



SAVONIA

Tiedonhallintajärjestelmän suunnittelu ja toteutus

Jarkko Tervakoski

Opinnäytetyö

Ammattikorkeakoulututkinto

Koulutusala Tekniikan ja liikenteen ala	
Koulutusohjelma Tietotekniikan koulutusohjelma	
Työn tekijä(t) Jarkko Tervakoski	
Työn nimi Tiedonhallintajärjestelmä	
Päiväys 8.5.2013	Sivumäärä/Liitteet 37/2
Ohjaaja(t) tutkimuspäällikkö Matti Sipilä, ohjelmistosuunnittelija Mikko Pääkkönen	
Toimeksiantaja/Yhteistyökumppani(t) Itä-Suomen yliopisto / tutkimusjohtaja Mikko Hiltunen	
Tiivistelmä <p>Työn tilaajana toimi Itä-Suomen yliopiston neurologian laitos. Työn tavoitteena oli rakentaa laitoksen käyttöön tiedonhallintajärjestelmä, jonka avulla voitiin varmistaa tiedon eheys ja säilyminen sekä helpottaa tarvittavan tiedon etsimistä. Järjestelmä sijoitettiin alustavasti Tietotekniikkakeskuksen palvelimelle, josta järjestelmä myöhemmin siirretään yliopiston omalle palvelimelle.</p> <p>Työ sisälsi käyttöliittymän ja tietokannan suunnittelun sekä toteuttamisen alusta loppuun. Apuna suunnittelussa käytettiin asiakkaan järjestelmälle suunnattuja tavoitteita sekä opinnäytetyön ohjaajien neuvoja järjestelmän rakentamisessa.</p> <p>Tietokanta rakennettiin Microsoft SQL Serverille Microsoftin omalla SQL Server Management Toolsin avulla. Itse varsinainen sovellus toteutettiin ASP.NET MVC 3:lla ja kehitystyökaluna käytettiin Microsoft Visual Studio 2010:ä. Muita tekniikoita olivat LINQ, Entity Framework sekä jQuery. Virheraportoinnista vastasi ELMAH-kirjasto.</p> <p>Työ saatiin loppuun joulukuussa 2012 ja järjestelmä otettiin käyttöön laitokselle. Järjestelmän käyttöönotossa ei havaittu mitään suuria virheitä, etenkin mitään sellaisia, jotka voisivat estää järjestelmän käytön. Järjestelmää on myös helppo jatkokehittää sen modulaarisuuden vuoksi.</p>	
Avainsanat tiedonhallintajärjestelmä, ASP.NET MVC 3, MSSQL	

Field of Study Technology, Communication and Transport			
Degree Programme Degree Programme in Information Technology			
Author(s) Jarkko Tervakoski			
Title of Thesis Data Management System			
Date	8 May 2013	Pages/Appendices	37/2
Supervisor(s) Mr. Matti Sipilä, Research Manager, Mr. Mikko Pääkkönen, Software Engineer			
Client Organisation /Partners University of Eastern Finland / Mr. Mikko Hiltunen, Research Director			
<p>Abstract</p> <p>The thesis was commissioned by the department of neurology from University of Eastern Finland. The goal was to build a data management system for the department which would secure the integrity and the preservation of the data and would also make it easier for employees to search the needed data. The system was tentatively set to work on IT Centre servers but it would be later moved to the departments own server.</p> <p>The system included a user interface and a database which were planned and implemented from the start to the end. To help plan the system, the university told about their goals and wishes on how the system should work and based on those requirements the system was planned. Also both thesis supervisors advised in building the system.</p> <p>The database was built on Microsoft SQL Server with Microsoft's own SQL Server Management Tool. The actual system was made with ASP.NET MVC 3 and the primary development program was Microsoft Visual Studio 2010. Other programming techniques were LINQ, Entity Framework and jQuery. ELMAH library, which is also known as Error Logging Modules and Handling, was used to handle errors which might happen in the system.</p> <p>The work was completed in December 2012 and the system was deployed for the department's use. While deploying the system no fatal errors were detected and definitely nothing that could prevent the system usage. It will also be easy to continue to develop the system because of the modularity of the system.</p>			
<p>Keywords Data Management System, ASP.NET MVC 3, MSSQL</p>			

ALKUSANAT

Opinnäytetyö tehtiin Itä-Suomen yliopiston neurologian laitokselle. Työ aloitettiin keväällä 2012 ja käytännön osuus saatettiin loppuun vuoden 2012 loppuun mennessä. Iso kiitos ohjaajilleni tutkimuspäällikkö Matti Sipilälle ja ohjelmistosuunnittelija Mikko Pääkköselle ohjaamisesta ja opinnäytetyön eteenpäin viemisestä sekä järjestelmän tekoon auttavista neuvoista. Kiitos myös projekti-insinööri Helena Meriläiselle käyttöösi liittymän parantamiseen annetuista ohjeista sekä Itä-Suomen Kuopion yksikön neurologian laitoksen henkilökunnalle ja etenkin tutkimusjohtaja Mikko Hiltuselle sekä laboratorioinsinööri Esa Koivistolle sulavasta yhteistyöstä ja avusta projektin aikana.

Kuopiossa 8.5.2013

Jarkko Tervakoski

SISÄLTÖ

1	JOHDANTO	9
2	ITÄ-SUOMEN YLIOPISTO	10
3	TYÖVAIHEET	11
3.1	Tietokannan suunnittelu ja toteutus.....	11
3.2	Käyttäjien todentaminen.....	11
3.3	Käyttöliittymä ja sovelluslogiikka	12
3.4	Käyttöönotto.....	12
4	TEKNIIKAT	14
4.1	MVC.....	14
4.1.1	Malli	14
4.1.2	Näkymä.....	15
4.1.3	Käsittelijä.....	15
4.2	ASP.NET MVC 3	16
4.3	Entity Framework.....	16
4.4	LINQ.....	17
4.5	JQuery	17
5	OHJELMISTOT	18
5.1	Microsoft Visual Studio 2010	18
5.2	Microsoft SQL Server Express 2008 R2.....	18
5.3	Microsoft SQL Server Management Studio	19
5.4	IIS - Internet Information Services	19
6	TIETOKANTA	20
6.1	Yleistä tietoa.....	20
6.2	Suunnittelu.....	20
6.3	Rakenne.....	20
6.3.1	KMSProcess	21
6.3.2	Plan.....	21
6.3.3	Format	21
6.3.4	Tiedosto-taulut.....	21
7	KÄYTTÖLIITTYMÄ	22
7.1	Peruskäyttäjän näkymät	22
7.1.1	Aloitussivu	23
7.1.2	Kirjautuminen.....	23
7.1.3	Rekisteröinti.....	23

7.1.4 Omien tietojen muokkaaminen	23
7.1.5 Uusi prosessi	24
7.1.6 Etsi prosessi	25
7.1.7 Omat prosessit.....	25
7.1.8 Prosessin suunnitelma	26
7.1.9 Formaattit	27
7.1.10 Prosessin analyysi	28
7.1.11 Prosessin lopputulos.....	29
7.2 Hallintapaneeli.....	30
7.2.1 Käyttäjienhallinta	30
7.2.2 Virheiden seuranta	31
7.2.3 Sivuston tekstien muokkaaminen.....	31
8 VIRHEENTARKISTUS JA -RAPORTOINTI.....	33
8.1 Virheentarkistus.....	33
8.2 Virheraportointi	33
9 TESTAUS.....	34
10 YHTEENVETO	35
LÄHTEET	37

LIITTEET

- Liite 1. Tietokantakaavio - Yleinen järjestelmä
- Liite 2. Tietokantakaavio - Käyttäjien todentaminen

TERMIT JA LYHENTEET

ASP.NET MVC	Microsoftin web-ohjelmistokehys, Model - View - Controller
C#	.Net ohjelmointikieli
.NET	Microsoftin kehittämä ohjelmistokomponenttikirjasto
LINQ	Language Integrated Query
EF	Entity Framework
MSSQL	Microsoft SQL Server
JQUERY	Avoimen lähdekoodin JavaScript-kirjasto
SSMS	SQL Server Management Studio
SQL	Structured Query Language
Razor	Asp.Net MVC 3:ssa mukana tullut näkymämoottori
HTML	Hypertext Markup Language
CSS	Cascading Style Sheets
IIS	Internet Information Services
ELMAH	Error Logging Modules and Handlers

1 JOHDANTO

Opinnäytetyössä tarkastellaan Itä-Suomen yliopiston neurologian laitokselle tehtävää tiedonhallintajärjestelmää. Asiakkaan pyynnöstä aloitettiin rakentaa tiedonhallintajärjestelmää, joka tulisi käyttöön AlzPoint-nimisen hankkeen johdosta. Järjestelmää tullaan käyttämään myös yleisesti neurologian laitoksen tutkimustulosten säilömiseen ja tallennetun tiedon tutkimiseen.

Järjestelmän päätavoitteina on tutkimustulosten tallentamisen keskittäminen, tutkijoiden projektien vaiheiden yhdenmukaistaminen, inhimillisten virheiden vähentäminen, omien ja muiden käyttäjien tutkimustulosten nopeampi paikallistaminen sekä tietojen varmuuskopiointi. Järjestelmän on pystyttävä luomaan jokaista projektia varten tarvittavat tallennusmahdollisuudet.

Yhtenä suurimpana ongelmana oli vanhojen tutkimustulosten löytäminen. Jos yliopistosta poistui joku tutkija ja hänen tutkimuksiaan tarvittiin esimerkiksi vuoden tai kahden päästä, oli oikean tiedoston löytäminen aikaisemmin lähes mahdotonta, vaikka tiedostot olisivatkin löytyneet, saattoi oikean tiedon löytäminen itse tiedostosta olla todellisen vaivan takana. Tämän takia järjestelmään rakennettiin helppokäyttöinen hakutoiminto, jonka avulla vanhat prosessit löytyvät helposti. Myös tutkimustulosten varmuuskopiointi oli aikaisemmin jokaisen tutkijan omalla vastuulla ja järjestelmän avulla haluttiin huolehtia tiedon varmuuskopioinnista automaattisesti.

Työssä esitellään työn tilaaja sekä käydään lävitse työvaiheet ja järjestelmän yleiskuvaus. Lisäksi kerrotaan järjestelmässä käytetyistä tekniikoista ja ohjelmistoista sekä järjestelmään rakennetusta tietokannasta ja käyttöliittymästä. Lopussa perehdytään vielä virheidenhallintaan sekä järjestelmän testaukseen.

2 ITÄ-SUOMEN YLIOPISTO

Kuopion ja Joensuun yliopistojen yhdistyessä vuonna 2010 Itä-Suomen yliopistoksi tästä syntyi yksi Suomen suurimmista tiedeyliopistoista. Tällä hetkellä sillä on kampukset Kuopion lisäksi Joensuussa ja Savonlinnassa. Opinnäytetyö tehtiin Itä-Suomen yliopiston Kuopion yksikössä toimivalle neurologian laitokselle. Yhteyshenkilönä yliopiston puolella oli laitoksen tutkimusjohtaja Mikko Hiltunen. Opinnäytetyö tilattiin AlzPoint-nimistä hanketta varten sekä muita tulevia projekteja silmällä pitäen.

Yliopistossa opiskelee 15 000 opiskelijaa, jonka lisäksi se työllistää noin 2 800 ihmistä. Yliopisto tarjoaa opetusta yli 100 pääaineessa. Itä-Suomen yliopiston Kuopion laitos on yksi johtavista Alzheimer-taudin tutkimuslaitoksista. Neurologin tutkimuskohteenä ovat muun muassa Alzheimerin tauti ja sen riskitekijät. Sieltä on valmistunut lähes 100 väitöskirjaa ja projekteissa työskentelee 20 väitöstyöntekijää. (Itä-Suomen yliopisto 2013.)

3 TYÖVAIHEET

Opinnäytetyön käytännön osuus rakentuu käyttöliittymästä sekä siihen liitetystä tietokannasta. Projekti aloitettiin keräämällä asiakkaalta tarvittavat tiedot sovelluksen suunnittelua varten aloituspalaverin sekä tarkemman perehtymiskierroksen avulla. Näiden tietojen avulla voitiin luoda toiminnallinen määrittely, jonka pohjalta voitiin aloittaa tietokannan suunnittelu.

3.1 Tietokannan suunnittelu ja toteutus

Tietokannasta tuli luoda relaatiotietokanta, jonka avulla tiedon eheys pystytään takaamaan paremmin. Tietokannan tuli myös olla helposti muokattavasti mahdollisia tulevia lisäominaisuuksia varten. Tietokannan suunnittelun jälkeen kanta toteutettiin Microsoft SQL Server Management Toolsien avulla. Tämän jälkeen valmis tietokanta tuotiin Microsoft Visual Studio 2010:ssa olevaan projektiin Entity Frameworkin avulla. Entity Framework on esitelty tarkemmin luvussa 4.2. Tietokantaa jouduttiin välillä muokkaamaan, mutta perusrunko pysyi koko ajan samana, joten muutokset olivat helposti toteutettavissa.

3.2 Käyttäjien todentaminen

Seuraavaksi toteutettiin käyttäjien todentaminen järjestelmään. Todentaminen olisi mahdollista toteuttaa joko Windows- tai Forms -todentamisella.

Windows -todentamisessa ei ole tarvetta luoda sovellukseen erillistä kirjautumissivustoa. Kirjautumisesta ja tunnuksista pitää huolen esimerkiksi Windows -palvelimella pyörivä Active Directory, joka tunnistaa käyttäjän ja antaa tälle tarvittavat oikeudet käyttää sovellusta. Active Directoryä voidaan verrata esimerkiksi puhelinluetteloon, johon on keskitetysti tallennettu kaikki käyttäjien kirjautumistiedot sekä käyttäjän tunnuksien oikeudet. Windows-todentamista käytetään erityisesti tietokoneissa, jotka jakavat saman toimialueen. Tämä mahdollistaa esimerkiksi sen, että käyttäjät voivat kirjautua mille tahansa toimialueen tietokoneelle, ilman että jokaiselle tietokoneelle määritellään kirjautumistunnukset käyttäjäkohtaisesti. (IIS, Windows Authentication 2013.)

Forms-todentamisessa sovellukseen luodaan oma kirjautumissivu, jonka avulla käyttäjät voivat päästä oikeilla tunnuksilla käyttämään sovellusta. Tunnuksien ylläpitämi-

nen on myös sen sovelluksen vastuulla, toisin kuin esimerkiksi Windows-todentamisessa käytetään Active Directoryä. Forms-todentamista käytetään etenkin silloin, jos järjestelmään halutaan mahdollisesti kirjautua myös toimialueen ulkopuolelta. Muun muassa tästä syystä päädyttiin Forms-todentamiseen. Toisena syynä oli mahdollisten yhteisten ryhmätunnusten luominen. Lisäksi sovelluksen käyttäjämäärä on suhteellisen pieni, jolloin tunnusten hallinnointi olisi tarpeeksi yksinkertaista oman hallintapaneelin avulla. Myös oikeuksien lisääminen ja muokkaaminen tarpeen vaatiessa olisi nopeampaa tällä tavalla. (MSDN, ASP.NET Forms Authentication 2013.)

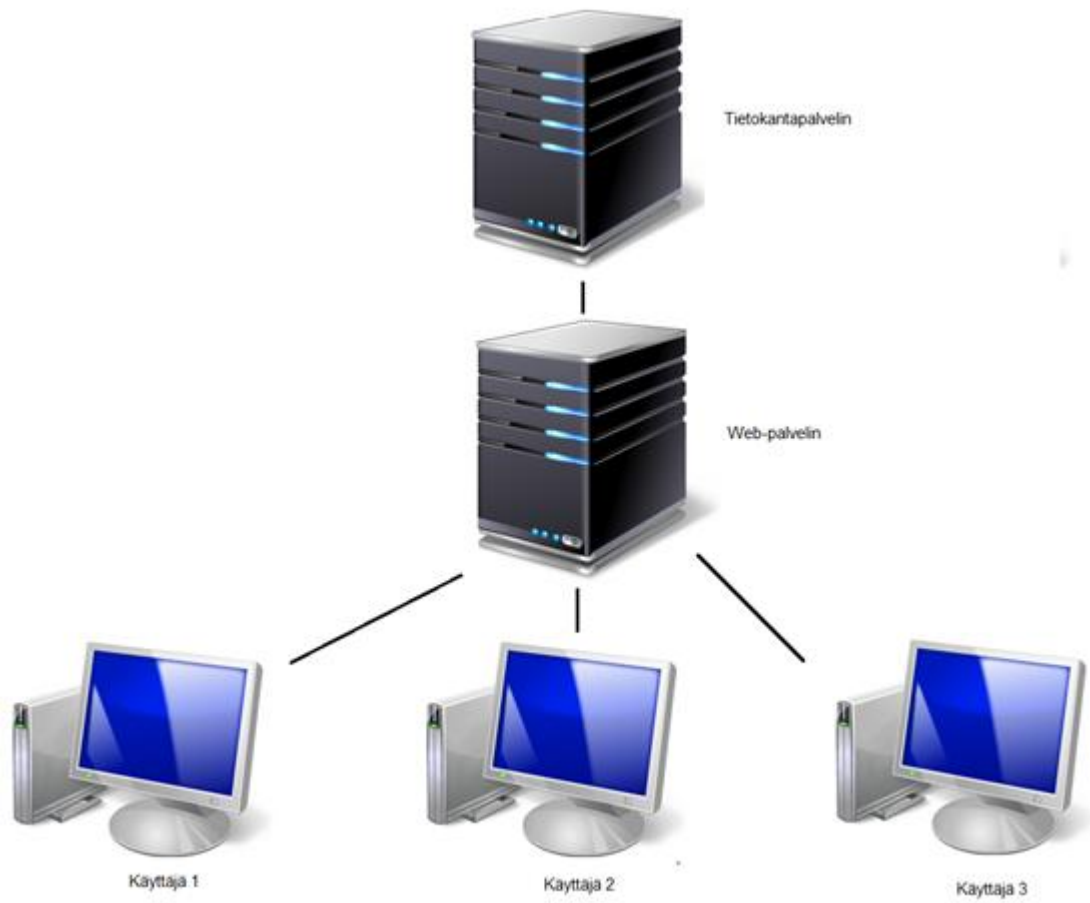
3.3 Käyttöliittymä ja sovelluslogiikka

Kun käyttäjien todentaminen ja hallinta oli luotu järjestelmään, voitiin siirtyä peruskäyttäjien käyttämän järjestelmän käyttöliittymän ja logiikan rakentamiseen. Kaikki bisneslogiikka luotiin pääasiassa erillisiin tiedostoihin, jolloin käsittelijöiden tehtäväksi jäi vain tiedon siirtäminen näkymältä toisella sekä tarvittavien mallien ja muiden näkymälle vietävien muuttujien alustaminen.

Näkymät luotiin käyttämällä Razor-näkymämoottorin tarjoamaa ASP.NET-syntaksia sekä HTML-kuvauskieltä. Ulkoasun toteuttamisessa käytettiin pääasiassa valmiina olevaa CSS-tyylitiedostoa, johon käyttöliittymän ulkoasumuutokset toteutettiin. Ohjelmassa olevat mallit luotiin pääasiassa siten, että yksi malli vastasi aina yhtä tietokannan taulua. Toki muutamia poikkeuksia toteutettiin, kuten virheilmoitusten välittämiseen tehtyä omaa mallia.

3.4 Käyttöönotto

Järjestelmä otettiin käyttöön luomalla sovellukselle oma application pool IIS:iin. IIS:ssä pyörivälle sovellukselle luotiin myös oma identiteetti. Järjestelmä hajautettiin siten, että sovellus laitettiin pyörimään omalle web-palvelimelle sekä tietokanta toiselle palvelimelle. Näille kahdelle luotiin oma käyttäjätunnus, jonka avulla järjestelmää voidaan käyttää. Seuraavassa kuvassa (kuva 1) on esitelty järjestelmän arkkitehtuuri.

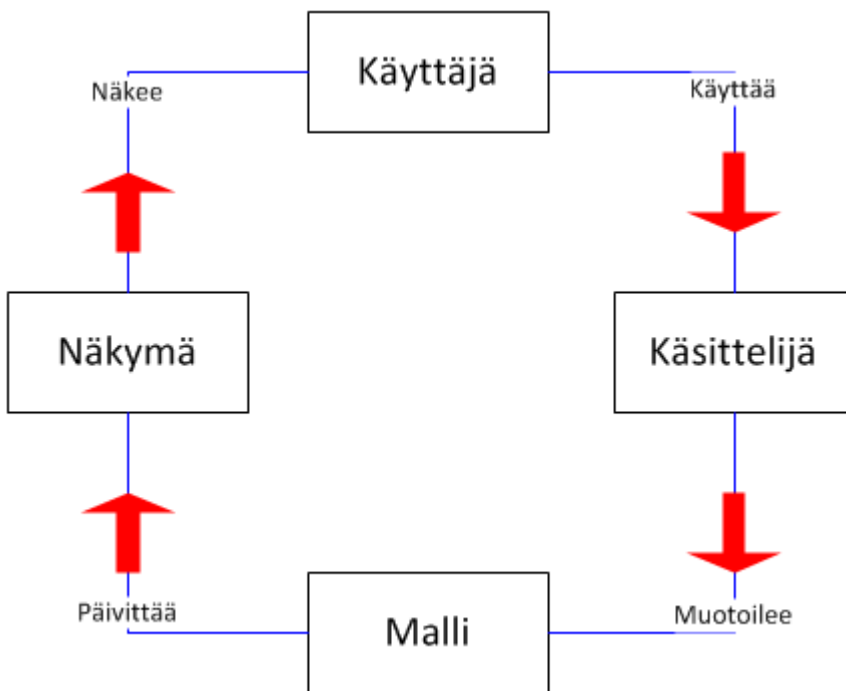


KUVA 1. Järjestelmän arkkitehtuuri

4 TEKNIIKAT

4.1 MVC

MVC eli Model-View-Controller-suunnittelumalli koostuu kolmesta osasta: Model eli malli, View eli näkymä ja Controller eli käsittelijä. Alapuolella oleva kuvio (kuvio 1) havainnollistaa MVC:n peruseriaatteen hyvin. Esimerkiksi kun käyttäjä painaa verkkosivulla olevaa painiketta, ohjelman käsittelijä ottaa käskyn kiinni ja tekee pyydetyt toiminnot, minkä jälkeen järjestelmä mallin avulla toimittaa saadut tiedot näkymälle, jonka käyttäjä näkee.



KUVIO 1. MVC:n peruseriaate

Seuraavissa alaluvuissa kerrotaan tarkemmin MVC-mallin osista, sillä suuri osa järjestelmästä perustuu juuri MVC-mallin käyttöön.

4.1.1 Malli

Malleja käytetään tiedon siirtämiseen paikasta toiseen, pääasiassa näkymien ja käsittelijöiden välillä. Yksi malli voi olla esimerkiksi yksi tietokantataulu. Ei voida kuitenkaan puhua poikkeuksesta, jos mallia ei löydy tietokannasta, sillä malleja voidaan räätälöidä omien tarpeiden mukaan. Esimerkiksi opinnäytetyön järjestelmässä hyö-

dynnettiin malleja myös ilmoitusviestien säilömisessä ja siirtämisessä käsittelijöiden ja näkymien välillä. Mallien jäsenmuuttujat toteutetaan pääasiassa Get- ja Set - metodien avulla, jolloin tiedon tallentaminen malliluokkaan onnistuu helposti. Seuraavassa kuvassa (kuva 2) on esitelty yksi perusmalli, joka vastaa tietokannasta löytyvää Format-taulua.

```
public class FormatModel
{
    public int ID { get; set; }
    public int PitCount { get; set; }
    public int PitSize { get; set; }
    public int PitID { get; set; }
}
```

KUVA 2. Esimerkki FormatModel-mallista

.Net mahdollistaa myös jäsenmuuttujille annettavan määritteitä, kuten että muuttuja on pakollinen tai pisimmän mahdollisen kirjainlukumäärän.

4.1.2 Näkymä

Näkymän tarkoituksena on luoda loppukäyttäjälle Internet-selaimessa näkyvä sivu. Näkymä luodaan tässä opinnäytetyössä HTML-syntaksin, CSS-tyylitiedoston sekä Razor-näkymämoottorin avulla. Web-palvelin osaa tulkita ASP.NET-syntaksin ja kääntää sen HTML-kuvauskieleksi, jota jokainen selain ymmärtää.

Näkymän ei ole tarkoitus sisältää juuri ollenkaan tiedonkäsittelyä, vaan tarvittavat tiedot tuodaan mallien avulla näkymälle ja haluttu tieto vain näytetään käyttäjälle näkymään syötetyllä tavalla. Tiedon muokkaaminen näkymässä ei ole kiellettyä, mutta on suositeltavaa tehdä se sille varatuissa paikoissa.

4.1.3 Käsittelijä

Käsittelijä ohjaa käyttäjän siirtymistä sivustolla. Sen tarkoituksena on välittää tietoa näkymältä toiselle. Käsittelijöiden ei tulisi sisältää juuri ollenkaan tiedonkäsittelyä vaan tarkoituksena on alustaa muuttujia ja malleja seuraavaa näkymää varten. Varsinainen bisneslogiikka on tehty omiin tiedostoihinsa. Tietoja tallennetaan näkymiä varten tämän opinnäytetyön tapauksessa ViewBag:in, istuntoihin ja ennen kaikkea

malleihin. ViewBag:in avulla tieto voidaan kätevästi viedä käsittelijältä näkymälle eikä sitä tarvitse erikseen poistaa vaan sovellus huolehti itse siitä. Istunnot ajavat muuten saman asian, mutta ne häviävät vasta kun ne poistetaan tai niille määritelty aika kuluu loppuun. Käsittelijän tehtävänä voisi olla esimerkiksi viedä sille tuotu arvo tietylle bisnesluokan metodille, joka tekee annetulla arvolla tietokantahaun ja palauttaa haun tuloksen käsittelijälle, joka toimittaa sen eteenpäin näkymälle ViewBag:in avulla. (ASP.NET MVC Overview 2013.)

4.2 ASP.NET MVC 3

ASP.NET MVC 3 on Microsoftin tarjoama verkkokehitysohjelmistokehys, joka yhdistää tehokkuuden, siisteyden ja MVC-suunnittelumallin arkkitehtuurin sekä muut parhaat ASP.NET:in ominaisuudet. Se on täydellinen vaihtoehto perinteiselle ASP.NET Web Forms:lle ja se tarjoaa huomattavia etuja useimpiin verkkokehitysprojekteihin. (Freeman, Sanderson 2011.)

Sovellus päätettiin alkaa rakentamaan kyseiselle ohjelmistokehitykselle, sillä ASP.NET MVC 4 oli vielä beta-vaiheessa, kun sovellusta alettiin toteuttamaan. Tällä haluttiin välttyä mahdollisilta virheiltä ja ongelmilta, joita beta-versiot yleensä sisältävät. Beta-versiossa olevasta ohjelmasta käyttäjät pyrkivät vielä paikantamaan sinne jääneitä virheitä ja testaamaan ohjelman käytettävyyttä.

Yhtenä suurimpana etuna MVC 2 -malliin oli MVC 3:een tuotu Razor-näkymämoottori, jonka avulla näkymien syntaksista pystytään tekemään selkeämpi, sillä tagien käyttö on huomattavasti yksinkertaisempaa kuin aikaisemmalla näkymämoottorilla. Razor sisälsi myös paremman IntelliSense-toiminnon. IntelliSense-toiminto tarjoaa listan vaihtoehtoja ohjelmoitaessa, jolloin käyttäjän ei tarvitse alkaa etsimään oikeaa termiä koodieditorin ulkopuolelta. (MSDN, Using IntelliSense 2013.)

4.3 Entity Framework

Entity Framework on Microsoftin tarjoama, avoimeen lähdekoodiin perustuva ohjelmistokehys, jonka avulla tietokannan käyttö on huomattavasti yksinkertaisempaa. Entity Framework luo objektimallin tietokannasta ja tarjoaa työkalut tiedonkäsittelyyn. Työkalujen avulla käyttäjä voi helposti hakea, muokata tai poistaa tietoa tietokannasta. Entity Frameworkin apuna voidaan käyttää esimerkiksi LINQ:a, josta lisää seuraavassa luvussa. Entity Framework tukee myös SQL:ää. (MSDN, Entity Framework 2013.)

4.4 LINQ

LINQ eli Language Integrated Query on ohjelmointimalli, joka tarjoaa täysin uudenlaisen tavan kyselyiden tekemiseen ja datan käsittelyyn Microsoftin .Net -projekteissa. LINQ pyrkii eroon tavallisen sulautetun SQL:n mukanaan tuomista rajoitteista ja samalla kasvattamaan ohjelmointityön tuottavuutta, tarjoamalla lyhyemmän, mielekkään ja ilmaisuvoimaisemman syntaksin datan käsittelyyn. SQL eli Structured Query Language on tietokantakieli, jolla voidaan tehdä muutoksia tietokantaan tai hakea sieltä tietoa. Sulautetussa SQL:ssä voidaan muita ohjelmointikieliä yhdistää SQL-kyselyihin. LINQ-syntaksi on helposti opittavissa sen yksinkertaisuuden takia, mutta se toimii vain .Net ympäristössä toisin kuin SQL. LINQ mielletään korkeamman tason kieleksi kuin SQL. Se on myös tehokkaampi ja syntaksi on lyhempi kuin SQL:ssä etenkin pidemmissä ja monimutkaisemmissa hauissa. (Halonen 2008.)

4.5 JQuery

JQuery on JavaScript-kirjasto, joka sisältää yleisimmät JavaScript-komennot. JavaScriptiä suoritetaan käyttäjän selaimessa ja sillä voidaan muokata HTML-sivua ja sen rakennetta. Opinnäytetyössä jQuery:ä käytettiin muun muassa käyttäjän syötteiden validointiin ajon aikana. (jQuery 2013.)

5 OHJELMISTOT

5.1 Microsoft Visual Studio 2010

Microsoft Visual Studio 2010 on Microsoftin tarjoama maksullinen kehitystyökalu. Microsoft on julkaissut myös ilmaiset, mutta hieman kevyemmät versiot Visual Studio Express ja Visual Web Developer Express. Microsoft Visual Studio on tällä hetkellä ehdottomasti yksi parhaista kehitystyökaluista markkinoilla .Net-ympäristöä varten sen monipuolisuuden ja eheyden johdosta.

Ohjelma sisältää lähes kaiken tarvittavan ASP.NET MVC 3:n tekemiseen. Se sisältäisi myös työkalut tietokannan luomiseen, mutta Microsoft SQL Server Management Tools -työkalun avulla tietokannan suunnittelu ja toteuttaminen on erittäin helppoa. Lisää Microsoft SQL Server Management Toolsin käytöstä luvussa 5.3.

Microsoft Visual Studio tarjoaa käyttäjälle mahdollisuuden luoda ASP.NET MVC 3 - pohjan, jos samalle koneelle on asennettuna ASP.NET MVC 3. Työkalusta löytyy myös graafiset käyttöliittymät tietokantayhteyden luomiseen Entity Frameworkin avulla, jolloin ohjelma luo myös objektimallin tietokannasta käyttäjälle.

Ohjelma sisältää myös virheilmoitusten käsittelyn sekä ennen kaikkea debuggaus-toiminnon, jonka avulla käyttäjän on ongelmatilanteissa yksinkertaista käydä koodiaan lävitse. Microsoft Visual Studio sisältää myös NuGet-paketinhallinnan, jonka avulla esimerkiksi jQuery-tiedostot voidaan päivittää uusimpiin mahdollisiin muuttaman napin painalluksen avulla. Myös virheiden tallentamisesta huolehtiva ELMAH-kirjasto lisättiin NuGet:n avulla.

5.2 Microsoft SQL Server Express 2008 R2

Microsoft SQL Server Express 2008 R2 on Microsoftin tarjoama tietokantaohjelmisto, johon voidaan luoda tietokantoja tiedon tallentamista varten. Käytetty versio on ilmainen, mutta se sopii erinomaisesti järjestelmien tietokantojen säilytyspaikaksi sovel-luksen kehitysvaiheessa. Express-versiolle tehty tietokanta on siirrettävissä muille Microsoft SQL Server -versioille. Ilmainen versio rajoittaa tietokannan koon korkeintaan 10 GB:iin ja kanta ei voi käyttää kuin yhtä palvelimen prosessoria ja 1 GB:n si-säistä muistia. Muun muassa tämän takia tätä MSSQL-versiota ei ole tarkoitettu pit-käaikaiseen käyttöön, kuten yrityksen tai muun liiketoiminnan tietokantojen hallinnoin-

tiin. Tietokannat luodaan MSSQL:ään tässä opinnäytetyössä Microsoft SQL Server Management Toolsin avulla. Lisää tästä työkalusta seuraavassa luvussa.

5.3 Microsoft SQL Server Management Studio

Microsoft SQL Server Management Studion eli SSMS:n avulla voidaan luoda tietokanta helposti ja vaivattomasti Microsoft SQL Serverille. Se tarjoaa laajan valikoiman toimintoja tietokannan hallintaan sekä melko yksinkertaisen graafisen käyttöliittymän. Ensimmäinen Management Studio julkaistiin SQL Server 2005:n yhteydessä, ja se on saanut vain muutamia parannuksia SQL Studio 2008:n versioon. (Vieira 2009.)

Ohjelma sisältää kaiken tarvittavan tämän opinnäytetyön tietokannan luomiseen. SSMS:n avulla voitiin suunnitella sekä rakentaa tietokanta sen työkalusta löytyvällä graafisen käyttöliittymän avulla. Graafinen käyttöliittymä on tehty käyttäjälle niin yksinkertaiseksi, että käyttäjän on helppo rakentaa tietokanta tämän sovelluksen avulla.

Ohjelman avulla voidaan tietokannasta luoda kaaviokuva, jonka avulla on helppo saada yleiskatsaus tietokannasta. Kaaviokuvaan voidaan myös tehdä muutoksia, jotka tallentuvat suoraan tietokantaan. SSMS:n avulla voidaan myös luoda relaatiot taulujen välille, jolloin tietokannasta saadaan tehtyä niin sanottu relaatiokanta. Relatiokannan avulla pystytään varmistamaan tiedon eheys huomattavasti paremmin kuin tietokannassa, jossa relaatioita ei käytetä, sillä tietojen pyritään aina olevan yhteydessä toisiinsa.

5.4 IIS - Internet Information Services

Järjestelmässä käytettiin web-palvelimena Windows-pohjaisille palvelimille tarkoitettua IIS 7:ää. IIS:n tarkoituksena, kuten minkä tahansa muun web-palvelimen, on pyörittää web-sovellusta, joka keskustelee tietokannan sekä käyttäjän selaimen kanssa. IIS-palvelimella sijaitseva ohjelma muuntaa tiedon Internet-selaimen ymmärtämään HTML-muotoon. IIS:lle on myös luotu oma application pool, jolle on annettu tarvittavat oikeudet, joiden avulla ohjelma pystyy erimerkiksi kirjoittamaan palvelimelle. Application pool on yksittäinen osoite tai joukko osoitteita, joita palvelee työntekijäprosessit tai ne on asetettu työntekijäprosesseiksi. Se asettaa rajat ohjelmille, jotka se sisältää, mikä tarkoittaa sitä, että mikä tahansa sovellus valitun application poolin ulkopuolelta ei voi vaikuttaa valitun application poolin sovelluksiin. (TechNet, Managing Application Pools in IIS 7 2013.)

6 TIETOKANTA

6.1 Yleistä tietoa

Tietokanta toteutettiin Microsoft SQL Server 2008 R2 Expressille. Tietokanta siirretään käyttöönoton yhteydessä Microsoft SQL Server 2008 R2 Enterpriselle. Tietokanta olisi ollut myös mahdollista toteuttaa suoraan MSSQL 2008 R2 Enterpriselle, mutta MSSQL 2008 R2 Express sisälsi kaikki tarvittavat toiminnot tietokannan toteuttamiseen kehitysvaiheessa.

6.2 Suunnittelu

Tietokannan suunnittelu lähti liikkeelle tiedon keruusta. Alkupalaverissa käytiin lävitse sovelluksen tarkoitus pääosin, minkä jälkeen yliopisto tarjosi vielä tarkemman perehtymiskierroksen omissa tiloissaan. Kierroksen aikana esiteltiin, mitä työntekijöiden työtehtävät tarkemmin ovat ja kuinka tämän sovelluksen avulla voitaisiin toistuvia työvaiheita vähentää. Kierroksen avulla voitiin myös tarkemmin määrittää sovelluksen toiminnallisuutta.

Tietokantaohjelmisto vaihtoehtoina olivat Microsoft SQL Server 2008 Express R2 sekä MySQL. Tietokantaohjelmaksi päätettiin MSSQL, kun lopullinen palvelinratkaisu saatiin selville. Tietokannasta oli heti alusta lähtien tarkoitus tehdä relaatiokanta, jolla varmistettaisiin tiedon eheys. Tietokannan kuvaa aloitettiin hahmottamaan käsin piirtämällä, minkä jälkeen alustava suunnitelma toteutettiin Microsoft SQL Server Management Studion avulla. Tietokantaa on kuitenkin päivitetty koko sovelluksen kehityksen ajan, mutta perusrunko pysyi koko ajan samana, joten muutokset olivat helposti ja yksinkertaisesti toteutettavissa.

6.3 Rakenne

Tietokanta koostuu 31 taulusta. Tauluista 12 on Microsoft Visual Studio luomia tauluja, joita tarvitaan järjestelmässä olevaan Forms-todentamiseen. Tietokannan keskeisin taulu on KMSPProcess, johon kaikki taulut, Forms-todentamisen tauluja lukuun ottamatta, ovat yhteyksissä. Seuraavissa alaluvuissa on esitelty tietokannan keskeisimmät taulut sekä niiden tarkoitus. Kuva yleisen järjestelmän tietokannasta on liitteessä 1 ja käyttäjien todentamisen tietokanta liitteessä 2.

6.3.1 KMSProcess

Koko järjestelmä nitoutuu KMSProcess-taulun ympärille. Tämä taulu sisältää kaikki keskeiset tiedot käyttäjän prosesseista. Jokainen rivi sisältää muun muassa prosessin nimen, sen omistajan sekä analyysien id:t. Jos prosessi poistetaan tästä taulusta, kaikki tiedot, jotka on liitetty kyseiseen prosessiin, häviävät tietokannasta sekä kaikki tiedostot, jotka on liitetty prosessiin poistetaan palvelimelta.

6.3.2 Plan

Plan-tiluun tallennetaan taulun nimen mukaisesti aina valitun prosessin suunnitelma. Suurin osa Plan-tilun tiedoista on vapaaehtoisia, mutta Plan-tilu on erittäin keskeinen prosessissa, sillä se antaa käyttäjälle yleiskatsauksen valitusta prosessista. Pakollisena tietona on muun muassa prosessin id, joka määräytyy KMSProcess-tilusta.

6.3.3 Format

Format-tilu sisältää prosessiin kuuluvien formaattien tiedot. Käyttäjä voi valita sovelluksessa erikokoisia formaatteja sekä niiden lukumäärän. Nämä tiedot ovat tallessa Plan-tilussa. Format-tilu sisältää tiedon prosessin jokaisesta formaatista ja niissä olevista kaivoista, joihin käyttäjät syöttävät näytteitään. Format-tilu on myös kytköksissä FormatNote-tiluun, josta löytyy formaateille syötetyt mahdolliset muistiinpanot.

6.3.4 Tiedosto-tilut

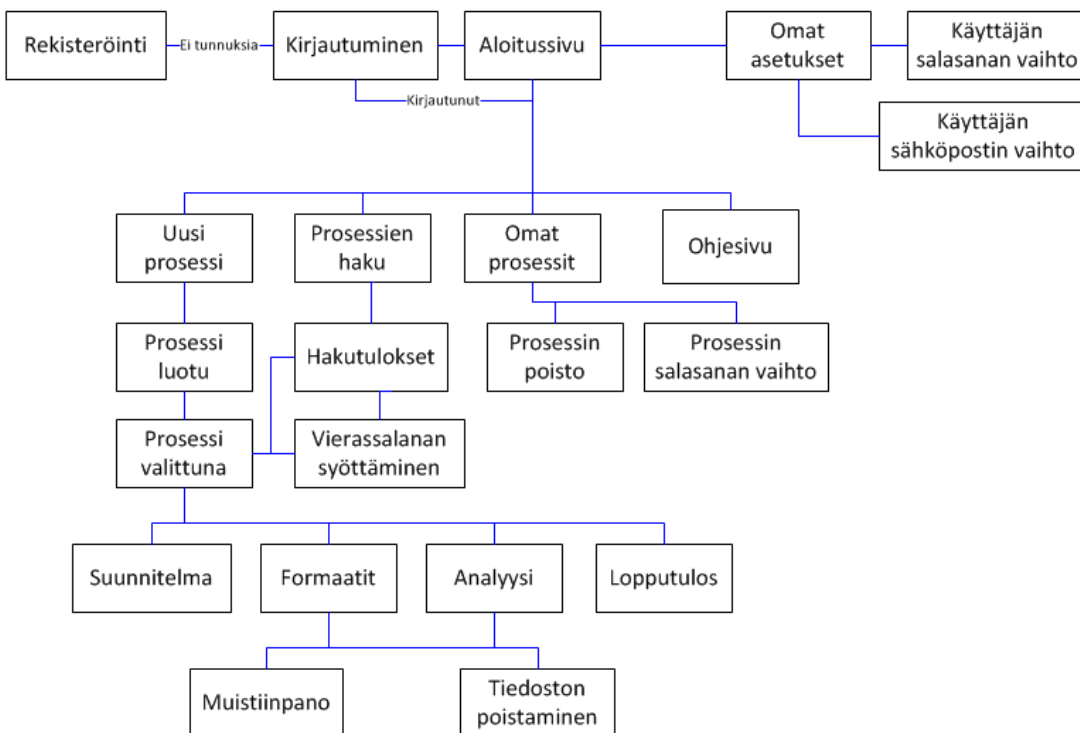
Tietokanta sisältää useampia tiedoston tallentamiseen tarkoitettuja tiluja. Itse tiedostoja ei koskaan tallenneta tietokantaan, vaan ne sijoitetaan palvelimella olevaan kansioon. Tietokantaan tallennetaan vain tieto tiedoston sijainnista eli sen polku. Näiden lisäksi tallennetaan muun muassa tiedoston nimi, prosessin id eli mihin prosessiin tiedosto on sidottu sekä tiedoston mime-type, jolloin kun tiedosto ladataan järjestelmään, selain osaa ehdottaa käyttäjälle sopivaa ohjelmaa tiedoston aukaisua varten.

7 KÄYTTÖLIITTYMÄ

Käyttöliittymä toteutettiin käyttämällä Razor-näkymämoottorin ASP.NET-syntaksia, HTML-kuvauskieltä sekä CSS-tyylitiedostoa. Näkymät toteutettiin siten, että näihin tiedostoihin kirjoitettiin ainoastaan viittaukset CSS-tyylitiedostoon, joten sovelluksen ulkoasun muuttaminen vaatii vain CSS-tyylitiedoston muokkaamista.

7.1 Peruskäyttäjän näkymät

Käyttöliittymä koostuu 34:stä eri sivusta, joista peruskäyttäjän nähtävissä on 22 ja loput 12 on varattu vain pääkäyttäjälle eli ylläpitäjälle. Seuraavassa kuviossa (kuvio 2) on esitelty sivukartta peruskäyttäjän kannalta ja seuraavissa alaluvuissa käydään jokainen sivu tarkemmin lävitse. Kaikista olennaisimmista sivuista on liitetty käyttöliittymäkuva.



KUVIO 2. Sivukartta peruskäyttäjän näkökulmasta

7.1.1 Aloitussivu

Aloitussivu on ensimmäinen sivu, jolle käyttäjä saapuu sovelluksen käynnistyessä. Tällä näkymällä on yleinen tekstiosio, jolla pyritään käyttäjälle kertomaan mistä on kyse ja antamaan hänelle mahdollisimman paljon tietoa sovelluksen tarkoituksesta. Ylläpidon on mahdollista muokata aloitussivun tekstiä hallintapaneelista löytyvän ominaisuuden avulla. Tästä näkymästä ilman kirjautumista käyttäjän on mahdollista jatkaa vain kirjautumis- tai ohjesivulle. Aloitussivulle on aina mahdollista palata sovelluksen vasemmassa yläkulmassa olevaa sovelluksen nimeä painamalla.

7.1.2 Kirjautuminen

Jokaisen sovellusta käyttävän on kirjaututtava validoiduilla tunnuksilla. Kirjautuminen tapahtuu yksinkertaisesti käyttäjätunnuksen sekä salasanan avulla. Käyttäjän ei tarvitse huolehtia tunnustensa turvallisuudesta, sillä tiedot salataan ennen kuin ne lähetetään palvelimelle. Käyttäjän syöttäessä väärät kirjautumistiedot sovellus ilmoittaa siitä käyttäjälle ja pyytää tarkastamaan syötteet. Sivulta löytyy myös Muista minut - valintalaatikko, jonka avulla käyttäjän ei tarvitse aina uudelleen kirjautua sovellukseen, kun selain käynnistetään uudelleen. Edellinen toiminto on selainkohtainen, sillä jotkut selaimet voivat muistaa tunnukset ilman valintaakin. Tältä sivulta löytyy myös linkki rekisteröintiin jos käyttäjältä puuttuu tunnukset sovellukseen.

7.1.3 Rekisteröinti

Rekisteröinnissä käyttäjän on annettava käyttäjätunnus, salasana kahteen kertaan sekä sähköposti. Samanlaista käyttäjätunnusta ei saa olla olemassa ja salasanan on oltava vähintään 6 merkkiä pitkä. Salasana kysytään kahteen kertaan, koska tällä halutaan minimoida käyttäjän inhimillisen syöttövirheen mahdollisuus. Sähköpostin on oltava normaalissa sähköpostin muodossa. Esimerkiksi erki.esimerkki@gmail.com on hyväksyttävä sähköpostiosoite. Virheellisen syötön satuesssa sovellus ilmoittaa käyttäjälle epäonnistuneesta rekisteröinnistä ja siitä, mikä meni vikaan.

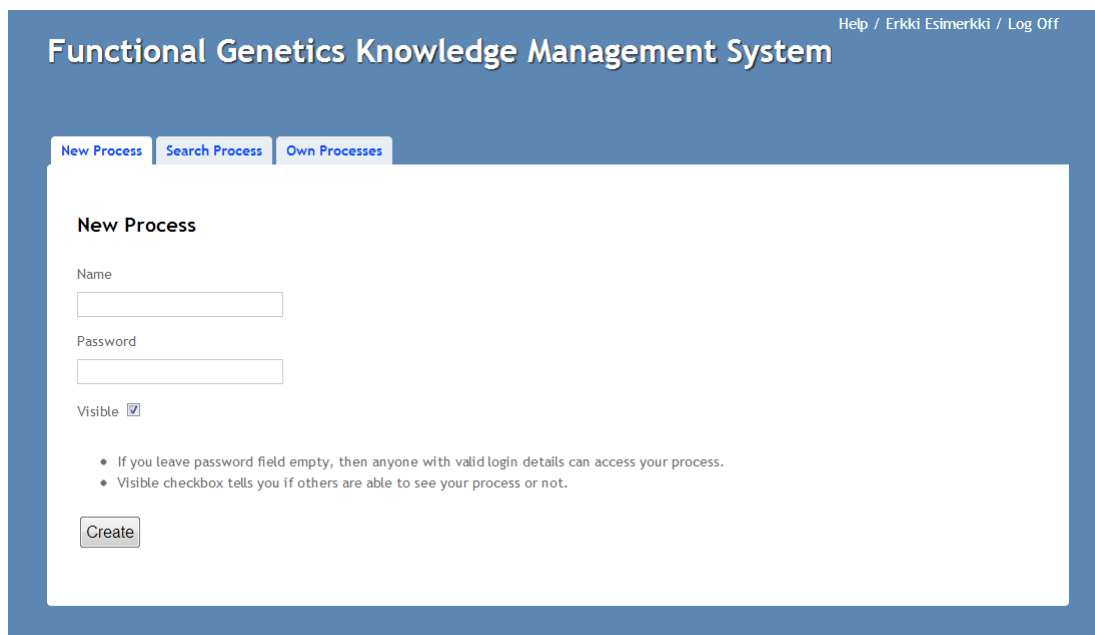
7.1.4 Omien tietojen muokkaaminen

Omien tietojen muokkaaminen tapahtuu sovelluksen oikeasta yläkulmasta omaa käyttäjätunnusta painamalla. Sivulta löytyvät linkit, joiden avulla käyttäjä voi vaihtaa oman henkilökohtaisen salasansansa tai sähköpostinsa. Molemmat linkit avaavat

oman näkymänsä, joissa ovat tarvittavat tekstilaatikot valitun tiedon muokkaamiselle. Salasanan vaihdossa on syötettävä vanha salasana, sekä uusi salasana kahteen kertaan. Sähköpostin vaihdossa käyttäjälle ilmoitetaan nykyinen sähköposti. Molemmissa tapauksissa pätevät samat säännöt kuin rekisteröintivaiheessa ja virheen sattuessa sovellus ilmoittaa siitä käyttäjälle samalla tavalla.

7.1.5 Uusi prosessi

Tältä sivulta käyttäjän on mahdollista luoda uusi henkilökohtainen prosessi. Seuraavassa kuvassa (kuva 3) on esimerkki kyseisestä sivusta. Prosessin luonnissa käyttäjän on syötettävä prosessin nimi sekä valittava onko prosessi muiden nähtävissä sivulta löytyvän Näkyvyys-valintaruuden avulla. Oletuksena valintaruutu on valittuna, jolloin prosessi näkyy kaikille käyttäjille hakutuloksissa. Jos valinta poistetaan, prosessi ei näy hakutuloksissa. Sivulta löytyy myös salasanan asettaminen prosessille, mutta tämä ominaisuus on valinnainen. Jos käyttäjä asettaa prosessille salasanan on muiden käyttäjien annettava syötetty salasana prosessia valitessa. Prosessin luonnissa prosessin nimelle tai salasanalle ei ole mitään ehtoja. Näitä ominaisuuksia voi muokata myöhemmin Omat prosessit -sivulla, prosessin nimeä lukuun ottamatta. Prosessin luonnin jälkeen käyttäjä ohjataan sivulle, jossa kerrotaan onnistuneen prosessin luonnista.



Functional Genetics Knowledge Management System Help / Erkki Esimerkki / Log Off

[New Process](#) [Search Process](#) [Own Processes](#)

New Process

Name

Password

Visible

- If you leave password field empty, then anyone with valid login details can access your process.
- Visible checkbox tells you if others are able to see your process or not.

KUVA 3. Uusi prosessi

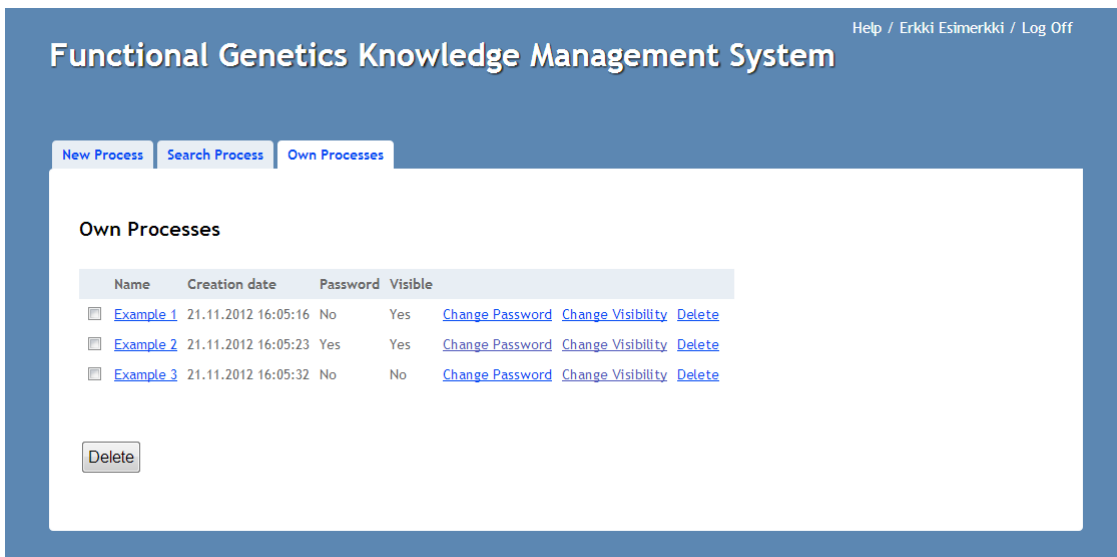
7.1.6 Etsi prosessi

Käyttäjän on mahdollista etsiä muiden käyttäjien luomia prosesseja. Haku tapahtuu syöttämällä halutut hakuehdot sivulta löytyviin tekstilaatikoihin. Käyttäjä voi etsiä haluttua prosessia prosessin nimen, omistajan tai luontipäivämäärän avulla. Kaikki kolme hakuehtoa ovat vapaaehtoisia, eikä kirjankoolla ole merkitystä. Käyttäjän jättäessä kaikki hakuehdot tyhjiksi sovellus näyttää käyttäjälle kaikki prosessit. Jos yksikään prosessi ei täytä annettuja hakuehtoja sovellus tulostaa käyttäjälle ilmoituksen tästä ja pysyy samalla sivulla.

Hakuehtojen täytyessä sovellus siirtää käyttäjän seuraavalle sivulle, johon listataan kaikki hakuehdot täyttäneet prosessit, paitsi ne joille kyseisen prosessin omistaja ei ole antanut oikeuksia näkyä hakutilanteessa. Tulostauksessa näytetään käyttäjälle prosessin nimi, omistaja, luontipäivämäärä sekä tieto siitä, vaatiiko kyseisen prosessin tutkiminen salasanan. Jos prosessia valittaessa prosessi vaatii salasanan käyttäjä ohjataan sivustolle, jossa hänen on mahdollista syöttää vaadittu salasana. Salasanan ollessa oikein prosessi valitaan aktiiviseksi ja käyttäjä ohjataan aloitussivulle. Jos salasanaa ei vaadita, käyttäjä ohjataan automaattisesti aloitussivulle ja prosessi valitaan aktiiviseksi.

7.1.7 Omat prosessit

Omissa prosesseissa käyttäjälle näytetään kaikki omat prosessit ja prosessiin liittyviä tietoja eli prosessin nimi, luontipäivämäärä, onko prosessille määrätty salasana ja onko prosessi näkyvissä muille käyttäjille hakutilanteissa. Käyttäjän on mahdollista valita prosessi aktiiviseksi painamalla prosessin nimeä. Sivulla on myös linkit prosessin salasanan vaihtoon, näkyvyyden muuttamiseen sekä prosessin poistamiseen. Käyttäjän valitessa salasanan vaihdon hänet ohjataan uudelle sivulle, jossa käyttäjältä kysytään vanha salasana sekä uusi salasana kahteen kertaan. Kaikkien tietojen ollessa oikein käyttäjä ohjataan takaisin Omat prosessit -sivulle. Jos kuitenkin yksi tai useampi tieto on väärin, sovellus ilmoittaa siitä käyttäjälle. Myös näkyvyyden vaihtaminen onnistuu halutun prosessin Vaihda näkyvyys -linkkiä painamalla, jolloin sovellus vaihtaa automaattisesti näkyvyyden vastakkaiseksi. Seuraavassa kuvassa (kuva 4) on esitelty edellä mainitun mukainen sivu.



KUVA 4. Omat prosessit

Prosessit voidaan poistaa yksitellen painamalla Poista-linkkiä halutun prosessin rivin kohdalla tai valitsemalla useampi prosessi kerrallaan prosessien edessä olevien valintaruutujen avulla. Jos käyttäjä valitsee poistaa prosesseja valintaruutujen avulla, tulee hänen painaa prosessitaulukon alareunassa olevaa Poista-painiketta. Kummassakin tapauksessa käyttäjä ohjataan seuraavalle sivulle, jossa käyttäjälle listataan kaikki valitut prosessit, jotka ollaan poistamassa. Käyttäjän on mahdollista hyväksyä poistot tai peruuttaa ne. Käyttäjän poistaessa prosessin sovellus poistaa kaiken prosessiin liittyvän tietokannasta, myös palvelimelle ladatut tiedostot.

7.1.8 Prosessin suunnitelma

Kun haluttu prosessi on valittu aktiiviseksi käyttäjän on mahdollista luoda prosessille suunnitelma. Sivulta löytyy kaksi isoa tekstikenttää, tavoite ja keruusuunnitelma. Tavoite-kenttään on pääasiassa tarkoitus syöttää prosessin tavoite ja keruusuunnitelma-kenttään suunnitelma siitä, miten saadut näytteet on tarkoitus kerätä. Käyttäjät voivat luonnollisesti lisätä kenttiin muitakin tietoja halutessaan. Kun käyttäjä on syötänyt halutut tiedot tulee hänen painaa sivun alareunassa olevaa Tallenna-painiketta. Jos painiketta ei paineta niin tehtyjä muutoksia ei tallenneta. Käyttäjän on otettava tämä erityisesti huomioon, jos hän haluaa käydä jollakin toisella sivulla kun kentät ovat keskeneräisiä. Sivunvaihto tai päivittäminen toimii myös tapana palauttaa kentät, jolloin kenttiin tulostuu alkuperäiset tekstit. Suunnitelma voidaan tulostaa selaimesta löytyvällä tulostusominaisuudella.

7.1.9 Formaattit

Formaatit-välilehdellä on tarkoitus luoda prosessissa käytettävät formaatit ja niiden kaivot. Formaateilla tarkoitetaan tässä opinnäytetyössä näytelevyjä, joissa on niin sanottuja kaivoja eli näytekuoppia, joihin eri näytteet voidaan eritellä. Aluksi käyttäjän on valittava, minkä kokoisia formaatteja halutaan eli kuinka monta kaivoa formaatti sisältää, sekä formaattien lukumäärä. Valintojen jälkeen käyttäjän tulee painaa vaihtoehtojen vieressä olevaa Vaihda-painiketta. Käyttäjän tulee huomioida, että painikkeen painaminen tyhjentää kaikki formaatit. Seuraavassa kuvassa (kuva 5) on esimerkki kyseisestä sivusta.

Process Formats

Select format size and amount

Size	Amount
<input checked="" type="radio"/> 3x2	1 ▾ <input type="button" value="Change"/> (Notice that changing format size or amount will delete all data from this page.)
<input type="radio"/> 4x3	
<input type="radio"/> 6x4	
<input type="radio"/> 8x6	
<input type="radio"/> 12x8	

Format

Format 1

Example 1_1	Example 1_2	Example 1_3
Example 1_4	Example 1_5	Example 1_6

Notes

KUVA 5. Prosessin formaatit

Kun alkuvalinnat on tehty, sovellus generoi formaatit sivulle. Jokainen kaivo sisältää nimen, joka luodaan prosessin nimestä, alaviivasta ja kokonaisluvusta. Kokonaisluku alkaa ykkösestä ja kasvaa aina yhdellä seuraavaan kaivoon siirryttäessä. Jokainen formaatti sisältää myös muistiinpano-kentän, johon käyttäjä voi halutessaan jättää tietoa formaatista tai sen kentistä. Kun käyttäjä on täyttänyt halutut kentät, tulee hänen painaa sivun alareunassa olevaa Tallenna-painiketta. Jos käyttäjä ei paina painiketta, syötetyt tiedot eivät tallennu järjestelmään. Formaattien tulostus onnistuu selaimesta löytyvällä tulostus-ominaisuudella.

Sivun alareunasta löytyy myös Luo PDF -painike, jonka avulla käyttäjälle luodaan PDF-tiedosto, joka sisältää kaikkien kaivojen nimet. Tiedosto voidaan haluttaessa avata suoraan tai tallentaa levyille. Tiedoston pääsääntöinen tarkoitus on tulostaa se tarrapaperille, josta tarrat voidaan kiinnittää niille varattuihin koeputkiin. PDF-tiedoston luomisessa on käytetty apuna avoimeen lähdekoodiin perustuvaa iTextSharp-luokkakirjastoa.

7.1.10 Prosessin analyysi

Prosessi analyysi -sivulle käyttäjän on tarkoitus ladata prosessin aikana tuotettuja tiedostoja. Sivu sisältää omat kohtansa proteiini-, RNA- sekä DNA-tiedostoille. Jokainen kohta on identtinen, mutta tarkoituksena oli jaotella erityyppiset tiedostot erikseen, jolloin tiedostojen etsiminen on yksinkertaisempaa. Tiedosto voi olla missä tahansa tiedostomuodossa. Aina tiedostoa ladattaessa on mahdollista lisätä muistiinpanoja ladattavaa tiedostoja varten. Tiedosto voidaan ladata palvelimelle jokaisesta kohdasta löytyvällä Lataa-painikkeella.

Kun tiedosto on ladattu palvelimelle, näytetään käyttäjälle taulukossa tiedoston nimi ja latauspäivämäärä. Lisäksi jokainen rivi sisältää linkin kyseisen tiedoston muistiinpanoon. Linkkiä painamalla käyttäjä siirtyy uudelle sivulle, jossa hän näkee tiedoston muistiinpanot ja voi halutessaan muokata niitä. Taulukko sisältää myös tiedoston poistomahdollisuuden. Käyttäjä voi poistaa tiedoston painamalla Poista-linkkiä halutun tiedoston rivin päästä. Kun linkkiä on painettu, ohjataan käyttäjä uudelle sivulle, jossa kysytään vielä varmistus halutun tiedoston poistoon. Käyttäjä voi vielä tässä vaiheessa perua tiedoston poiston.

7.1.11 Prosessin lopputulos

Prosessin lopputulos -sivu on lähes identtinen luvun 7.1.10 sivuun. Sivulta löytyvät kohdat kuva- ja lopputulostiedostojen lataamiselle. Tälle sivulle on tarkoitus liittää prosessin lopputuotteet. Sivun sisältää siis eri osioiden nimiä vaille täysin identtisen sivun, jossa on kaikki samat toiminnot kuin luvussa 7.1.10. Seuraavassa kuvassa (kuva 6) on esimerkki kyseisestä sivusta.

The screenshot shows a web application interface with a blue header containing navigation tabs: 'Process Plan', 'Formats', 'Analysis', and 'Summary'. The main content area is titled 'Process Summary - Images' and contains a table with the following data:

Download	Image name	Uploading date		
Download	Example 1.png	21.11.2012 16:17:16	Note	Delete
Download	Example 2.png	21.11.2012 16:17:33	Note	Delete

Below the table is a 'Valitse tiedosto' button and the text 'Ei valittua tiedostoa'. A text area for notes is provided with the instruction: 'You can leave notes about file you are going to upload here. Optional.' Below this is an 'Upload Image File' button.

The second section is titled 'Process Summary - Other files' and contains a similar table:

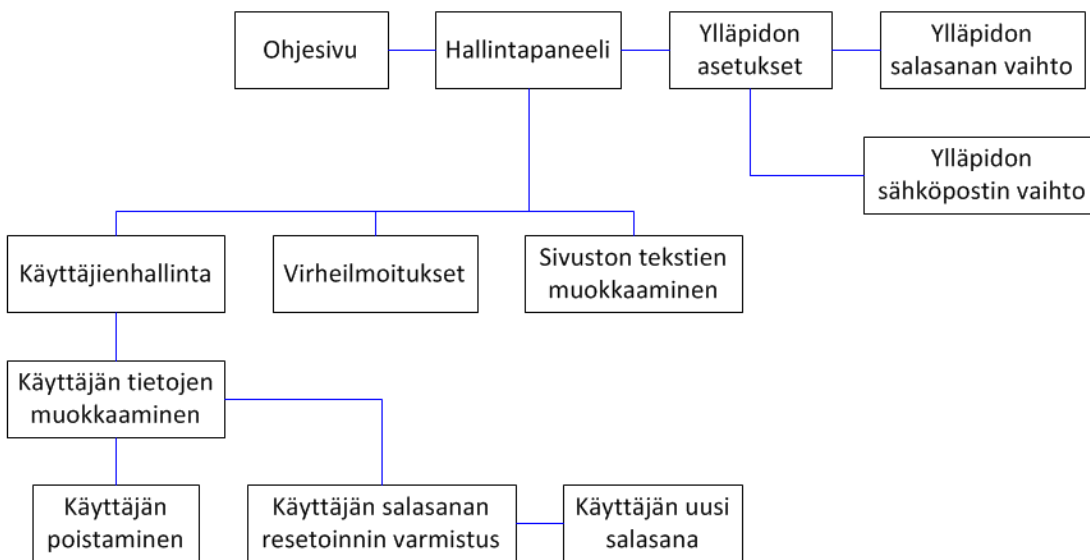
Download	File name	Uploading date		
Download	Example 3.xlsx	21.11.2012 16:17:39	Note	Delete
Download	Example 4.xlsx	21.11.2012 16:17:43	Note	Delete

Below this table is another 'Valitse tiedosto' button and the same note instruction. At the bottom is an 'Upload Summary File' button.

KUVA 6. Prosessin yhteenveto

7.2 Hallintapaneeli

Hallintapaneeliin on oikeudet vain ylläpidolla. Hallintapaneelistä löytyvät käyttäjienhallinta, virheilmoitusten seuranta sekä sivuston tekstien muokkaus -linkit. Linkkejä painamalla käyttäjälle aukeaa uusi sivu, josta kyseinen toiminto voidaan toteuttaa. Alla olevassa kuvassa (kuvio 3) on esitelty sivukartta ylläpidon näkökulmasta. Seuraavissa alaluvuissa on käyty yksityiskohtaisesti kunkin sivun toiminnot lävitse.



KUVIO 3. Sivukartta ylläpidon näkökulmasta

7.2.1 Käyttäjienhallinta

Käyttäjienhallinta-sivulle tulostuu kaikki sovelluksen tietokannassa olevat käyttäjät. Ylläpidolle näytetään jokaisesta käyttäjästä nimi, rooli, käyttäjän aktiivisuus, sähköposti sekä muokkaus- ja poistolinkit. Poista-linkki avaa uuden näkymän, jossa poistolle kysytään vielä varmistus ja muistutetaan käyttäjää, että jos poisto hyväksytään, poistetaan käyttäjä kokonaan järjestelmästä sekä kaikki hänen prosessinsa ja niihin liittyvät tiedostot. Toisena vaihtoehtona on asettaa käyttäjä ei-aktiiviseksi, jolloin hänen ei ole mahdollista päästä kirjautumaan järjestelmään mutta hänen prosessinsa säilyvät myöhempiä tutkimuksia varten.

Muokkaa-linkki avaa uuden näkymän, jossa käyttäjälle näytetään käyttäjätunnus, sähköposti, aktiivisuus-valintaruutu, rooli, salasanan palautus sekä tallenna-, että peruuta-painikkeet. Ylläpidolla ei ole mahdollisuutta vaihtaa käyttäjän käyttäjätunnus-

ta, mutta sähköpostin muokkaaminen, aktiivisuuden vaihtaminen sekä roolin valinta onnistuvat. Kaikkien edellä mainittujen muutoksien jälkeen käyttäjän on painettava Tallenna -painiketta, jotta muutokset tulevat voimaan. Palauta salasana-painike avaa uuden sivun, jossa varmistetaan kyseisen käyttäjän salasanan palautus. Jos valinta hyväksytään generoi järjestelmä valitulle käyttäjälle uuden salasanan. Ylläpito voi käyttää tätä toimintoa esimerkiksi silloin, kun käyttäjä unohtaa salasansansa. Ylläpidon tulee toimittaa uusi generoitu salasana käyttäjälle, jonka on suositeltavaa vaihtaa uusi salasana välittömästi kirjaututtuaan sovellukseen.

7.2.2 Virheiden seuranta

Tämä sivusto on luotu täysin ELMAH-kirjaston avulla. Sivulla näytetään virhekoodin tunnus, virheen tyyppi, virheviesti, käyttäjä sekä päivämäärä ja aika kätevästi taulukossa. Jokainen virheilmoitus on vielä avattavissa tarkempaan tarkasteluun. Tarkemmassa tarkastelussa on muun muassa mahdollisuus nähdä millä sivulla virhe tapahtui, jolloin sen jäljittäminen on vielä helpompaa. Sivusto lukee virheet tietokannassa olevasta taulusta. Seuraavassa kuvassa (kuva 7) esitelty osa ELMAH:n luomasta virhetaulukosta.

Code	Type	Error	Date	Time
0	Argument	Value cannot be null or empty. Parameter name: contentType Details...	19.11.2012	10:33
0	Argument	Value cannot be null or empty. Parameter name: contentType Details...	19.11.2012	10:32
0	InvalidOperationException	Sequence contains no elements Details...	19.11.2012	9:57
0	InvalidOperationException	Sequence contains no elements Details...	19.11.2012	9:56
0	InvalidOperationException	Sequence contains no elements Details...	19.11.2012	9:50
0	InvalidOperationException	Sequence contains no elements Details...	19.11.2012	9:44
0	InvalidOperationException	Sequence contains no elements Details...	19.11.2012	9:43
0	InvalidOperationException	Sequence contains no elements Details...	19.11.2012	9:42
404	Http	The controller for path '/Scripts/modernizr-2.5.3.min.js' was not found or does not implement IController. Details...	19.11.2012	9:39
404	Http	The controller for path '/Scripts/modernizr-2.5.3.min.js' was not found or does not implement IController. Details...	19.11.2012	9:39

KUVA 7. Osa virheiden seuranta taulukosta

7.2.3 Sivuston tekstien muokkaaminen

Hallintapaneeli sisältää kaksi linkkiä sivuston tekstien muokkaamiseen: etu- ja ohjesivun tekstit. Kummastakin linkistä aukeaa sama näkymä, joka sisältää ison tekstikentän sekä Tallenna, että Peruuta-painikkeet. Tekstikentän teksti riippuu siitä, kumpaa linkkiä käyttäjä painoi. Etusivun teksti -linkki avaa tekstin, joka on tarkoitettu etusivulle ja Ohje-linkki avaa tekstin, joka sisältää ohjesivun tekstin. Tekstikentän teksti sisältää

myös HTML-tageja, joiden avulla ylläpito voi muotoilla tekstin haluansa näköiseksi. Muutokset varmistetaan painamalla Tallenna-painiketta.

8 VIRHEENTARKISTUS JA -RAPORTOINTI

8.1 Virheentarkistus

Virheentarkistus toteutettiin rakentamalla järjestelmään jo ennalta ehkäiseviä ratkaisuja. Esimerkiksi jokaisen tietokantahaun jälkeen saatu tulos tarkastetaan ennen sen käyttöä, jolloin epäonnistuneesta hausta johtuvat muut tulokset eivät kaada sovellusta. Jos saatu tulos oli null eli tyhjä, niin metodin suoritus keskeytettiin välittömästi.

Virheet voitiin jakaa kahteen eri osa-alueeseen: käyttäjän tekemät virheet sekä sovellusvirheet. Käyttäjän tekemillä virheillä tarkoitetaan tilannetta, jossa käyttäjä on tehnyt jotain, jota sovelluksen ei ole tarkoitus tehdä. Esimerkiksi käyttäjä yrittää ladata sovellukseen tiedoston, mutta unohtaa valita mikä tiedosto palvelimelle ladataan. Näissä tapauksissa järjestelmä ilmoittaa käyttäjälle mikä meni vikaan ja miten asian voi korjata. Sovellusvirheiden tapauksissa käyttäjä ohjataan virhesivulle, jossa hänelle näytetään yleinen virheilmoitus ja tieto siitä, että virhe on tallennettu ja raportoitu eteenpäin sekä asia tullaan korjaamaan mahdollisimman pian.

8.2 Virheraportointi

Virheiden tallentaminen toteutettiin vapaaseen lähdekoodiin perustuvan ELMAH-kirjaston avulla, jolla voitiin tallentaa kaikki sovelluksessa tapahtuneet virheet. ELMAH tallentaa lähes kaikki virheet, joita järjestelmässä tapahtuu ajon aikana. Ohjelman asennus tapahtui Microsoft Visual Studiosta löytyvän NuGet:n avulla. ELMAH sisältää valmiit kirjastot virhetapahtumien raportoimiseen sekä virheilmoitusten näyttämiseen. Virheilmoituksia varten tietokantaan luotiin uusi taulu, johon kaikki virheilmoitukset tallentuvat. ELMAH sisältää myös kätevän käyttöliittymän virheiden seuraamiseen ja tutkimiseen. Tätä käyttöliittymää ei pääse tarkastelemaan kuin ylläpito. Virheraportti sisältää kaiken olennaisen tiedon virheestä, kuten virheilmoituksen, ajankohdan, käyttäjän jolle virhe tapahtui sekä näkymän, jossa virhe tapahtui. Virheen sattuessa se kirjataan vain ylös tietokantaan eikä sitä lähetetä eteenpäin esimerkiksi sähköpostilla, joten virheiden seuraaminen on ylläpidon vastuulle. (ELMAH 2013.)

9 TESTAUS

Sovellus testattiin pääasiassa ohjelmoinnin yhteydessä. Lähes kaikki bisneslogiikan metodit tarkastettiin Microsoft Visual Studiosta löytyvällä yksikkötestaustoiminnolla. Ohjelma luo valitusta metodista testausta varten uuden metodin. Metodin sisällä testitapaus voidaan alustaa asettamalla parametreille arvot, haluttu testitapaus suoritetaan ja saatuja tuloksia verrataan ennalta asetettuihin arvoihin.

Muiden osien, kuten käsittelijöiden ja näkymien testaaminen toteutettiin aina, kun uusi osa ohjelmasta saatiin valmiiksi. Testaaminen sisälsi erinäisten arvojen syöttämisen käyttöliittymän kautta ja näiden tietojen debuggaamisen sekä lopullisten arvojen vertailun haluttuihin. Tietojen tallentamista seurattiin myös tietokannasta, että sovellus varmasti tallensi, muokkasi tai poisti kaikki oikeat tiedot palvelimelta.

Järjestelmätestejä suoritettiin koko ajan, mutta testejä ei missään vaiheessa dokumentoitu millään tasolla. Testitapauksia ei kirjattu, mutta kaikki mahdolliset skenaariot pyrittiin käymään lävitse.

10 YHTEENVETO

Lopullinen käyttöön otettu järjestelmä tuli sisältämään Microsoft SQL Serverissä olevan tietokannan sekä ASP.NET MVC 3:lla tehdyn käyttöliittymän. Tietokannasta tuli kaikin puolin valmis, ehyt kokonaisuus, jolloin sitä on helppo lähteä jatkokehittämään tarpeen vaatiessa. Järjestelmään saatiin kaikki toiminnot, joita lähetettiin toteuttamaan ja käyttöliittymästä tuli melko käyttäjäystävällinen. Kaiken kaikkiaan järjestelmä rakennettiin asiakkaan toiveiden mukaisesti. Toimintojen oikeellisuus varmistettiin palavereilla neurologian laitoksen henkilökunnan kanssa. Opinnäytetyö oli kaiken kaikkiaan todella opettavainen ja se syvensi hyvin jo aikaisemmin opittua.

Projektin aikana havaitsin, kuinka hyvin eri Microsoftin tuotteet toimivat keskenään harmoniassa. Järjestelmän tekeminen on nopeaa ja hyvin yksinkertaista, jos kaikki osat saadaan helposti toimimaan keskenään. Itselläni on kokemusta PHP:sta kurssien sekä oman harrastuneisuuden kautta ja se on lisäksi yksi käytetyimpiä palvelinpuolen kieliä tällä hetkellä. Tämän takia vertailen hieman ASP.NET:iä PHP:seen.

ASP.NET on nopeudessaan PHP:ta huomattavasti nopeampi kieli ohjelmoida palvelinpään logiikka muutamasta syystä. Ensimmäisenä mainittakoon ASP.NET-kielen yksinkertaisuus sekä valmiiden luokkakirjastojen monipuolisuus verrattuna PHP:seen. PHP:n heikkoutena on juuri se, että lähes kaikki joudutaan tekemään alusta asti, ellei valmista vapaaseen lähdekoodiin perustavaa kirjastoa löydy jostakin. Toisena erona ovat kehitystyökalut. Siinä, missä Microsoft tarjoaa loppuun asti hiotun kehitystyökalun ASP.NET MVC 3:en ohjelmointiin, ei PHP:tä varten löydy vastaavia kehitystyökaluja.

Tietokantojen luomien on aina oma haasteensa. Miten saada tietokannasta sellainen, että se vastaa järjestelmän tarpeita ja olisi samalla mahdollisimman yksinkertainen, helposti muokattavissa, mutta tarjoaisi kuitenkin virheturvallisen ympäristön? Kaikeksi onneksi Microsoftin MSSQL Toolsien avulla on yksinkertaista suunnitella sekä toteuttaa tietokanta. Etenkin toteuttaminen on tehty todella yksinkertaiseksi, minkä johdosta sitä on myös myöhemmin helppo ja nopea muokata tarpeen niin vaatiessa.

Järjestelmän suunnitteleminen ja toteuttaminen oli kaiken kaikkiaan mielenkiintoinen kokemus ja uskon, että tämä kokemus on tulevaisuudessa todella tärkeä. Etenkin monet suuret yritykset käyttävät tällä hetkellä paljon Microsoftin tuotteita, jolloin ASP.NET:in osaaminen nousee todella tärkeään rooliin. Nykypäivänä yhä useammat

sovellukset siirtyvät Internetiin, jolloin niitä voidaan käyttää yhä useammalla laitteella milloin ja missä tahansa. Sovellusten mukana kulkeminen on nykyisin tärkeä osa liikkuvan ihmisen arkea etenkin näin älypuhelimien ja tablet-tietokoneiden aikakautena.

LÄHTEET

ASP.NET MVC Overview 2013. [verkkodokumentti]. [viitattu 27.4]. Saatavissa: <http://www.asp.net/mvc/tutorials/older-versions/overview/asp-net-mvc-overview>

Freeman, A., Sanderson S. 2011. Pro ASP.NET MVC 3 Framework. Apress.

ELMAH 2013. [verkkodokumentti]. [viitattu 8.4.2013]. Saatavissa: <https://code.google.com/p/elmah/>

Halonen, S. 2008. Microsoft LINQ. [verkkodokumentti]. [viitattu 8.4.2013]. Saatavissa: http://student.labranet.iamk.fi/~huojo/opetus/raportit/Linq_Salonen.pdf

IIS, Windows Authentication 2013. [verkkodokumentti]. [viitattu 26.4]. Saatavissa: <http://www.iis.net/configreference/system.webserver/security/authentication/windowsauthentication>

Itä-Suomen yliopisto 2013. [verkkodokumentti]. [viitattu 7.5]. Saatavissa: <http://www.uef.fi/fi/tutustu>

MSDN, ASP.NET Forms Authentication 2013. [verkkodokumentti]. [viitattu 26.4]. Saatavissa: [http://msdn.microsoft.com/en-us/library/7t6b43z4\(v=vs.100\).aspx](http://msdn.microsoft.com/en-us/library/7t6b43z4(v=vs.100).aspx)

MSDN, Entity Framework 2013. [verkkodokumentti]. [viitattu 27.4]. Saatavissa: <http://msdn.microsoft.com/en-US/data/ef>

MSDN, Using IntelliSense 2013. [verkkodokumentti]. [viitattu 26.4]. Saatavissa: [http://msdn.microsoft.com/en-us/library/hcw1s69b\(v=vs.71\).aspx](http://msdn.microsoft.com/en-us/library/hcw1s69b(v=vs.71).aspx)

JQuery 2013. [verkkodokumentti]. [viitattu 29.4]. Saatavissa: <http://jquery.com/>

TechNet, Managing Application Pools in IIS 7 2013. [verkkodokumentti]. [viitattu 28.4]. Saatavissa: [http://technet.microsoft.com/en-us/library/cc753449\(v=ws.10\).aspx](http://technet.microsoft.com/en-us/library/cc753449(v=ws.10).aspx)

Vieira, R. 2009. Professional Microsoft SQL Server 2008 Programming. Indianapolis: Wiley Publishing Inc.

Tietokannan kuva - Yleinen järjestelmä



Tietokannan kuva - Käyttäjien todentaminen

