rondo aineistojen tulostus. Siispä oli loogista lähteä lisäämään uusi aineistomuoto mukaan manuaaliseen tulostukseen, sillä automatiikkaa ei vielä ollut.

AX-aineiston tulostusta oli sinänsä helppo lähteä luomaan, koska aineisto pohjimmiltaan on selkeä xml-rakenteinen tiedosto. Visual Studio 2017 käyttäessä myös apuna on paljon hyviä työkaluja, esimerkiksi "Paste XML as classes", joka luo kopioidusta xml:stä luokat valmiiksi. Tämä säästää paljon käsityötä ja aikaa, kun jokaista elementtiä ei tarvitse itse luoda.

Reskontra-aineistoon tulevan datan haku tehtiin yhteiseksi kaiken tyyppisille reskontra-aineistoille, niin sitä voitiin myös käyttää AX-aineiston kanssa. Koska kaikki aineistot tarvitsevat samaa dataa, mutta parametrit, joilla dataa rajataan voi muuttua. Siispä jäljellä oli enää datan mappaus AX:n xml muotoon, joka oli helppo toteuttaa Visual Studion luomiin luokkiin.

3.3.2 Reskonta-aineiston tulostuksen automatisointi

Asiakkaan toinen vaatimus oli, että aineiston tulostus tulisi pystyä automatisoimaan. Eli tässä tapauksessa aineiston pitäisi tulostua automaattisesti kerran päivässä ja sen jälkeen siirtyä asiakkaan ympäristöön Sftp-siirtona. Kun automatisointia lähdettiin suunnittelemaan, niin tässä vaiheessa meillä oli jo siis aineiston luonti manuaalisesti valmiina, joten ei tarvittu enää kuin rajapinta, jota kutsumalla oikeilla parametreilla voidaan saada oikeanlainen aineisto. Sen jälkeen tämä aineisto täytyi saada tallennettua meidän palvelimelle, josta se siirrettäisiin Sftp-siirtona asiakkaalle ja lähetetty tiedosto sitten siirrettiisiin arkistoon myös meidän palvelimellamme.

Automatisointi tehtiin yksinkertaisella konsolisovelluksella, joka lukee halutut parametrit xml-tiedostosta ja kutsuu niillä parametreilla rajapintaa, jonka takana itse aineiston luonti on. Xml-tiedostoon laitetaan muun muassa tieto asiakkaasta, jonka järjestelmästä aineisto halutaan luoda, mikä aineisto luodaan ja esimerkiksi miltä aikaväliltä aineisto luodaan.

Kun aineisto on luotu ja tallennettu palvelimelle, niin myöhemmin toinen sovellus käynnistyy, joka suorittaa itse Sftp-siirron ja lähetetyn tiedoston siirtämisen arkistoon. Haluttiin kaksi erillistä ajoa sen takia että varmistetaan, että kaikki aineistot on varmasti ehditty luoda ennen kuin lähdetään yrittämään aineistojen lähetystä.

Konsolisovellusten ajo ajastettiin palvelimella Windowsin Task Schedulerilla. Tämä oli erittäin yksinkertainen tapa tehdä ajastettuja toimintoja ja samanlainen tekniikka meillä oli jo käytössä aiemminkin.

```
static void SftpUpload()
{
    string settingsPath = ConfigurationManager.AppSettings["settings_filepath"];
    string sSource = "Kongressi.LedgerSftpScheduler";
    string sLog = "Application";
    string sEvent = "LedgerSftpScheduler";

    if (!string.IsNullOrEmpty(settingsPath))
    {
        System.Xml.XmlDocument xml = new System.Xml.XmlDocument();
        xml.Load(settingsPath);
        System.Xml.XmlNodeList nodeList = xml.SelectNodes("/LedgerSftpSettings/Item");
        var endPointList = new List<string>();

        foreach (System.Xml.XmlNode xn in nodeList)
        {
            endPointList.Add(xn.ChildNodes[0].InnerText);
        }

        foreach (var endpointAddress in endPointList)
        {
            using (var client = new KongressBL.InvoicingHandler.InvoicingHandlerClient("WSHttpBinding_IInvoicingHandler", endpointAddress))
            {
                try
                {
                    client.Open();
                    client.SftpUploadLedgerRecords();
                }
                catch (Exception ex)
                {
                    if (!EventLog.SourceExists(sSource))
                        EventLog.CreateEventSource(sSource, sLog);

                    EventLog.WriteEntry(sSource, sEvent + Environment.NewLine + "Virhe tapahtui ajastetussa reskontra-aineiston sftp-tiedostonsiirrossa: " + Environment.NewLine + ex.ToString(),
                        EventLogEntryType.Error, 1010);
                }
                finally
                {
                    if (client.State == System.ServiceModel.CommunicationState.Opened)
                    {
                        client.Close();
                    }
                }
            }
        }
    }
}
```

Kuva 8 Sftp-siirrossa käytettävä yksinkertainen konsolisovellus

3.4 Pankkitunnistautuminen

Kongressin ilmoittautumislomakkeelle haluttiin ominaisuus, jolla jäsenrekisteriin ilmoittuvat voisivat tunnistautua. Meillä oli jo ilmoittautumislomakkeella mahdollisuus kirjautua järjestelmään, jolloin jo olemassa oleva Kongressin käyttäjä voisi kirjautua sisään ja hänen tietonsa esitetytään lomakkeelle. Tähän tehtiin lisäys, joka mahdollisti myös pankkitunnistautumisen. Onnistuneen tunnistautumisen jälkeen esitetytään ilmoittautujan etu- ja sukunimi sekä henkilötunnus, mikäli ne ovat kysyttäviä tietoja lomakkeella. Samalla myös estetään käyttäjää muokkaamasta näitä tietoja tunnistautumisen jälkeen.

Kirjaudu sisään

Kirjaudu Kongressi-tunnuksilla

Käyttäjätunnus

Salasana

KIRJAUDU

[Unohtuiko salasana?](#)

Sähköinen tunnistautuminen

Nordea, Aktia, Handelsbanken, S-Pankki, Banki.fi, omaOP, Mobilivarmenne

Asiakastyypit: *
Valitse

Yksikkö:
Valitse

Jäsentunnus:

Hetu/Y-Tunnus:

Etunimi: *

Sukunimi: *

Postitoimipaikka: *

Titteli:

Maa:
Valitse

Osoite: *

Esimies:
Valitse

Tietosuojaa-asetukset ja muut kysymykset. Ole hyvä ja vastaa alla oleviin tietosuojakysymyksiin. Kysymykset ovat pakollisia.

Kuva 9 Ilmoittautumislomakkeella käyttäjä voi tunnistautua valitsemallaan sähköisellä tunnistautumalla

Pankkitunnistautuminen tehtiin Finanssialan keskusliiton standardin mukaan yksinkertaisella lomake-rakenteella. Itse tunnistautumisen rakentaminen oli suoraviivaista, koska jokainen tarvittava tunnistautumispalvelu toimii tällä samalla periaattella, joten jokaiselle tunnistautumispalvelulle ei tarvinnut luoda erillistä toteutusta. Tämän lisäksi ilmoittautumislomakkeen ylläpitoon haluttiin asetus, jolla voidaan määritellä se että onko kyseisellä lomakkeella käytössä pankkitunnistautuminen vai ei. Myös yleisiin asetuksiin tuli ylläpito kaikille tunnistautumistavoille. Yleisissä asetuksissa Kongressin käyttäjä voi ylläpitää muun muassa tunnistautumispalvelun tunnuksia.

```
<FORM METHOD="POST" ACTION="pankin Tupas-palvelun URL">
<INPUT NAME="A01Y_ACTION_ID" TYPE="hidden" VALUE="701">
<INPUT NAME="A01Y_VERS" TYPE="hidden" VALUE="...">
<INPUT NAME="A01Y_RCVID" TYPE="hidden" VALUE="...">
<INPUT NAME="A01Y_LANGCODE" TYPE="hidden" VALUE="...">
<INPUT NAME="A01Y_STAMP" TYPE="hidden" VALUE="...">
<INPUT NAME="A01Y_IDTYPE" TYPE="hidden" VALUE="...">
<INPUT NAME="A01Y_RETLINK" TYPE="hidden" VALUE="...">
<INPUT NAME="A01Y_CANLINK" TYPE="hidden" VALUE="...">
<INPUT NAME="A01Y_REJLINK" TYPE="hidden" VALUE="...">
<INPUT NAME="A01Y_KEYVERS" TYPE="hidden" VALUE="...">
<INPUT NAME="A01Y_ALG" TYPE="hidden" VALUE="...">
<INPUT NAME="A01Y_MAC" TYPE="hidden" VALUE="...">
</FORM>
```

Kuva 10 Sähköisen tunnistautumisen lomake (finanssiala.fi)

3.5 GDPR tietosuoja muutoksia

Euroopan unionin tietosuojalain uudistuksien myötä tuli tarpeelliseksi edistää Kongressissa GDPR (General Data Protection Regulation) tietosuoja-asetuksen sisältämiä ohjeistuksia, jotka liittyvät henkilötietojen käsittelyyn. (Tietosuoja.fi, 23.3.2018) Koska tietosuoja-asetusta aletaan soveltamaan kaikissa EU-jäsenmaissa, niin nämä muutokset tulivat tarpeelliseksi hyvinkin nopeasti. Alustavasti haluttiin Kongressiin käyttöön erilaisten tietosuoja-asetusten ylläpito, jossa Kongressin käyttäjä voi luoda ja ylläpitää esimerkiksi erilaisia kieltoja ja lupia sekä asettaa näille asetuksille oletusarvon, ovatko ne käytössä vai eivät. Käyttäjä voi myös asettaa nämä luodut asetukset kysyttäväksi ilmoittautumisloMAKEella, jolloin ilmoittautuvan asiakkaan on pakko vastata tietosuojakyselyyn.

Kun luodaan uusi tietosuoja Kongressiin, niin se lisätään aina kaikille järjestelmän asiakkaille. Mikäli olemassa olevan tietosuojan aktiivisuutta muutetaan, niin se tallennetaan myös kaikille asiakkaille. Tällä varmistetaan se, ettei järjestelmästä löydy yhtään asiakasta, jolla ei olisi kaikkia tietosuoja-asetuksia.

The screenshot displays the 'Tietosuojien ylläpito' (Data Protection Settings) modal window. The window is titled 'Tietosuojien ylläpito' and contains the following fields and options:

- Suomi:** Tietojenluovutuskielto
- Englanti:** Tietojenluovutuskielto EN
- Ruotsi:** Tietojenluovutuskielto SWE
- Tyyppi:** Kielto
- Aktiivinen
- Oletus:** Ei Kyllä Ei tiedossa

At the bottom of the modal, there are buttons for 'Peruuta' (Cancel) and 'Tallenna' (Save). Below the modal, the background interface shows a list of 'Tietosuoja-asetukset' (Data Protection Settings) for 'Suomi' with the following options:

- Kaiken kielto
- Tietojenluovutuskielto
- Käsittelyn esto

Below the list, there are two columns of settings for 'Tietojenluovutuskielto EN' and 'Tietojenluovutuskielto SWE', both with 'Processing ban' as the default value.

Kuva 11 Tietosuoja-asetusten ylläpito yleisistä asetuksista

Ilmoittautumislomakkeella ilmoittautunut asiakas saa automaattisesti kaikki Kongressin järjestelmässä olevat tietosuoja-asetukset niiden oletusarvoilla, jotka Kongressin käyttäjä on tietosuojuille asettanut niitä luodessaan. Mikäli tietosuoja-asetuksia on kysytty lomakkeella, niin käyttäjä saa kysytyt tietosuoja-asetukset niillä arvoilla kuin hän on niihin vastannut. Ilmoittautunut asiakas pystyy myöhemmin muuttamaan tietosuoja-asetuksiaan kirjautumalla Portaaliin. Kongressin pääkäyttäjä pystyy myös muokkaamaan asiakkaiden tietosuoja-asetuksia, mikäli tälle on tarvetta.

Roolit: *
Esiintyjä

Etunimi: *
Mikko

Sukunimi: *
Testaa

Osoite: *

Puhelin: (Syötä muodossa +358401234567)

Sähköposti: *

Organisaatio:

Esimies:
Valitse

+ LISÄÄ TIEDOSTOJA...

Tiedostojen yhteenlaskettu maksimikoko on 10 MB

Tietosuojat FI
Käsittelyn esto

Ei
 Kyllä

Kaiken kieltö

Ei
 Kyllä

Osasto:

Postinumero: *

Lisätiedot: *

Postitoimipaikka: *

Maa:
Valitse

Kuva 12 Tietosuojien kysyminen ilmoittautumislomakkeella

4 YHTEENVETO

Tuloksena saatiin asiakkaille tärkeimmät ominaisuudet valmiiksi ja päivitettyä tuotantoon. Alustavat toimenpiteet tulevien Euroopan unionin tietosuojauudistusten hoitamiseksi saatiin valmiiksi ja nyt Kongressin käyttäjillä on mahdollista luoda sekä ylläpitää tietosuoja-asetuksia. Laskujen tulostaminen Finvoice verkkolaskuformaattiin sekä reskontra-aineistojen tulostaminen moneen erilaiseen formaattiin, muun muassa Fivaldi-aineistoon, onnistui hyvin ja validit aineistot voidaan lukea sisään ulkoisiin järjestelmiin sekä lähettää eteenpäin pankkeihin laskutusta varten.

Mielestäni lopputulos on onnistunut ja kaikki ominaisuudet on rakennettu siten, että niiden ylläpitäminen on helppoa ja mahdolliset lisäominaisuudet on helppo liittää nykyiseen toteutukseen. Kongressiin liitetyt uudet ominaisuudet sopivat saumattomasti uuteen järjestelmään ja ne tuovat huomattavaa lisäarvoa Kongressi-tuotteelle ja näin ollen mahdollistaen lisämyynnin sekä nykyisten asiakkaiden sitomisen tuotteeseen.

Työn tarkoituksena oli vanhan tuotteen parantaminen, migraatio uuteen sekä vanhan tekniikan päivittäminen uudempaan niin mieleeni väistämättä tuli myös Kongressin parantaminen. Millä tavoin Kongressia voisi parantaa? Kongressi sisältää joitain vanhentuneita ominaisuuksia, joita olisi hyvä tulevaisuudessa refaktoroida sujuvammaksi sekä helpommaksi ylläpitää. Tästä hyvänä esimerkkinä toimii esimerkiksi Portaali, jossa käyttäjät voivat käydä muokkaamassa omia tietojaan. Portaali on ollut kokonaan oma projektinsa, mutta se on otettu käyttöön myös Kongressissa ja siihen on tullut paljon lisäominaisuuksia joita ei ole aiemmin ollut. Tämä on johtanut siihen, että Portaali on suhteellisen hidas käyttää ja uusien ominaisuuksien tekeminen Portaaliin on melko hankalaa. Tästä syystä se tulisi refaktoroida paremmin osaksi Kongressia.

LÄHTEET JA TUOTETUT AINEISTOT

Microsoft 2018. ASP.Net MVC overview. [Viitattu 2018-04-29.] Saatavissa:

[https://msdn.microsoft.com/en-us/library/dd381412\(v=vs.108\).aspx](https://msdn.microsoft.com/en-us/library/dd381412(v=vs.108).aspx)

Microsoft 2018. Entity Framework documentation. [Viitattu 2018-05-16.] Saatavissa:

<https://docs.microsoft.com/en-us/ef/>

Finassiala ry 2018. Finvoice-verkkolasku yrityksille. [Viitattu 2018-04-29.] Saatavissa:

<http://www.finanssiala.fi/finvoice/Sivut/default.aspx>

Microsoft 2018. Microsoft Dynamics AX. [Viitattu 2018-05-16.] Saatavissa:

<https://www.microsoft.com/fi-fi/dynamics365/ax-overview>

Finanssiala ry 2013. Pankkien TUPAS-tunnistuspalvelu palveluntarjoajille. [Viitattu 2018-04-29.] Saatavissa: http://www.finanssiala.fi/maksujenvalitys/dokumentit/Tupas_varmennepalvelu_V_2.4.pdf

Tietosuoja.fi 2018. EU:n tietosuojauudistus. [Viitattu 2018-04-29.]. Saatavissa:

<http://www.tietosuoja.fi/fi/index/euntietosuojauudistus.html>