



SAVONIA

■ OPINNÄYTETYÖ - AMMATTIKORKEAKOULUTUTKINTO
TEKNIIKAN JA LIIKENTEEN ALA

INFO- JA RAPORTTIPORTAALI

TEKIJÄ: Sauli Ryyänen

Koulutusala Tekniikan ja liikenteen ala	
Koulutusohjelma Tietotekniikan koulutusohjelma	
Työn tekijä(t) Sauli Ryyänen	
Työn nimi Info – ja raporttiportaali	
Päiväys	11.2.2015
Sivumäärä/Liitteet	26/0
Ohjaaja(t) lehtori Sami Lahti, lehtori Keijo Kuosmanen	
Toimeksiantaja/Yhteistyökumppani(t) Enfo Oyj	
<p>Tiivistelmä</p> <p>Tämän opinnäytetyön aiheena oli suunnitella ja toteuttaa Enfon Service Desk -yksikölle info- ja raporttiportaali, jolla pääsee kätevästi käsiksi ajankohtaisiin info- ja raporttitietoihin. Portaali tuli toteuttaa web-pohjaisena, jotta sovellusta voidaan käyttää verkkoselaimella ja siitä voidaan toteuttaa myös seinänäyttönäkymä. Portaalin pitää pystyä esittämään ajankohtainen tieto selvästi sekä taltioimaan historiatietoja, joita voi jälkikäteen tarkastella järjestelmästä.</p> <p>Järjestelmä toteutettiin käyttäen ASP.NET MVC –sovelluskehystä ja toteuttamiseen käytettiin C#, HTML ja JavaScript -kieliä. Järjestelmän tietokanta toteutettiin MVC-mallien mukaisesti SQL-kantana. Järjestelmässä käytettiin Google Charts –työkalua historiatietojen graafeja varten. Järjestelmän kehittämiseen käytettiin Microsoftin Visual Studiota sekä SQL-tietokantojen käsittelyyn Microsoft SQL Management Studio -työkalua.</p> <p>Järjestelmä toimii Microsoft Windows Server 2008R2 –palvelimella, jonne on asennettu järjestelmän käyttöön IIS-websivupalvelin ja Microsoft SQL Server –tietokantapalvelin.</p> <p>Työn tavoite saavutettiin: Toteutettu järjestelmä sopii hyvin ajankohtaisen tiedon ja tilastojen välittämiseen Service Deskissä</p>	
Avainsanat ASP.NET MVC, .NET, info, raportti, Web, C#	

Field of Study Technology, Communication and Transport			
Degree Programme Degree Programme in Information Technology			
Author(s) Sauli Ryyänen			
Title of Thesis Info and Report Portal			
Date	11 February 2015	Pages/Appendices	26/0
Supervisor(s) Mr Sami Lahti, Lecturer, Mr Keijo Kuosmanen, Lecturer			
Client Organisation /Partners Enfo Oyj			
<p>Abstract</p> <p>The purpose of this thesis was to design and create an info and report portal to Enfo's Service Desk unit, whereby it is possible to seamlessly reach to current information and report data. The portal was implemented as web-based, so that the application can be used in a web browser, and it can also be implemented as a wall display view. The portal must be able to present current information clearly and to store history data, which can afterwards be looked at in the system.</p> <p>The system was implemented using the ASP.NET MVC application framework and the implementation was done using the C#, HTML, and JavaScript languages. The database was implemented in compliance with the MVC model in the SQL database. The Google Charts tool was used in the graphs for history data. The system was developed using Microsoft Visual Studio. The Microsoft SQL Management Studio tool was used for handling the SQL databases.</p> <p>The system runs on the Microsoft Windows Server 2008R2 server, in which the IIS webserver and Microsoft SQL database server were installed.</p> <p>The goal was achieved: The implemented system is very suitable for sharing up to date information and statistics at the Service Desk.</p>			
Keywords ASP.NET MVC, .NET, info, report, Web, C#			

SISÄLTÖ

LYHENTEET	6
1 JOHDANTO	7
2 ENFO OYJ	8
3 KÄYTETYT TEKNIIKAT JA TYÖKALUT	9
3.1 ASP.NET MVC 4	9
3.2 C#	10
3.3 HTML	10
3.4 CSS	11
3.5 Microsoft Visual Studio	11
3.6 Microsoft Windows Server	11
3.7 Microsoft SQL Server	11
3.8 MS SQL Management Studio	11
3.9 jQuery	12
3.10 Google Charts API	12
4 PORTAALIN SUUNNITTELU	13
5 PORTAALIN TOTEUTUS	14
6 TIEDON LÄHDEJÄRJESTELMÄT	16
6.1 MyEnfo	16
6.2 SAP BCM	16
7 INFO- JA RAPORTTIPORTAALI	17
7.1 Mallit	17
7.2 Näkymät	18
7.2.1 Päänäkymä	18
7.2.2 Tiimikohtainen näkymä	19
7.2.3 Asiakaskohtainen näkymä	20
7.2.4 Historiagraafit	21
7.2.5 Seinänäyttönäkymä	22
7.3 Kontrollerit	22
7.4 Datan hakeminen ja muodostaminen	22
8 TESTAUS	24
9 YHTEENVETO	25

10 LÄHTEET 26

LYHENTEET

API	API (Application programming interface) eli ohjelmointirajapinta, jonka avulla erilliset ohjelmat voivat tehdä pyyntöjä ja tietojen vaihtoa keskenään.
CSS	CSS (Cascading Style Sheets) on tyyliohjeiden laji. Yksittäisistä CSS tyyliohjeista voidaan muodostaa kokonainen tyyllisäännöstä. Käytetään erityisesti WWW-dokumenttien tyylinmäärittelyihin.
HTML	HTML (Hypertext Markup Language) eli hypertekstin merkintäkieli. Käytetään erityisesti WWW-sivujen kuvaamiseen.
ITSM	ITSM (IT service management) eli IT-palvelunhallinta.
JavaScript	JavaScript on WWW-sivuilla käytettävä komentosarjakieli.
jQuery	jQuery on JavaScriptille tehty kirjastolaajennus, joka on tehty yksinkertaistamaan JavaScriptin käyttöä.
MVC	MVC (Model-View-Controller) on ohjelmoinnissa käytettävä arkkitehtuurityyli, jota käytetään sovellusalueiden erottamisesta käyttöliittymästä.
SQL	SQL (Structured Query Language) on standardoitu kyselykieli, jota käytetään hakujen, muutosten ja lisäysten tekoon relaatiotietokantoihin.
WWW	WWW (World Wide Web) on internet-verkossa toimiva hajautettu hypertekstijärjestelmä.
XML	XML (Extensible Markup Language) on tietynlainen merkintäkielien standardi, jolla voidaan kuvata tietojen merkitys tiedon seassa.

1 JOHDANTO

Tämän opinnäytetyön aiheena on suunnitella ja toteuttaa info- ja raporttiportaali työn tilaajan Enfo Oyj:n Service Desk -yksikön tarpeisiin. Työn tavoitteena on toimiva, tarkoituksenmukainen ja luonteva informaatiolähde työntekijöiden päivittäiseen käyttöön. Tässä raportissa paneudutaan järjestelmässä käytettäviin tekniikoihin, suunnitteluun ja toteutukseen.

Työn aihe tuli Service Desk -yksikössä ilmenneestä tarpeesta parantaa ajantasaisen informaation ja raporttien hyödyntämistä päivittäisen toiminnan tukena. Tietoja raporteja varten saadaan haettua käytössä olevista ITMS-järjestelmä MyEnfosta sekä SAP BCM -puhelinjärjestelmästä. Järjestelmän avulla eri lähdejärjestelmistä saatuja tietoja voidaan käyttää tehokkaasti yhdessä. Järjestelmän tehtävänä on tukea raporttitiedon hyödyntämistä ajantasaisen toiminnan suunnittelussa ja ohjauksessa sekä jakaa ajantasaista tietoa nopeasti koko yksikölle. Ensimmäisenä ajatuksena oli kehittää infotaulujärjestelmä, mutta siinä käytettävän datan hyödyntämiseen ideoitiin myös raportointijärjestelmä. Raporttijärjestelmä pureutuu tarkemmin eri järjestelmistä haettuun tietoon ja heti alun suunnitteluvaiheessa siitä tuli tärkeämpi osa toteutuksessa kuin infotaulu.

2 ENFO OYJ

Enfo on Kuopiossa vuonna 1964 perustettu IT-palvelutalo. Yritys tarjoaa IT-ulkoistusta, IT-konsultointipalveluita ja talousprosessipalveluja Suomessa, Ruotsissa, Norjassa ja Tanskassa. Enfon Suomen -päätoimipisteet sijaitsevat Kuopiossa ja Espoossa. Pääkonttori on Kuopiossa. (Enfo 11.2.2015.)

Enfon vahvuutena on yli 50 vuoden kokemus erilaisista toimivien tietotekniikkaratkaisujen ja -konseptien kehittämisestä. Yrityksen kantava ajatus on, että asiakas voi keskittyä omaan ydinliiketoimintaansa. Yrityksen palveluksessa on lähes 800 henkilöä ja sen liikevaihto on liki 160 miljoonaa euroa. (Enfo 11.2.2015.)

Enfo Service Desk ja käyttäjätukipalvelut tarjoavat kattavia ja laadukkaita tukipalveluita asiakkaan ympäristöjen ja vaatimusten mukaisesti. Palvelut koostuvat mm. palvelupyyntöjen ja vikailmoitusten hallinnannasta sekä työasemien ja mobiililaitteiden käyttäjätukipalvelusta. Yhteydenottokeinoja ovat puhelin, sähköposti, itsepalveluportaalit sekä chat-palvelu. Käyttäjätukea annetaan myös etäyhteyden avulla sekä tarvittaessa lähitukena asiakkaan toimipisteellä. Enfo Service Deskille on myönnetty ISO9001-laatusertifikaatti. (Enfo 11.2.2015.)

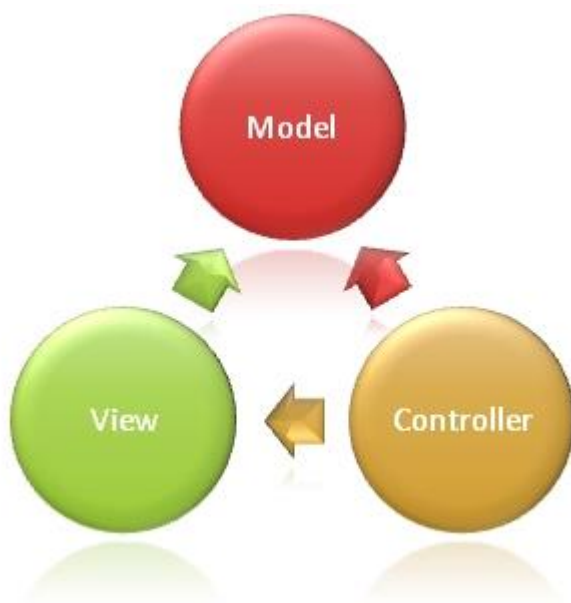
3 KÄYTETYT TEKNIIKAT JA TYÖKALUT

Järjestelmän toteuttamiseen käytettiin ASP.NET MVC4 –ohjelmistokehystä. Kehitystyökaluna käytettiin Microsoft Visual Studio -ohjelmistoa. Järjestelmän toteutukseen käytettiin pääasiassa C#:ia, HTML:ää ja jQuerya. Tiedonhakuohjelmissa käytettiin C#-kieltä ja tiedonhakuun SQL-hakulausekkeita sekä XML-kuvauskieltä.

Tietokantana käytettiin ASP.NET MVC4:sen kanssa luontevaa Microsoft SQL Server –tietokantaa. Järjestelmän palvelimena toimii virtuaalinen Windows Server 2008R2 –palvelin, jossa pyöriivät IIS –webpalvelin sekä Microsoft SQL Server -ohjelmisto.

3.1 ASP.NET MVC 4

ASP.NET MVC on MVC-arkkitehtuuria noudettava ASP.NET -pohjainen ohjelmistokehys, jota käytetään dynaamisten verkkosivustojen luomiseen. Ohjelmistokehysten kielinä voi käyttää C#- sekä VB.NET -kieliä. Sovellukset koostuvat kolmesta loogisesta osasta: Model, View sekä Controller. (ASP.NET MVC Overview 11.2.2015.) Model eli malli-osio määrittää käytettävän tiedon muodon, View eli näkymä-osio huolehtii siitä kuinka lopputulos näkyy käyttäjille ja Controller-osio ottaa vastaan komentoja ja parametreja, sekä määrittelee näytettävät näkymät ja niille syötettävän datan. Kuva 1 havainnollistaa MVC-mallin mukaista rakennetta. Malleille, näkymille ja kontrollereille on Visual Studion projektissa erilliset kansiot ja tiedostot, joiden avulla ohjelman eri osat pysyvät selkeästi erillään. Kehyksen etuina ovat mm. koodin selkeä hallinta ja rakenne sekä nopea ja tehokas tapa luoda uusia näkymiä.



KUVA 1. MVC rakenne

ASP.NET MVC:n versio 1.0 julkaistiin vuonna 2009, uusin versio on 5.2, joka on julkaistu vuonna 2014. Työ toteutettiin versiolla 4.0, joka oli uusin saatavilla oleva versio, kun projekti aloitettiin.

3.2 C#

C# on Microsoftin .NET-ohjelmointiympäristöä varten kehitetty nykyaikainen oliopohjainen ohjelmointikieli jossa on pyritty yhdistämään monen jo aikaisemmin olemassa olleen kielen parhaita puolia onnistuneesti. C# julkaistiin vuonna 2000.

Esimerkkinä C#:n syntaksista kuva 2 esittää "Hei maailma" – tekstin tulostamista konsoliohjelmassa.

```
using System;

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hei maailma!");
        }
    }
}
```

KUVA 2. C# esimerkki

3.3 HTML

HTML on hypertekstin kuvauskieli, jolla kuvataan mm. www-sivujen rakennetta ja jäsentelyä sekä sisällön muotoja, kuten tekstin korostamista. Yhdistelemällä HTML:n tunnisteita saadaan luotua HTML-sivu, jonka web-selaimet osaavat näyttää tulkitsemalla tiedostossa olevia tunnisteita. Tuorein versio HTML:sta on HTML 5. (w3schools.com 11.2.2015.) Projektissa toteuttavan järjestelmän näkymiä määritellään HTML:n avulla.

Esimerkki tekstin vahventamisesta HTML-kielellä:

```
<b>Text</b>
```

"b"-viittaa sanaan bold ja tarkoittaa tekstin vahvistamista. "<>"-merkeillä määritellään, koska sisään sijoitettu tunniste alkaa vaikuttaa ja "</>"-merkeillä, koska vaikutus päättyy.

3.4 CSS

CSS on HTML-sivujen tyylinmäärittelyyn tarkoitettu tekniikka. CSS-tietoja voidaan sijoittaa suoraan HTML-sivun koodiin tai sivulla viitattavaan erilliseen css-muotoiseen tiedostoon.

```
p {  
    font-family: "Arial";  
    font-size: 32px;  
    color: red;  
}
```

KUVA 3. CSS-esimerkki

Kuvan 3 esimerkki määrittelee "<p></p>"-merkkien väliin määritellyn tekstin fontin, koon ja värin.

3.5 Microsoft Visual Studio

Microsoft Visual Studio on Microsoftin kehittämä monipuolinen ja kattava kehitysympäristö. Visual Studiolla voidaan luoda mm. Windows-työasema, web- ja mobiilisovelluksia. Visual Studio tukee useita eri ohjelmointikieliä, kuten Visual Basic, C# ja C++. (Microsoft 11.2.2015.)

Visual Studio on virallinen työkalu ASP.NET MVC pohjaisten sovellusten suunnitteluun, toteuttamiseen ja testaamiseen ja soveltuu siksi järjestelmän toteuttamiseen parhaiten. Työssä käytettiin Visual Studion versiota 2013.

3.6 Microsoft Windows Server

Microsoft Windows Server on Microsoftin tuoteperhe, joka sisältää palvelinkäyttöön tarkoitettuja käyttöjärjestelmiä. Projektissa käytettiin Microsoft Server versiota 2008 R2.

3.7 Microsoft SQL Server

Microsoft SQL Server on kaupallinen ja monipuolinen tietokantapalvelin -ratkaisu, joka soveltuu pienistä sovelluksista aina mittavien sovellusten tietokantapalvelimeksi.

Projektissa käytettiin Microsoft SQL Server 2008 R2 -versiota.

3.8 MS SQL Management Studio

Microsoft SQL Server Management Studio on monipuolinen ja helppokäyttöinen hallinnointityökalu SQL-tietokantoja luomiseen, hallinnointiin ja muokkaamiseen. Työkalu soveltuu myös hyvin datan syöttämiseen ja hakemiseen tietokannoista sekä varmuuskopioiden tekemiseen tietokannoista.

3.9 jQuery

jQuery on monipuolinen JavaScript-kirjasto joka helpottaa JavaScript pohjaisten ratkaisujen tuottamista valmiiden ominaisuuksien ansiosta. (About jQuery 11.2.2015)

jQuery on ilmainen ja avointa lähdekoodia ja noussut todella suosituksi helpon käytettävyyden ja monipuolisuutensa ansiosta. jQuery on julkaistu vuonna 2006.

3.10 Google Charts API

Google Charts on Googlen Developers – tuoteperheeseen kuuluva kokonaisuus, joka tarjoaa monipuolisen työkalun jolla voi helposti luoda erilaisia graafeja ja kaavioita ja käyttää niitä omilla verkkosivuratkaisuissaan. (Using Google Charts 11.2.2015) Vaihtoehtoja on aina hyvin yksinkertaisista viiva-diagrammeista monimutkaisiin skaalattaviin ja laajasti tietoa sisältäviin graafeihin.

4 PORTAALIN SUUNNITTELU

Järjestelmän suunnittelu aloitettiin kartoittamalla Service Desk –yksikössä, minkälaisille raporteille ja tiedolle olisi tarvetta päivittäisen toiminnan tukemiseksi. Kartoitusvaiheessa kerättiin näkemyksiä ja ideoita siitä, mitä järjestelmä voisi sisältää, minkälaisista raporteista ja tiedoista olisi eniten hyötyä ja miten tietoa haluttaisiin näytettävän. Kartoitusvaiheesta koostettiin lista, josta valittiin toteutuskelpoisimmat ideat ja ryhdyttiin kehittämään ensimmäisenä niitä eteenpäin.

Ideoita saatiin hyvin, osa toteutuskelpoisista ideoista jäi odottamaan myöhempää toteuttamista. Ideoissa korostui tiedon helppo saavutettavuus sekä ajankohtaisen tiimi- ja asiakaskohtaisen tiedon sekä koostettavan historiatiedon tarve. Kyseiset asiat päätettiin suunnitteluvaiheessa nostaa järjestelmän ensimmäisen version tavoitteeksi.

Tiedon hakemista varten ryhdyttiin kartoittamaan MyEnfon rajapintamahdollisuuksia järjestelmätoimittaja ServiceNow'n ohjeistuksista. ServiceNow:n tarjoamat rajapinnat osoittautuivat selkeiksi ja rajapintojen testikäyttö onnistui ilman erityisiä ongelmia.

Toisena rajapintana selvitettiin Enfon käyttämän puhelinjärjestelmän SAP BCM:n rajapintoja. Tiedon hakeminen järjestelmästä osoittautui MyEnfon rajapintoja monimutkaisemmiksi ja vaativat enemmän suunnittelua kuinka dataa tullaan järjestelmästä hakemaan ja kuinka siitä saadaan koostettua haluttu tieto.

Järjestelmän käyttöliittymäksi suunniteltiin web-käyttöliittymää, jota voi käyttää selaimella sekä näkymiä näyttävät monitoriruudut seinille. Näiden avulla järjestelmä saadaan mahdollisimman laajan käyttäjäkunnan saavutettavaksi ja helposti hyödynnettäväksi. Seinänäytöillä suunniteltiin näytettävän oleellisinta tietoa yleisellä tasolla, selainkäyttöliittymän kautta pääsee tietoihin ja raporteihin pureutumaan tarkemmin.

Järjestelmään suunniteltiin myös hallintapaneelia, jossa järjestelmään syötettyjä tiimejä ja asiakkaita sekä muita tietoja voi hallita. Tämä kuitenkin päätettiin jo alkuvaiheessa jättää myöhemmin toteuttavaksi ja hoitaa alkuvaiheessa hallinta suoraan tietokannasta käsin ja keskittyä ensin tietoa hakevan ja näyttävän järjestelmän toteuttamiseen.

Suunnitteluvaiheessa päätettiin ohjelmassa käytettävät toteutustekniikat. Merkittävimpinä vaihtoehtoina olivat ASP.NET MVC 4 ja PHP, joista ASP.NET MVC 4 valikoitui nopeasti toteutustekniikaksi. Toteutustekniikan ansiosta kehitysympäristöksi tuli Microsoft Visual Studio. Suunnitteluvaiheessa ei tehty tiukkoja linjauksia toteutusvaiheen lopputuloksista vaan annettiin tilaa myös toteutusvaiheessa esille tuleville tarpeille ja ideoille.

5 PORTAALIN TOTEUTUS

Työn toteutus aloitettiin luomalla pohjaksi ASP.NET MVC -projekti Visual Studiolla, johon ohjelman runkoa alettiin hahmotella. Tiedonhakua varten alussa testattiin lähdejärjestelmiä, jotta selviäisi, mitä tietoa ja miten järjestelmään on mahdollista saada portaalin käytettäväksi.

Suunnitteluvaiheessa kartoitettujen toiveiden ja lähdetiedon järjestelmiin tutustumisen avulla saatiin ensimmäinen yhteenveto siitä, mitä dataa lähdetään lähdejärjestelmistä hakemaan ja raporttijärjestelmässä näyttämään. Suunnitelmien pohjalta voitiin järjestelmään suunnitella tarvittavat tietomallit. Tässä vaiheessa huomioitiin jo, että datan määrää lisätään myöhemmin, kun järjestelmä saadaan ensin toimimaan suppeammalla tietomäärällä.

Tiedon hakemiseen lähdejärjestelmistä toteutettiin alkuun C#-pohjaiset ohjelmat, jotka hakivat dataa ja tallensivat sen ilman käsittelyä tiedostoon. Tiedostojen avulla voitiin testata tiedon haun onnistumista ja oikeellisuutta. Kun haun tuloksena saatu tieto oli oikeaa, voitiin toteuttaa ohjelmiin ominaisuuksia, jotka laskivat saadusta datasta lukuja suoraan portaalin kannan tarvitsemaan muotoon.

Järjestelmän testitietokanta oli alkuvaiheessa paikallinen SQL-kanta, johon testidataa syöttämällä voitiin testata tietomallien ja testinäkömien toimintaa. Tässä vaiheessa tiedon hakemista ja näyttämistä varten olevat osat toimivat erillään toisistaan eikä kantaan saatu vielä oikeaa dataa.

Testinäkömän onnistuneiden testausten jälkeen voitiin ryhtyä toteuttamaan järjestelmään ensimmäisiä varsinaisia näkymiä. Ensimmäisenä näkömänä toteutettiin päänäkömän ensimmäinen versio eli järjestelmän etusivu, jossa näkyy yleisellä tasolla tiimien tilanne sekä mahdollista muuta yleistä tietoa. Päänäkömän toteuttamisen jälkeen toteutettiin tiiminäkömän ensimmäinen versio, joka keskittyy yksittäisen tiimin kannalta tärkeän tiedon näyttämiseen.

Päänäkömä ja tiiminäkömä keskittyivät teksti- ja lukutyypin datan näyttämiseen. Historiadataa varten toteutettiin graafinäkömät. Näiden tekniseen toteuttamiseen valikoitui toteuttamisvaiheessa Google Charts, jolla saatiin kätevästi luotua toimivat ratkaisut historiadataan esittämiseen. Ensimmäisinä graafinäköminä toteutettiin tiimikohtaisten puhelu- ja keikkamäärien historiagraafit.

Järjestelmää varten pystytettiin virtuaalinen Microsoft Windows Server 2008R2 -palvelin, johon pystytettiin IIS sekä Microsoft SQL Server. Kun palvelin saatiin käyttöön, voitiin ASP.NET MVC -projektin tietokanta siirtää palvelimelle.

Tietokannan testauksessa ja hallinnassa käytettiin Microsoft SQL Server Management Studiota, joka osoittautui erittäin käytännölliseksi työkaluksi eri työvaiheisiin, kuten datan oikeuden varmistamiseen, testidatan syöttämiseen sekä datan varmuuskopiointiin ja siirtämiseen toiseen versioon projektista.

Palvelimen pystyttämisen jälkeen voitiin järjestelmän tietoa hakeviin osiin toteuttaa ominaisuuksia, jotka hyödynsivät järjestelmän tietokantaan syötettyjä asiakas ja tiimikohtaisia tietoja.

Näiden tietojen avulla ohjelmat parsivat haetun tiedon suoraan järjestelmän kannan mukaiseen muotoon ja parsitun tiedon tallentaminen onnistui suoraan tietokantaan.

Järjestelmän MCV-projekti siirrettiin IIS-palvelimelle testattavaksi ja toteutettiin tarvittavat määritykset, jotta raporttiportaalin ensimmäinen testiversio saatiin näkymään työaseman selaimelta. Onnistuneiden testausten ja toteutettujen korjausten jälkeen voitiin viedä järjestelmä testattavaksi käyttäjille.

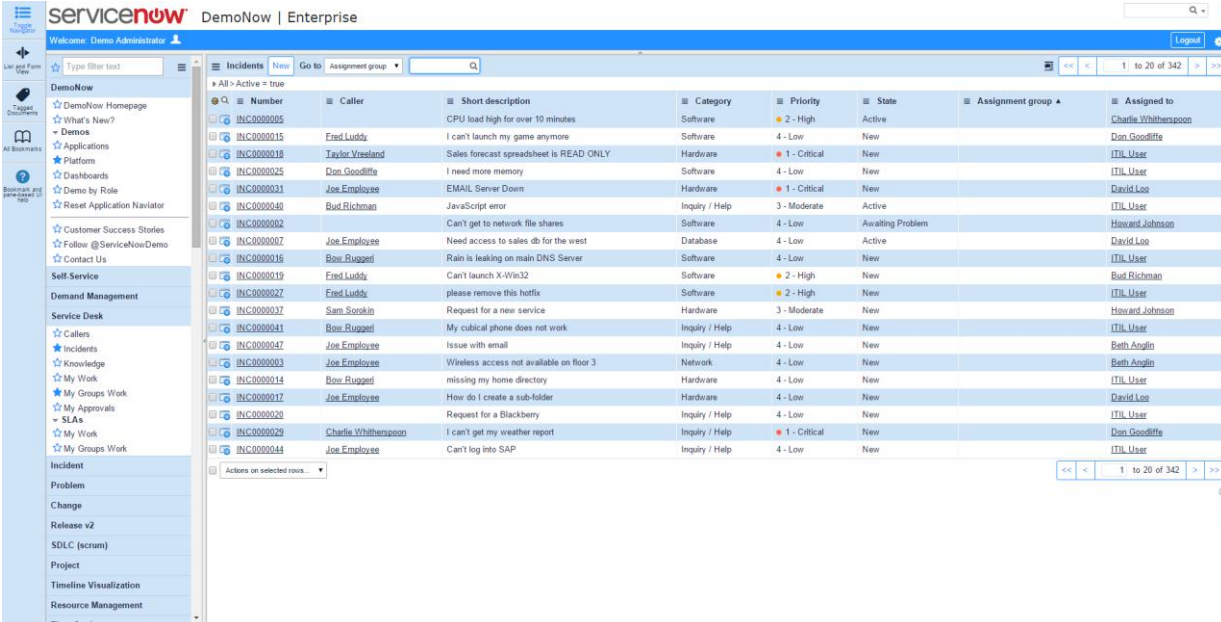
6 TIEDON LÄHDEJÄRJESTELMÄT

Järjestelmän tiedon lähdejärjestelmiä ovat Service Deskin käytössä olevat ITSM-järjestelmä MyEnfo sekä puhelinjärjestelmä SAP BCM.

6.1 MyEnfo

MyEnfo on monipuolinen ja tehokas ServiceNow -pohjainen ITSM-ratkaisu, jolla mm. hallitaan ja toteutetaan työpyyntöjä ja häiriöilmoituksia.

ServiceNow on täysin ITIL-yhteensopiva skaalautuva pilvipalvelupohjainen ITMS-ratkaisu (ServiceNow 11.2.2015). Kuvassa 4 näkyy ServiceNow-järjestelmän demon näkymä, jossa listataan järjestelmässä olevia aktiivisia häiriöilmoituksia.



Number	Caller	Short description	Category	Priority	State	Assignment group	Assigned to
INC0000095		CPU load high for over 10 minutes	Software	2 - High	Active		Charlie Witherspoon
INC0000015	Fred Ludlo	I can't launch my game anymore	Software	4 - Low	New		Don Goodliffe
INC0000018	Taylor Vreeland	Sales forecast spreadsheet is READ ONLY	Hardware	1 - Critical	New		ITIL User
INC0000025	Don Goodliffe	I need more memory	Software	4 - Low	New		ITIL User
INC0000031	Joe Employee	EMAIL Server Down	Hardware	1 - Critical	New		David Loo
INC0000040	Bud Richman	JavaScript error	Inquiry / Help	3 - Moderate	Active		ITIL User
INC0000052		Can't get to network file shares	Software	4 - Low	Awaiting Problem		Howard Johnson
INC0000007	Joe Employee	Need access to sales db for the west	Database	4 - Low	Active		David Loo
INC0000016	Row Ruggert	Rain is leaking on main DNS Server	Software	4 - Low	New		ITIL User
INC0000019	Fred Ludlo	Can't launch X-Win32	Software	2 - High	New		Bud Richman
INC0000027	Fred Ludlo	please remove this hotfix	Software	2 - High	New		ITIL User
INC0000037	Sam Sorenin	Request for a new service	Hardware	3 - Moderate	New		Howard Johnson
INC0000041	Row Ruggert	My cubical phone does not work	Inquiry / Help	4 - Low	New		ITIL User
INC0000047	Joe Employee	Issue with email	Inquiry / Help	4 - Low	New		Beth Anglin
INC0000003	Joe Employee	Wireless access not available on floor 3	Network	4 - Low	New		Beth Anglin
INC0000014	Row Ruggert	missing my home directory	Hardware	4 - Low	New		ITIL User
INC0000017	Joe Employee	How do I create a sub-folder	Hardware	4 - Low	New		David Loo
INC0000020		Request for a Blackberry	Inquiry / Help	4 - Low	New		ITIL User
INC0000029	Charlie Witherspoon	I can't get my weather report	Inquiry / Help	1 - Critical	New		Don Goodliffe
INC0000044	Joe Employee	Can't log into SAP	Inquiry / Help	4 - Low	New		ITIL User

KUVA 4. Service Now -demonäkymä

6.2 SAP BCM

SAP BCM on työasemilla käytettävä puhelinjärjestelmä. Nykyiseltä nimeltään järjestelmä on SAP Contact Center. (SAP 11.2.2015) Alun perin suomalaisyritys Wicomin kehittämä ratkaisu on nykyisin osa SAPin tarjoamia palveluratkaisuja.

7 INFO- JA RAPORTTIPORTAALI

7.1 Mallit

Järjestelmän toteuttamisen ensimmäisiin vaiheisiin kuului järjestelmään sopivien mallien luominen ASP.NET MVC -projektiin tehtyjen suunnitelmien pohjalta. Tietomallien oikeanlainen suunnittelu oli äärimmäisen tärkeää järjestelmän onnistumisen ja jatkokehitysmahdollisuuksien kannalta.

Visual Studio luo toteutettujen mallien pohjalta tietokantatiedoston tai tietokannan, jonne testidataa syöttämällä voidaan suunnitella sekä testata suunniteltujen mallien toimivuutta. Tietokantaa ei tarvinnut suunnitella erikseen, vaan se mukautuu automaattisesti tietomallin mukaisesti. Tämän takia tietomalleja ei jälkikäteen muokata vapaasti, koska se vaikuttaa myös tietokannan rakenteeseen. Mallien suunnittelussa täytyi ottaa huomioon mallien laajentamismahdollisuudet jatkossa, kun järjestelmän tietomäärää lisätään.

Mallit voivat sisältää muuttujia, vierasavaimia, joilla viitataan toiseen malliin, sekä listoja jotka sisältävät muita tietomalleja, joihin on määritelty kyseisen mallin vierasavain.

Järjestelmään luotiin mallit asiakkaille, tiimeille sekä asiakkaiden ja tiimien ajankohtaiselle ja historiatatalle. Kuvassa 5 esimerkkinä Customer-mallin rakenne.

```

}
    public class Customer
    {
        public int ID { get; set; }
        public string Name { get; set; }
        public string MyEnfoId { get; set; }

        [ForeignKey("Team")]
        public int TeamID { get; set; }
        public virtual Team Team { get; set; }

        public List<Line> Lines { get; set; }
        public List<CustomerNow> CustomerNow { get; set; }
        public List<CustomerStats> CustomerStats { get; set; }
    }
}

```

KUVA 5. Customer-mallin rakenne

Customer mallissa ovat:

- asiakkaan yksilöivä id
- asiakkaan nimi
- MyEnfoId, jolla asiakas voidaan yhdistää MyEnfosta haettaviin tietoihin
- tiimin id ja viittaus tiimin tietoihin.
- lista asiakkaan Lines -tiedoista
- lista asiakkaan nykyhetken tietoa sisältäviin CustoNow – tietoihin
- lista asiakkaan historiatiedoista.

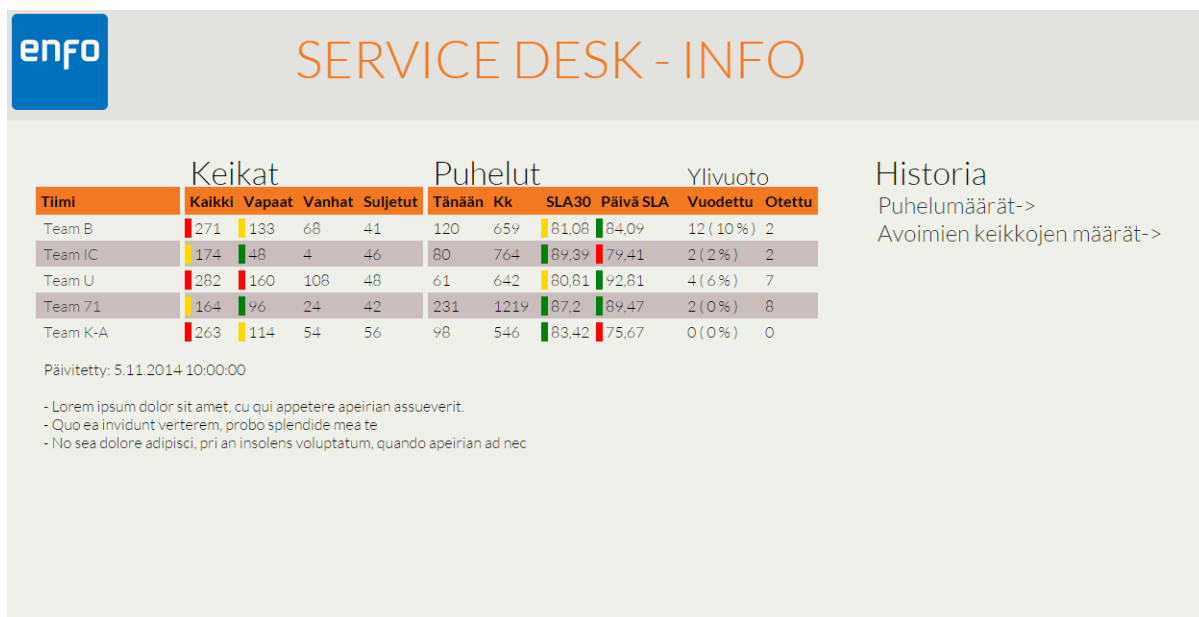
7.2 Näkymät

Järjestelmän näkymät toteutettiin MVC-mallin mukaisesti. Näkymät toteutettiin käytettäväksi internetiselaimilla. Toteutuksessa pääperiaatteena oli näkymien selkeys ja helppokäyttöisyys ja tässä onnistuttiin hyvin.

Näkymien luominen MVC-projektiin on yksinkertaista ja tiedon hakemiseen näkymään voidaan käyttää samoja kontroleja, jos tietyissä näkymissä on tarve hakea samantyyppistä tietoa. Näkymien näkymiseen vaikuttavat myös kontrollerille verkko-osoitteessa syötetyt parametrit, joilla voidaan määrittellä esimerkiksi minkä tiimin tiedot näytetään sivulla.

7.2.1 Päänäkymä

Päänäkymässä näytetään keskeisintä ajankohtaista tilannetietoa tiimikohtaisesti kuvan 6 mukaisesti. Näkymässä näytetään myös ajankohtaisia tiedotteita.



KUVA 6. Portaalin päänäkymä

Klikkaamalla tiimin nimeä pääsee sivulta etenemään tiimikohtaiselle sivulle. Näkymässä on myös linkit puhelumäärien ja avoimien keikkojen määrän tiimitasolla oleviin historianäkymiin.

Näkymässä näkyvät myös kaikkien, vapaiden ja vanhojen keikkojen määrien sekä puheluiden SLA-tietojen edessä väri-indikaattorit, jotka kertovat kuinka hyvin arvot ovat niille etukäteen määritettyjen tavoite-arvojen sisällä. Arvot voidaan määrittellä tiimikohtaisesti ja näiden hallinta tulee myöhemmin toteutettavaan hallintapaneeliin. Näkymän tiedot päivittyvät tietokantaan tunnin välein.

7.2.2 Tiimikohtainen näkymä

Tiimikohtaisessa näkymässä pureudutaan yksittäisen tiimin yleisiin raporttitietoihin sekä kyseisen tiimin asiakaskohtaisiin keikka ja puhelumääriin.

Team 71		Keikat									SLA %					Puhelut				
Asiakas	Ritm	Inc	Vapaat	Pending	Old	Suljetut	Email	Response	Resolve	Tänään	Kk	SLA60	SLA30	Päivä SLA30						
Customer A	97	9	67	29	18	26	96	98	100	19	61	95,08	93,44	94,74						
Customer B	6	4	4	6	1	8	100	100	100	8	34	82,35	73,53	75						
Customer C	22	8	12	14	5	5	85	94	100	7	19	89,47	89,47	100						
Customer 46	1	1	1	1	0	0	66	100	100	0	1	100	100	100						
Customer 49	6	5	10	1	0	2	55	78	100	0	4	100	100	100						
Customer 50	0	0	0	0	0	0				0	0	100	100	100						
Customer H16	0	0	0	0	0	0				4	6	100	83,33	75						
Customer 78	0	0	0	0	0	0	100	100	100	0	0	100	100	100						

Avoimet	164
RITM	137
INC	27
Pending	51
Hold	18
Vapaat	96
Vanhat	24
Suljetut	42
Puhelut	231
Puhelut kk	1219
SLA30	87,2
SLA60	91,2
Päivä SLA30	89,47
Päivä SLA60	94,74

<-alkusivu Historia->

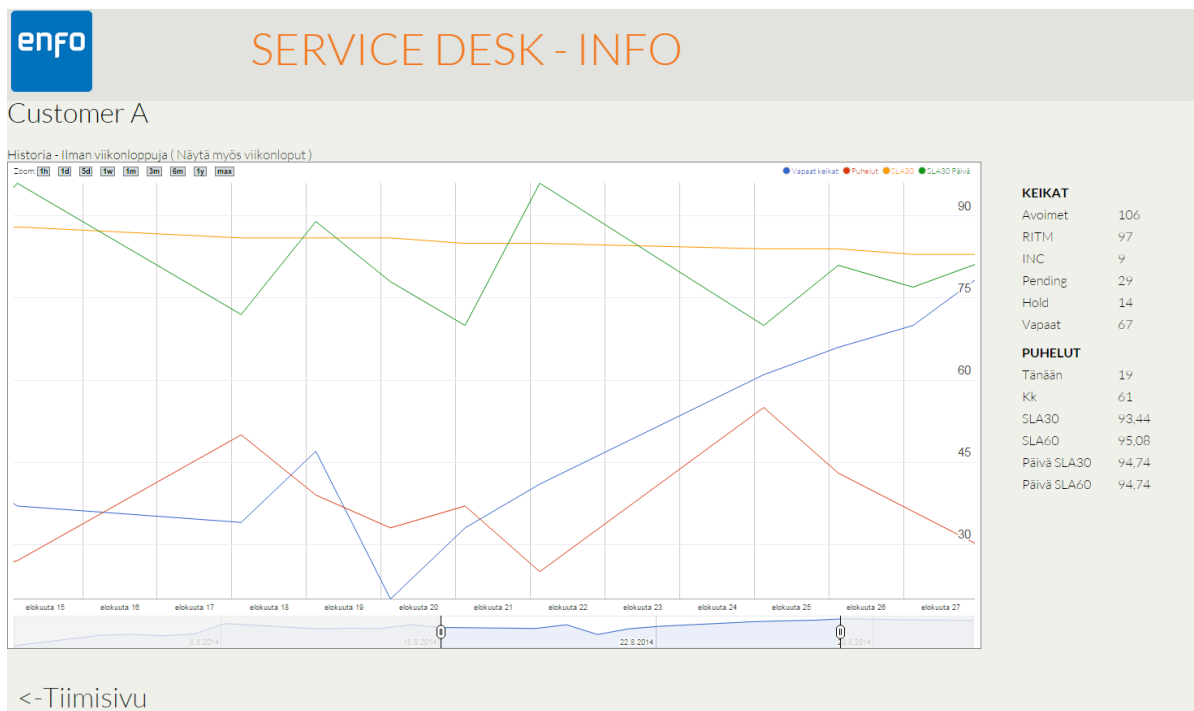
KUVA 7. Tiimikohtainen näkymä

Tiimikohtaisella sivulla on kuvan 7 mukaisesti allekkain tiimille kuuluvat asiakkaat ja niiden keikkamäärien tilanne eriteltynä työpyynnöiksi ja häiriöilmoituksiksi. Näkymällä näytetään myös vapaiden, odottavien, vanhojen ja suljettujen keikkojen määrä sekä keikkojen SLA-tilanne. Puhelutiedoista näkymässä näytetään puhelumäärät ja SLA – tiedot kuluvalta päivältä ja kuukaudelta.

Näkymässä on historia-linkki, josta käyttäjä pääsee tiimikohtaisen historian näyttävään graafinäkymään. Näkymässä on myös linkki, jolla voi palata takaisin alku eli päänäkymään.

7.2.3 Asiakaskohtainen näkymä

Asiakaskuntaisessa näkymässä näytetään sekä ajantasaista tietoa että historiatietoa asiakkaan keikkojen ja puheluiden määrästä kuvan 8 mukaisesti.



KUVA 8. Asiakaskohtainen näkymä

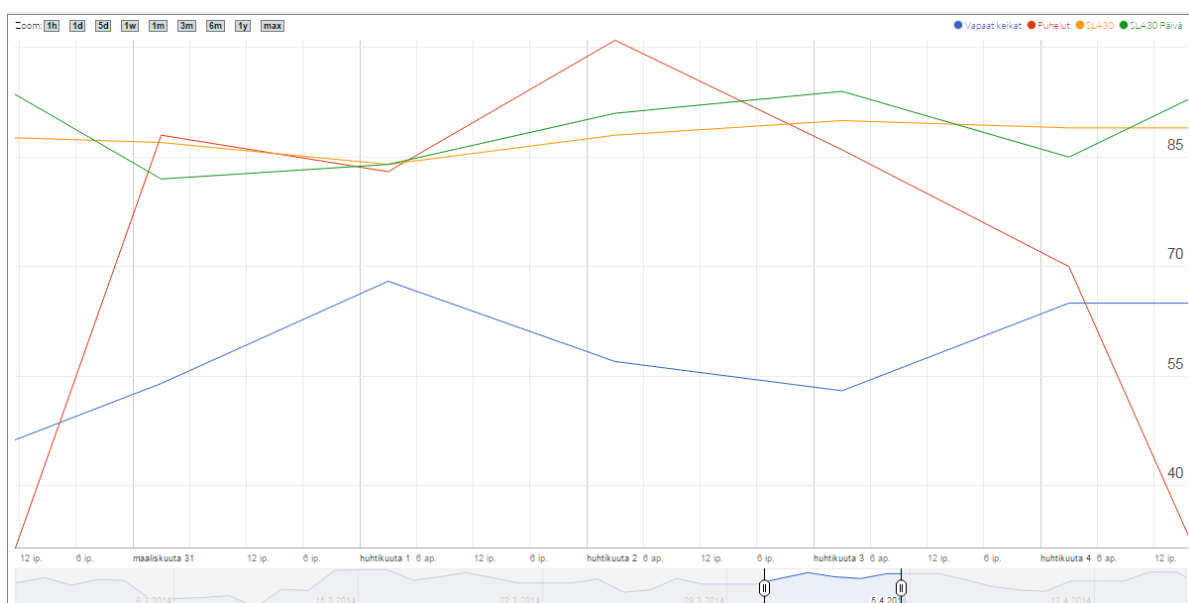
Asiakaskohtaisessa näkymässä on myös linkki takaisin tiimin näkymään.

7.2.4 Historiagraafit

Järjestelmään toteutettiin graafit, jotka näyttävät historiadataa tiedoista. Graafit toteutettiin Google Chartsin "Annotation Charts" – työkalun avulla. Google Chartsin näkymät käyttävät mm. JavaScriptia ja HTML5 – ominaisuuksia sekä tiedon välittämiseen JSON:ia.

Graafi kutsuu kontrolleria joka hakee tietoa tietokannasta ja näyttää näkymän mukaiset tiedot päivä-järjestyksessä viivadiagrammeina. Graafia voi skaalata tietyn mittaisella aikavälillä ja tai määritellä itse miltä aikaväliltä tieto näytetään. Kuva 9 esittää esimerkin historiagraafista.

Kyseiset Googlen graafit sopivat hyvin käyttötarkoitukseen ja tuovat myös mahdollisuuden lisätä kommentteja graafien eri tapahtumiin, joiden avulla voidaan jälkikäteen paremmin havainnoida mistä mikäkin poikkeama tai muu tapahtuma graafille johtuu.



KUVA 9. Historiagraafi

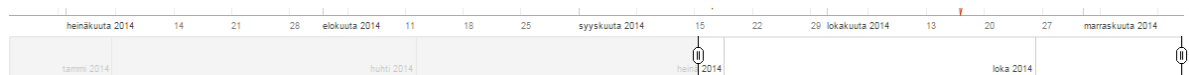
Näytettävää aika-aluetta voi rajata valitsemalla ylhäältä zoomaus-tason.



KUVA 10. Zoomaus-tason valinta

Zoomauksen voi rajata kuvan 10 mukaisesti aikaväliltä 1 tunti – 1 vuosi, tai valita maksimin, joka näyttää kaiken näytettävän datan linjalla.

Aika-alueen voi valita myös tarkalta ajanjaksolta vetämällä alhaalta palkkien väliin haluansa aikavälin kuvan 11 mukaisesti.



KUVA 11. Historiagraafin aikajana

Osaan historiagraafeista toteutettiin myös valinta, jolla pystyy valitsemaan, näytetäänkö viikonloput mukana tiedossa.

7.2.5 Seinänäyttönäkymä

Seinänäyttönäkymäksi toteutettiin yksinkertaistetumpi versio päänäkymästä sekä ajastetusti vaihtuvat seinänäkymää varten toteutetut graafinäkymät kuvaamaan tiettyjen arvojen kehitystä.

7.3 Kontrollerit

Järjestelmään toteutettiin kontrollerit, jotka ohjaavat näkymien ja näkymässä tarvittavan datan toimintaa. Käyttäjän siirtyessä toiselle sivulle kutsuu järjestelmä kontrolleria, joka hakee tarvittavat tiedot ja ohjaa ne näkymälle ja käyttäjän näkymään. Interaktiivisissa näkymissä näkymä kutsuu kontrolleria, joka hoitaa varsinaisen kutsun käsittelyn ja tuo uutta dataa näkymään.

```
public ActionResult Info()
{
    return View(db.TeamNow.ToList());
}
```

KUVA 12. Info-kontrolleri

Kuvassa 12 Info-niminen kontrolleri ohjaa käyttäjän Info-nimiseen näkymään ja syöttää näkymän käytettäväksi TeamNow-mallin mukaisen datan listana.

7.4 Datan hakeminen ja muodostaminen

Datan hakemiseen ja muodostamiseen toteutettiin sovellukset, jotka hakevat MyEnfosta ja SAP BCM-puhelinjärjestelmästä tietoa ajastetusti.

Lähdejärjestelmistä haettu data ei ole valmista raporttidataa, vaan ohjelmat laskevat ja muodostavat tarvittavat raporttiluvut käyttäen hyväksi raporttijärjestelmään määriteltyjä tiimi ja asiakaskohtaisia määrittämiä. Kaikki järjestelmässä näytettävä data on sen omassa tietokannassa, eikä järjestelmä hae lähdejärjestelmistä erikseen mitään tietoja näkymien muodostamiseen käyttäjän selatessa näkymiä.

Tietoa hakevien ohjelmien suunnittelussa piti ottaa huomioon, että niiden täytyy kyetä toimimaan itsenäisesti ja niiden on oltava varmatoimisia, jotta vikatilanteita ei syntyisi. Tiedon hakemiseen käytettiin SQL-kyselyjä ja olemassa olevaa web serviceä, jonka avulla saatiin haettua XML-muotoista dataa.

8 TESTAUS

Järjestelmää testattiin toiminnallisuuden ja järjestelmän esittämän raporttidatan paikkaansa pitävyyden kannalta. Toiminnallista testausta suoritettiin jatkuvasti järjestelmän kehitysvaiheessa osien valmistuessa. Dataa hakevien osien toimivuutta testattiin seuraamalla ajastettujen tehtävien onnistumista ja käymällä läpi esiin tulevat virhetilanteet.

Raporttidatan paikkansapitävyyttä testattiin ajamalla satunnaisesti samaa tietoa jo aiemmin käytössä olevista raporttityökaluista ja vertaamalla sitä järjestelmän tuottamaan dataan. Dataa katselmoitiin myös silmämääräisesti läpi ja etsittiin mahdollisia poikkeamia.

Järjestelmä osoittautui testikäytössä vakaaksi ja varmatoimiseksi.

9 YHTEENVETO

Työn tavoitteena oli luoda Enfo Oyj:n Service Deskin käyttöön toimiva ja käytännöllinen järjestelmä, jolla voidaan seurata erilaisia ajankohtaisia raportti- ja infotietoja helposti ja monipuolisesti.

Järjestelmän toteutus onnistui erittäin hyvin ja suuremmilta vastoinkäymisiltä vältyttiin. Järjestelmästä tuli selkeä ja helppokäyttöinen: näkymien käyttö onnistuu hyvin verkkoselaimilla ja se soveltuu myös seinänäytölle. Historiagraafien toteuttamiseen valittu Google Charts osoittautui hyväksi valinnaksi ja antaa myös mahdollisuuksia jatkokehittää historianäkymien muotoa ja datan näyttämistä. Järjestelmän kehittämisessä päänvaivaa aiheutti ajoittain selainyhteensopivuuden takaaminen kaikille yleisimmille selaimille.

Yhteistyö toimeksiantajan kanssa sujui ongelmitta. Järjestelmän eri kehitysvaiheissa asioiden toteuttamiseen sai hyvin palautetta ja ideoita. Palaverit sujuivat hyvässä hengessä ja työn lopputuloksesta oli koko ajan yhteinen näkemys.

Järjestelmän kehittäminen oli todella mielenkiintoista ja opettavaista. Työssä oppi hyödyntämään monipuolisesti ASP.NET MVC:n tuomia mahdollisuuksia, tiedon hakemista lähdejärjestelmistä ja käyttöliittymäsuunnittelua. Työssä pääsi tutustumaan myös SQL-kantoihin ja IIS-palvelimen toimintaan.

Järjestelmä on saanut käyttäjiltä hyvän vastaanoton ja on ahkerassa päivittäisessä käytössä Service Deskissä. Järjestelmän kehittämistä on jatkettu ja siihen on lisätty tietoa sekä näkymiä.

Jos järjestelmän kehittäminen aloitettaisiin alusta, voisi järjestelmän tietomallien suunnittelua tehdä enemmän yleisellä tasolla. Tällöin järjestelmän käyttökohteiden laajentaminen olisi helpompaa eikä vaatisi suuria muutoksia järjestelmään.

10 LÄHTEET

ABOUT JQUERY. [verkkosivu] [Viitattu 2015-02-11] Saatavissa:

<http://learn.jquery.com/about-jquery/>

ASP.NET MVC OVERVIEW. [verkkosivu] [Viitattu 2015-02-11] Saatavissa:

<https://msdn.microsoft.com/en-us/library/dd381412%28v=vs.108%29.aspx>

ENFO. [verkkosivu] [Viitattu 2015-02-11] Saatavissa:

<http://www.enfo.fi>

SAP. SAP Contact Center Software. [verkkosivu] [Viitattu 2015-02-11] Saatavissa:

<http://www.sap.com/solution/lob/customer-svc/software/call-contact-center/index.html>

SERVICENOW. ServiceNow IT Service Management. [verkkosivu] [Viitattu 2015-02-11] Saatavissa:

<https://www.servicenow.com/solutions/it-service-management.html>

USING GOOGLE CHARTS. [verkkosivu] [Viitattu 2015-02-11] Saatavissa:

<https://developers.google.com/chart/interactive/docs/index>

MICROSOFT. Visual Studio, Application Development. [verkkosivu] [Viitattu 2015-02-11] Saatavissa:

<http://www.visualstudio.com/explore/application-development-vs>

W3SCHOOLS.COM. HTML Introduction. [verkkosivu] [Viitattu 2015-02-11] Saatavissa:

http://www.w3schools.com/html/html_intro.asp