

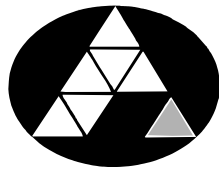
POHJOIS-KARJALAN AMMATTIKORKEAKOULU

Tietojenkäsittelyn koulutusohjelma

Mikko Mäkihonko

POHJOIS-KARJALAN AMMATTIKORKEAKOULUN  
OPINTOPSYKOLOGIN SÄHKÖISEN LOMAKKEEN  
KEHITTÄMINEN

Opinnäytetyö  
Kesäkuu 2012



POHJOIS-KARJALAN  
AMMATTIKORKEAKOULU

**OPINNÄYTETYÖ**  
**Toukokuu 2012**  
**Tietojenkäsittelyn koulutusohjelma**

Länsikatu 15, 80110 Joensuu

Tekijä  
Mikko Mäkihonko

Nimeke  
Pohjois-Karjalan Ammattikorkeakoulun opintopsykologin sähköisen lomakkeen kehittäminen  
Toimeksiantaja  
Pohjois-Karjalan ammattikorkeakoulu

**Tiivistelmä**

Opinnäytetyön toimeksiantajana toimi Pohjois-Karjalan ammattikorkeakoulu, ja opinnäytetyön tarkoituksena oli tuottaa Pohjois-Karjalan ammattikorkeakoulun opintopsykologille sähköinen lomake, mikä helpottaisi opiskelijan yhteydenottoa, sekä ajanvarausta opintopsykologille. Sähköisellä lomakkeella olisi myös tarkoitus kerätä tarvittavat esitiedot opiskelijasta ja hänen opiskeluun liittyvästä ongelmastaan ja näin helpottaa opintopsykologin työtä. Sähköinen lomake tuotettiin sekä englanniksi että suomeksi.

Opinnäytetyössä selvitetään sähköisen lomakkeen toteutusvaihtoehtoja sen ominaisuuksien ja tuotantotapojen näkökulmasta. Opinnäytetyö rakentuu lomakkeen toteutukseen ja kirjalliseen osuuteen. Kirjallinen osuus käsittelee toteutuksen eri työvaiheita sen aloituksesta lopetukseen.

Sähköisen lomakkeen toteutusmenetelmiksi valittiin Pohjois-Karjalan ammattikorkeakoulun tietohallinnon suosittelemat menetelmät ylläpidollisista sekä käyttöön otettavien syiden vuoksi. Tuotantoympäristönä oli Visual Studio 2010, ohjelmointikielenä C# ja sähköisen lomakkeen tietojen lähetykseen käytettiin System.Net.Mail-nimiavaruuskirjastoa.

Kieli  
suomi

Sivuja 24  
Liitteet  
Liitesivumäärä

Asiasanat  
verkko-ohjelmointi, Visual Studio, C#-ohjelmointi, ohjelmistotuotanto



NORTH KARELIA  
UNIVERSITY OF APPLIED SCIENCES

**THESIS**  
**May 2012**  
**Degree Programme in Business Information  
Information Technology**

Länsikatu 15  
Fin 80110 Joensuu  
FINLAND

Author  
Mikko Mäkihonko

Title  
Developing an Electronic Form for the Study Counselling Psychologist of North Karelia University of Applied Sciences

Commissioned by  
North Karelia University of Applied Sciences

Abstract

This study is aimed at developing an electronic form for the study counseling psychologist to help the students at North Karelia University of Applied Sciences to contact the psychologist and make an appointment. The study was made for North Karelia University of Applied Sciences. The form was also meant for gathering the necessary information about the student and the study related problem he/she had and thus help the study counseling psychologist's work. The electronic form was implemented both in Finnish and English.

The thesis examines different possibilities to design and implement an electronic form from the perspective of features as well as methods of the work. The study consists of realization of the form and of its written section. My study consists of realization of different stages of the study during the project.

For this thesis was chosen to use Visual Studio 2010 for production environment because, it is well suited for electronic form. As programming language was used C# and for sending the information of the electronic form was used System.Net.Mail namespace. The wishes of North Karelia University of Applied Sciences Technology Services influenced for the used techniques and manners of manufacture.

Language  
Finnish

Pages 24  
Appendices  
Pages of Appendices

Keywords  
web-programming, Visual Studio, C#, software engineering

## Sisältö

1	Johdanto .....	5
2	Toimeksianto .....	6
	2.1 Toimeksiantajan esittely .....	6
	2.2 Toimeksiannon aihe .....	8
3	Sähköisen lomakkeen toteutuksen suunnittelu .....	8
	3.1 Sovelluskehitysalusta .....	9
	3.2 Ohjelmointiympäristöt .....	10
	3.3 Ohjelmointi .....	10
	3.4 Tietokanta .....	12
	3.5 Tietoturva .....	13
	3.6 Sähköisen lomakkeen sisältö .....	14
4	Projektin toteutus .....	14
	4.1 Vaatimukset .....	14
	4.2 Toteutusmenetelmät .....	15
	4.3 Sähköisen lomakkeen toteuttaminen .....	17
	4.4 Testaus .....	19
	4.5 Käyttöönotto ja projektin päättäminen .....	22
5	Yhteenveto .....	22
	Lähteet .....	24

# 1 Johdanto

Opinnäytetyön tarkoituksena on tuottaa Pohjois-Karjalan ammattikorkeakoulun opintopsykologille sähköinen yhteydenottolomake, joka helpottaisi opiskelijan yhteydenottoa, sekä ajanvarausta opintopsykologille. Sähköisellä lomakkeella olisi myös tarkoitus kerätä tarvittavat esitiedot opiskelijasta ja hänen opiskeluun liittyvästä ongelmastaan. Nykyisellä toimintamallilla opiskelijat ovat joutuneet lähettämään omatoimisesti sähköpostiviestiä opintopsykologille. Ongelmana on ollut, että opintopsykologin sähköpostiosoite on ollut liian vaikea löytää tai opiskelijan lähettämän viestin sisältö on ollut puutteellinen tai liian suppea. Sähköisellä lomakkeella pyritään helpottamaan opiskelijan yhteydenottokynnystä, sekä keräämään yhteydenottoa varten tarvittavat esitiedot.

Pohjois-Karjalan ammattikorkeakoulun määrittelemä opintopsykologin tehtäväkuvaus on seuraavanlainen:

”Pohjois-Karjalan ammattikorkeakoulun opintopsykologin tehtävänä on tukea ja edistää opiskelijoiden opiskeluhyvinvointia ja opintojen sujuvaa etenemistä sekä antaa neuvontaa ja ohjausta opiskeluun liittyvissä ongelmatilanteissa. Opintopsykologipalvelujen on tarkoitus täydentää opiskelijapalveluiden, opinto-ohjaajien, opettaja- ja opiskelijatutoreiden sekä opettajien tarjoamaa ohjausta, tukien samalla ohjaus- ja opetushenkilöstöä heidän opetus- ja ohjaustehtävässään. Opiskelijoille opintopsykologi tarjoaa yksilö- ja ryhmäohjausta sekä luentoja/kursseja esim. opiskelutaitoihin liittyen. Opintopsykologin työn tärkeinä lähtökohtina ovat luottamuksellisuus, vapaaehtoisuus ja opiskelijälähtöinen yhteistyötoimeen.”

(Pohjois-Karjalan ammattikorkeakoulu 2011.)

Opinnäytetyö koostuu projektin toteutuksesta ja kirjallisesta osuudesta. Projektin tarkoitus on luoda yksinkertainen sähköinen lomake, jonka opiskelija osaa täyttää ensin näkemällään. Tämän kirjallisen osuuden tarkoituksena on antaa kattava kuva sähköisen lomakkeen tuottamisesta sisällöllisesti ja teknisesti, sekä antaa lukijalle perustietoa sähköisen lomakkeen eri toteuttamisvaihtoehdoista. Sähköinen lomake toteutetaan suomen ja englannin kielellä. Projektin toteutukseen käytetään Visual Studio 2010 ohjelmaa, ja ohjelmointikielenä käytetään C#. Opinnäytetyön toimeksiantaja on Pohjois-Karjalan ammattikorkeakoulun opintopsykologi.

## 2 Toimeksianto

Aloitin opinnäytetyön etsimisen kolmannen vuoden loppupuolella. Aiheeksi halusin löytää projektiluontoisesti toteuttavan opinnäytetyön, jossa itse työ toteutetaan mahdollisen toimeksiantajan vaatimuksien mukaan ja tehtäisiin myös kirjallinen raportti.

Opiskeluni aikanani olen työskennellyt matkapuhelin- ja laajakaistaoperaattori-yrityksessä. Aluksi yritin löytää sopivaa opinnäytetyön aihetta yrityksen sisältä, mutta lukuisista etsintäyrityksistä huolimatta sopivaa aihetta ei löytynyt. Koska sopivaa aihetta en itse löytänyt, niin otin yhteyttä oppilaitokseni opinnäytetyökoordinaattoriin, jonka kanssa sovimme tapaamisen opinnäytetyön aiheeseen liittyen.

Tapaamisen yhteydessä opinnäytetyökoordinaattori esitteli monia mielenkiintoisia opinnäytetyön aiheita sekä toimeksiantoja. Mielenkiintoisemmaksi aiheeksi nousi opintopsykologille toteuttava projektiluontoinen sähköiseen yhteydenottoon liittyvä toimeksianto. Aiheen tarkka sisältö, sekä toimeksiantajan vaatimukset olivat vielä selvittämättä, mutta aihe kiinnosti minua silti, joten sovimme opinnäytetyökoordinaattorin kanssa, että valitsen tämän aiheen opinnäytetyökseni.

Siinä yhteydessä myös sovittiin, että olen yhteydessä toimeksiantajaan tarkempaa opinnäytetyön määrittelyä varten. Tämän opinnäytetyön toimeksiantajana on siis Pohjois-Karjalan ammattikorkeakoulun opintopsykologi. Otin yhteydettä opintopsykologisiin sähköpostitse ja sovimme hänen kanssaan tapaamisen. Opintopsykologin tapaamisessa sovimme, että otan toimeksiannon vastaan opinnäytetyönäni ja teemme aiheesta toimeksiantosopimuksen.

### 2.1 Toimeksiantajan esittely

Opiskelijalle voi tulla opiskeluiden välissä tilanteita, jolloin opiskelija kaipaa tukea hyvinvoinnin, opiskelun sekä opintojen etenemisen edistämiseksi. Pohjois-Karjalan ammattikorkeakoulun opintopsykologin tehtävänä on tukea ja edistää opiskelijoiden opiskeluhyvinvointia ja opintojen sujuvaa etenemistä sekä antaa neuvontaa ja ohjausta

opiskeluun liittyvissä ongelmatilanteissa. Opintopsykologipalvelujen on tarkoitus täydentää ja edistää opiskelijapalveluiden, opinto-ohjaajien, opettaja- ja opiskelijatutoreiden sekä opettajien tarjoamaa ohjausta, tukien samalla ohjaus- ja opetushenkilöstöä heidän opetus- ja ohjaustehtävässään. (Pohjois-Karjalan ammattikorkeakoulu 2011 Opintopsykologipalvelut.)

Opiskelijoille opintopsykologi tarjoaa yksilö- ja ryhmäohjausta sekä luentoja/kursseja esim. opiskelutaitoihin liittyen. Opintopsykologin työn tärkeinä lähtökohtina ovat luottamuksellisuus, vapaaehtoisuus ja opiskelijälähtöinen yhteistyö. Opintopsykologin kanssa on mahdollista keskustella erilaisista opiskeluun, oppimiseen ja oppimisympäristöön liittyvistä pulmista. Opiskelija voi ottaa yhteyttä opintopsykologiin esim. opiskelutaitoihin, tavoitteisiin, motivaatioon, jaksamiseen tai ajankäyttöön liittyvissä asioissa. Ajatusten vaihtamisesta voi olla apua, jos oppimisvaikeuteen, jännittämiseen, mielialaan tai elämäntilanteeseen liittyvät haasteet kuormittavat opiskelua. Myös opinnäytetyön tekemiseen, valmistumiseen, ammatti-identiteettiin ja työelämään siirtymiseen liittyvät teemat ovat yleisiä keskustelun aiheita. Keskustelujen tavoitteena on yhdessä opiskelijan kanssa jäsentää opiskelun esteitä ja haasteita, etsiä ratkaisuja – uudenlaisia näkökulmia ja toimintatapoja - opiskelu edellytyksen parantamiseksi. Oman oppijaminäkuvan tutkiskeleminen, itsetuntemuksen lisääntyminen ja omien voimavarojen tunnistaminen ja vahvistaminen on keskusteluissa tärkeää. (Pohjois-Karjalan ammattikorkeakoulu 2011, Opintopsykologipalvelut.)

Opintopsykologipalvelut ovat opiskelijoille maksuttomia. Ensimmäisellä tapaamiskerralla kartoitetaan opiskelijan tilannetta ja sovitaan jatkosta. Tapaamiskertoja voi olla esim. 1 – 5, tarpeen ja tilanteen mukaan ja niistä sovitaan opintopsykologin kanssa. Ajanvaraus tapahtuu tällä hetkellä sähköpostilla. Opintopsykologi toimii yhteistyössä opiskeluterveydenhuollon kanssa ja ohjaa tarvittaessa muiden terveydenhuollon palvelujen käyttöön. (Pohjois-Karjalan ammattikorkeakoulu 2011, Opintopsykologipalvelut.)

## 2.2 Toimeksiannon aihe

Opintopsykologille on ollut pitkään ongelmana opiskelijoiden yhteydenotto ja ajanvaraus. Nykyisellä toimintamallilla opiskelijat joutuvat omatoimisesti lähettämään sähköpostia opintopsykologille ja liittämään viestiin opintopsykologin pyytämät esitiedot, jotka löytyvät Pohjois-Karjalan ammattikorkeakoulun opiskelijaportaali sivustolta. Ongelmana on ollut, että opintopsykologin sähköpostiosoite on ollut vaikea löytää tai, opiskelijan lähettämän viestin sisältö on ollut puutteellinen tai liian suppea.

Opintopsykologin toiveena on ollut, että hänelle tuotettaisiin sähköinen lomake, jonka avulla opiskelija pystyisi ottamaan yhteyttä häneen ilman, että opiskelijan tulisi itse lähettää viestiä omasta sähköpostistaan. Sähköiseltä yhteydenottolomakkeelta opiskelijan tulee täyttää vaaditut kohdat, jolla opiskelija pystytään tunnistamaan, sekä hänen ongelmastaan saadaan lyhyt kuvaus. Kun opiskelija on täyttänyt vaaditut kohdat lomakkeesta hän voi lähettää sen eteenpäin opintopsykologille. Viestin saapumisen jälkeen opintopsykologi pystyisi olemaan yhteydessä opiskelijaan annetuilla esitiedoilla ja sopimaan tarvittaessa ajanvarauksen opintopsykologin vastaanotolle.

Sähköisellä yhteydenottolomakkeella on myös tarkoitus helpottaa ajanvarauksia ja siihen kohdistuvia muutoksia. Opintopsykologi pystyy saamaan yhteyden opiskelijan antamien yhteystietojen kautta. Mahdollinen ajanvaraus tapahtuisi vasta kun opintopsykologi on ottanut yhteyttä opiskelijaan, joten itse ajanvaraustapahtumaa ei käsitellä sähköisellä lomakkeella. Opintopsykologi palvelee myös kansainvälisiä opiskelijoita, joten sähköinen lomake tulee tuottaa myöskin englanniksi. Englanninkielisestä versiosta tulee löytyä samat tiedon kuin suomenkielisestä lomakkeesta.

## 3 Sähköisen lomakkeen toteutuksen suunnittelu

Projektin tarkoituksena on tuottaa opintopsykologille sähköinen yhteydenottolomake, joka helpottaisi opiskelijan yhteydenottoa opintopsykologiin. Tämä sähköinen lomake sijoitetaan Pohjois-Karjalan ammattikorkeakoulun opiskelijapalvelun tuottamalle opiskelijaportaali-sivustolle, jota kutsutaan nimellä Pakki. Pakki-sivusto on opiskelijoiden

tukisivusto, jonka kautta opiskelijat saavat ohjeita ja apua opiskeluun liittyvissä asioissa.

Sähköinen lomake pystytään toteuttamana monella eritavalla, joten sähköisen lomakkeen määrittely ja sen sisältö perustuu toimeksiantajan vaatimukseen. Toimeksiantaja on omalta osaltaan antanut tiedot ja toiveet mitä sähköisen lomakkeen tulisi sisältää, ja miltä sen tulisi näyttää. (Haikala & Mikkonen 2011, 19.)

Työhön liittyvät tekniset vaatimukset tulevat Pohjois-Karjalan ammattikorkeakoulun tietohallinnon puolelta. Tietohallinto on antanut teknisissä vaatimuksissaan oman näkemyksensä millä tekniikalla ja missä ympäristössä toteutettavan työn tulisi toimia. Tarkempi työn sisältö käsitellään luvussa 4.

### **3.1 Sovelluskehitysalusta**

Sovelluskehitysalustoja on tarjolla monia eri vaihtoehtoja. Tämän projektin kannalta sovelluskehitysalustan valinta pitää tehdä toimeksiantajan vaatimusten mukaisesti alustan ylläpidon ja päivitettävyyden kannalta. Pohjois-Karjalan ammattikorkeakoulun tietohallinnon vaatimusten mukaan sovelluskehitysalustan tulisi olla .Net framework yhteensopiva.

#### **.NET framework**

.NET-sovelluskehitys (.NET framework) on tarkoitettu seuraavan teknologia sukupolven alustaksi. Se on suunniteltu toimivaksi hajautetuissa ympäristöissä, joissa tietokoneet ja laitteet ovat yhteydessä toisiinsa Internetin kautta.

Sovelluskehitys on ohjelmisto, joka muodostaa rungon sen päälle rakennettaville tietokoneohjelmille. Sovelluskehitys on ohjelmoinnin apuväline, jonka tarkoituksena on nopeuttaa uusien ohjelmistojen valmistumista ja niiden toteuttamista. Se nopeuttaa ohjelmoinnin kehitystyötä tarjoamalla valmiiksi rakennettuja osia, joita ei tarvitse kirjoittaa uudelleen ohjelman kehityksen aikana. Tyypillisesti toimivat ohjelmat kirjoitetaan kehityksen päälle. (Moghadampour 2009, 12.)

.NET on alun perin Windows-käyttöjärjestelmien kehitysympäristöksi luotu sovelluskehys, jonka ensimmäinen kaupallinen versio ilmestyi vuoden 2000 alkupuolella. .NET-sovelluskehityksen tavoitteena on tarjota yhdenmukainen ympäristö ja laaja toiminnallisuuksia tarjoava kirjasto, jonka erikoisuutena on lähdekoodikielien vapaus. .NET-sovelluskehys tarjoaa virtuaalikoneen, joka huolehtii ohjelman ajamisesta. Virtuaalikone tarjoaa näin yhteisen rajapinnan eri ohjelmointikielille toteutettavien moduulien yhteensovittamisesta sekä yhteisen ajoympäristön riippumatta käytettävän laitteiston järjestelmätasosta. .NET-sovelluskehityksellä voidaan kirjoittaa muun muassa Windows- ja web-sovelluksia, mobiililaitteisiin tarkoitettuja sovelluksia, sekä sellaisia järjestelmiä, jotka koostuvat keskenään Internetin kautta kommunikoivista toisiinsa liitettyistä palveluista. (Moghadampour 2009, 12.)

### **3.2 Ohjelmointiympäristöt**

.NET ohjelmointiympäristöjä on rajallinen määrä, joten oikean ja käyttötarkoituksiin sopivan ohjelmointiympäristön valinta on tärkeää projektin toteutuksen ja jatkokehityksen kannalta. .NET ohjelmia voidaan kehittää, sekä Microsoftin että muiden toimittajien tarjoamilla ohjelmointityökaluilla. Tärkein Microsoftin tarjoama ohjelmointityökalu on Visual Studio, josta on saatavilla eri käyttötarkoituksiin sopivia kokoonpanoja. Visual Studio on saatavana kaupallisena sekä kokeilu- ja ilmaisversiona. (Moghadampour 2009, 12.)

Yksi parhaimmista avoimen lähdekoodin .NET ohjelmointiympäristöissä on Mono. Mono on Xamarinin sponsoroima avoimen lähdekoodin .NET-kehitysympäristö, joka on käyttöjärjestelmästä riippumaton. Sillä on mahdollista suorittaa .NET-ohjelmia millä tahansa alustalla. (Moghadampour 2009, 12.)

### **3.3 Ohjelmointi**

Yksi tapa sähköisen yhteydenottolomakkeen tuottamiseen on HTML koodilla toteuttava täytettävä lomake, mutta tällöin sähköisen lomakkeen käyttäjälle tulisi olla omalla tietokoneella käytössä sähköpostiohjelma, johon lomake tiedot siirtyisivät. HTML koo-

dilla toteutettava lomake ei myöskään vastaisi toiminnallisuuksiltaan toimeksiantajan toiveita. Toimeksiantajan toiveena on ollut, että käyttäjällä ei tarvitse olla viestin lähettämiseen omaa sähköpostia käytössään vaan kun käyttäjä täyttää lomakkeen ja lähettää sen eteenpäin niin tällöin lomaketiedot menevät automaattisesti opintopsykologin sähköpostiin. Tämän kaltaisen toiminnon toteuttamiseen tarvitaan virtuaalikone, joka huolehtii ohjelman ajamisesta ja sen toiminnasta.

Toteutettavan työn kannalta on tärkeää, että ohjelmointi tehdään toimeksiantajan tietohallinnon vaatimuksien mukaan, jotta valmistuneen työn ylläpito ja päivittäminen olisi mahdollisimman yksinkertaista ja tehokasta. Ohjelmoinnin toteuttamiseen tarvitaan sovelluskehitysalusta, joka tukee yllämainittuja toimenpiteitä. Tähän tarpeeseen .NET-sovelluskehitysalusta on kaikkein sopivin, yksinkertaisin ja monipuolisin. Näin ollen .NET tukee myös toimeksiantajan tietohallinnon asettamaa vaatimusta ohjelmointi-sovelluskehityksen valinnassa.

.NET-ohjelmointiympäristöllä on tuki neljälle eri ohjelmointikielille, ja näistä neljästä ohjelmointikielestä tulisi valita yksi ohjelmointikieli, jolla lomakkeen ohjelmointi tettäisiin. .NET-ympäristöön kuuluvat seuraavat ohjelmointikielet: Visual Basic .NET, C#, Visual C++ sekä Visual J#. (Moghadampour 2009, 16.) Seuraavaksi kuvataan mainitut .NET-ohjelmointiympäristön ohjelmointikielet.

### **Visual Basic .NET**

VB.NET on .NET ympäristöön tarkoitettu oliopohjainen versio Microsoftin Visual Basic -ohjelmointikielestä. Sen käyttötarkoituksena on ollut helpottaa webpalveluiden kehitystä. VB.NET on täysin oliopohjainen, joten se tukee olio-ohjelmoinnin peruskäsitteitä. Visual Basic on puolestaan kolmannen sukupolven tapahtumakeskeinen ohjelmointikieli. Se kehitettiin Basic-ohjelmointikielestä ja sillä pystytään kehittämään nopeasti graafisia käyttöliittymiä, luomaan tietokanta yhteyksiä, sekä ohjaimia että olioita. Visual Basic viimeinen versio (Visual Basic 6) julkaistiin vuonna 1998, mutta Microsoftin tuki Visual Basic -kieleen loppui marraskuussa 2008, kun Visual Basic.NET:sta tuli sen seuraaja. (Moghadampour 2009, 16.)

### **Visual C++**

Visual C++ (Microsoft Visual C++, MSVC) voidaan kirjoittaa sovelluksia Microsoftin Windows-käyttöjärjestelmälle. Pohjimmiltaan se nojaa C++ ohjelmointikielen tarjoamiin ominaisuuksiin, mutta se käyttää myös muita apukirjastoja. MSVC tukee isoa joukkoa ohjelmointikieliä, kuten C++ ja Managed C++. Ne ovat Visual C++ -kielelle ja kääntäjälle tehtyjä laajennuksia, joilla voidaan kirjoittaa .NET-ohjelmia ja ottaa käyttöön sovelluskehysten funktioina. (Moghadampour 2009, 13-14.)

### **Visual C#**

Visual C# yhdistää C++:n laskentatehokkuuden ja Visual Basic helppokäyttöisyyden C# perustuu C++ -kieleen ja se sisältää Java-kielen piirteitä. .NET-teknologian tarkoitus on helpottaa tiedon vaihtoa, webpalveluiden kehitystä sekä helposti siirrettävien sovellusten kehitystä. C# helpottaa juurinkin edellä mainittuja osa-alueita ohjelmointityössä, koska se käyttää tiedonsiirtovälineenä XML:ää ja olioiden välisen kommunikoinnin toteuttamiseen SOAP:ia. (Moghadampour 2009, 14.)

### **Visual J#**

Visual J# koostuu joukosta ohjelmointityökaluja, jotka mahdollistavat Java-kielen ohjelmien kirjoittamista ja ajamista .NET-ajoympäristössä. .NET-ympäristö ei sisällä Java-ohjelmien ajamista varten tarvittavaa virtuaalikonetta, mutta se mahdollistaa niiden ajamisen sovelluskehysten CLR:llä. CLR on virtuaalikone, jossa ohjelmat suoritetaan toisistaan eristetyissä ja hallituissa ympäristöissä erillään koneen muista prosesseista. (Moghadampour 2009, 14.)

## **3.4 Tietokanta**

Tietokannan tarkoituksena tiedon järkevä tallennus ja tiedon haku. Tietokannoiksi luetaan erilaiset luettelo, rekisteri, mittaustiedot, laskutustiedot, hinnasto – kaikki tiedot mitkä sisältävät tietoa eli dataa. Laajoja tietokantoja käytetään raskaissa tuotantoympäristöissä (pankit, kirjastot, kauppaketjut). Pienempiä tietokantoja käytetään itse sovelluskehyksessä henkilökohtaisena tietokantana tai pienimuotoisena sovelluksena. (Polvinen 1999, 2.)

Sähköinen lomake ei tarvitse tietokantaa tässä vaiheessa, koska lähetettävä tieto siirtyy suoraan postipalvelimelta vastaanottajalle. Lomakkeella käsitellään lähettäjän luottamuksellisia tietoja, joten tietojen tallennus ei ole myöskään tietoturvasyistä johtuen suotavaa. Tietokanta on mahdollista sisällyttää lomakkeeseen tulevaisuuden jatkokehityksessä, mikäli se nähdään tarpeelliseksi ominaisuudeksi.

### 3.5 Tietoturva

”Luonnollisen henkilön tietoturvallisuudella tarkoitetaan tietojen, järjestelmien ja palvelujen suojaamista sekä normaali- että poikkeusoloissa hallinnollisten ja teknisten toimenpiteiden avulla. Tietoturvallisuus rakentuu tiedon kolmen ominaisuuden - luottamuksellisuuden, eheyden ja käytettävyyden - turvaamisesta.” (Viestintävirasto 2012.)

Tietoturva on yksi lomakkeen tärkeimmistä asioista, koska jokaisella Suomessa oleskelevalla luonnollisella henkilöllä on oikeus luottamukselliseen viestintään. Sillä tarkoitetaan sitä, että kukaan ei saa ilman laillista tai laissa säädettyä oikeutta tai ilman viestinnän osapuolten suostumusta käsitellä toisen henkilön viestejä tai viestintään liittyviä tunnistamistietoja sähköisen viestinnän tietosuojalain mukaisesti. Sähköisellä lomakkeella tullaan käsittelemään lähettäjän henkilökohtaisia tietoja, joten tietojen ehdoton luottamuksellisuus on tärkeä huomioida tietoja käsitellessä, koska Suomessa sähköisen viestinnän luottamuksellisuus on turvattu laissa. (Viestintävirasto 2012.)

Suomessa toimivien viestintäpalvelujen tarjoajien, kuten esimerkiksi internet sivujen ylläpitäjien on huolehdittava palvelujensa tietoturvasta ja viestinnässä tapahtuvan sisällön luottamuksellisuudesta. Maailmassa on valtioita, jotka eivät takaa lainsäädännöllisesti viestinnän luottamuksellisuudelle yhtä ehdotonta ja kattavaa suojaa kuin Suomessa. (Viestintävirasto 2008.) Sähköisen lomakkeen tietoturva-asioita käsitellään tarkemmin luvussa 4.

### **3.6 Sähköisen lomakkeen sisältö**

Lomakkeen suunnittelussa otettava huomioon tietojen looginen ja yksinkertainen asettelu, tietojen yksinkertaisuus, sekä käyttäjälle on suotava mahdollisuus korjata virheensä. Sähköinen lomake tulee sisältämään monia eri tekstikenttiä, jotka käyttäjän tulee täyttää ennen lomakkeen lähettämistä. Nykyisellä toimintamallilla käyttäjä on lisännyt sähköpostiviestiin opintopsykologin vaatimat kirjalliset esitetiedot, joilla opintopsykologi saa kuvan opiskelijasta, sekä hänen mahdollisesta opiskeluun liittyvästä ongelmastaan. Opintopsykologin pyytämät kirjalliset tiedot ovat seuraavat:

1. Mikä motivoi sinua hakeutumaan nykyiselle opintoalallesi?
2. Mikä opinnoissasi (nyt tai aikaisemmin) on sujunut hyvin, missä olet onnistunut?
3. Mikä opinnoissasi sinua tällä hetkellä huolestuttaa?
4. Mistä erityisesti haluat keskustella opintopsykologin kanssa?

Toteutettavalle sähköiselle lomakkeelle on tarkoitus saada perustietoja opiskelijasta, hänen ongelmastaan, sekä opiskelijan yhteystietoja. Sähköisen lomakkeen sisältö käsitellään myös tarkemmin seuraavassa luvussa.

## **4 Projektin toteutus**

### **4.1 Vaatimukset**

Projektille toteutukselle asetettiin vaatimuksia toimeksiantajalta, sekä Pohjois-Karjalan ammattikorkeakoulun tietohallinnosta teknisien vaatimusten osalta. Toimeksiantajan vaatimukset sähköiselle lomakkeelle olivat lomakkeeseen sisällytettävät tekstikenttätiedot, joilla saadaan tunnistettua opiskelija, sekä opiskelijan yhteydenotto syy. Sähköiselle lomakkeella täytettävät esitetiedot on tarkoitus pitää suppeana luottamuksellisista syistä johtuen, sekä että lomakkeella ei ole tarkoitus saada selville kokonaan opiskelijan yhteydenotto syytä vaan ainoastaan lähtökohdat yhteydenotolle. Opintopsykologin toiveena on ollut, että lomake säilyttäisi inhimillisen otteen tietoja täyttäessä ja niitä käsitel-

lessä. Tällä tarkoitetaan sitä, että opiskelijan ja opintopsykologin keskenään tapahtuva vuorovaikutus ei tapahdu jatkossa sähköisesti vaan kasvotusten.

Sähköisen lomakkeen tulee sisältämään seuraavia tietoja, jotka opiskelijan tulee täyttää ennen lomakkeen lähettämistä:

- Etunimi.
- Sukunimi.
- Syntymäaika
- Puhelinnumero.
- Sähköposti.
- Saapumisryhmätunnus
- Opiskelijanumero
- Viesti ja ajanvaraus opintopsykologille.

Etunimi, sukunimi, syntymäaika, puhelinnumero, sähköposti, saapumisryhmätunnus, sekä opiskelijanumero ovat lomakkeen pakollisia tietoja. Ilman näitä tietoja opintopsykologilla ei ole mahdollisuutta tunnistaa opiskelijaa. Lisäksi hän tarvitsee näitä tietoja mahdollisissa ajanvarausmuutoksissa.

Pohjois-Karjalan ammattikorkeakoulun tietohallinto antoi sähköiselle lomakkeelle teknisen vaatimukset, koska valmistunut lomake tullaan siirtämään heidän palvelimelleen, sekä he vastaavat sähköisen lomakkeen käyttöönotosta että ylläpidosta, sekä mahdollisesti jatkokehityksestä. Lomakkeen vaatimuksena on, että se tuotetaan C# kielellä Microsoft Visual Studio -tuotantoympäristössä, sekä mikäli tietokanta puoli halutaan lisätä myöhemmin niin tulisi tällöin käyttää SQL Server 2005, 2008 tai 2008R2. Ohjelmoinnilla toteutettava sähköpostin lähetys tulisi toteuttaa System.Net.Mail -kirjastoavaruutta hyödyntäen.

## **4.2 Toteutusmenetelmät**

Projektin toteutus menetelmiksi valitsin Pohjois-Karjalan ammattikorkeakoulun tietohallinnon vaatimat menetelmät ylläpidollisista, että käyttöönottavista syistä johtuen.

Ylläpidolliset ja käyttöönotettavat syyt käsitellään kohdassa 4.5. Ohjelmointi kielenä käytetään C# kieltä. Tuotantoympäristöksi valitsin tietohallinnon suosittelman Visual Studio 2010. Sähköpostin lähetysohjelmointina käytetään System.Net.Mail -kirjastoa. Tietokanta ominaisuutta ei lisätä lomakkeeseen tässä vaiheessa.

## **C#**

Sähköisen lomakkeen ohjelmointiin käytetään C#-ohjelmointikieltä. Ohjelmointikielen valintaan vaikuttaneita kriteereitä selvitetään vielä tarkemmin alla olevissa kappaleissa. C# yhdistää C++:n laskentatehokkuuden ja Visual Basic helppokäyttöisyyden C# perustuu C++ -kieleen ja se sisältää Java-kielen piirteitä. .NET teknologian tarkoitus on helpottaa ja nopeuttaa tiedon vaihtoa, webpalveluiden kehitystä sekä helposti siirrettävien sovellusten kehitystä. C# helpottaa juurinkin edellä mainittuja osa-alueita ohjelmointityössä, koska se käyttää tiedonsiirtovälineenä XML:ää ja olioiden välisen kommunikoinnin toteuttamiseen SOAP:ia. (Moghadampour 2009, 14.)

C#-kieli on Microsoftin vuonna 2000 julkaisema oliopohjainen ohjelmointikieli, jonka ensimmäinen standardi hyväksyttiin vuonna 2001. Pääsuunnittelijana toimi Anders Heljsberg, joka on ollut myös kehittämässä Turbo Pascalia, sekä Delphiä ohjelmointikieliä. Ennen C#:n kehittämistä Anders oli kehittämässä Microsoftilla J++-kieltä, joka on Java-kielinen ohjelmointiympäristö. C#-ohjelmointikielen tarkoitus oli korvata J++-kieli ja luoda moniin eri ympäristöihin soveltuva ja helppokäyttöinen ohjelmointikieli. (Horsmalahi & Toivanen 2011.)

## **Visual Studio**

Projektin toteutuksen sovelluskehittimeksi valittiin Visual Studio. Se on Microsoftin sovelluskehitin Windows-pohjaisiin ympäristöihin. Kun Visual Studio ensi kertaa julkaistiin vuonna 1997, oli Microsoftin visiona luoda kehitysväline, joka saumattomasti integroisi yhteen erilaiset ohjelmointikieliset ja tarjoaisi kaikille ohjelmoijille joukon yhteisiä työkaluja. Visual Studion uusin versio on Visual Studio 2010. Uusin versio on tuonut mukanaan uusia mielenkiintoisia tekniikoita, kuten usean näytön tuen, sekä Sharepoint tuen. (Järvinen 2008, 2-3.)

Perinteisesti nimi Visual Studio on yhdistetty Microsoftin tärkeimpään Windows-kehittimeen. Visual Studiolla kehitetään pääasiassa .NET pohjaisia sovelluksia valmistajan omiin käyttöjärjestelmäympäristöihin, toisin sanoen Windows-pohjaisiin työasemiin, palvelimiin, sekä nykyisin myös viihde- ja mobiililaitteisiin. Voidaan sanoa, että Visual Studio on kokonainen kehittäjätuoteperhe, josta löytyy eri tuotekokonaisuuksia harrastelija-versioista aina laajoihin ammattilaisille tarkoitettuihin versioihin asti. Visual Studio on sovelluskehittimenä erittäin nykyaikainen ja hyvin monipuolinen, sekä kilpailukykyinen nykyaikaisilla markkinoilla. Siinä integroituvat saumattomasti keskenään kääntäjä, koodieditori, visuaaliset suunnitteluvälineet, sekä virheenjäljitin. (Järvinen 2008, 2-3.)

### **System.Net.Mail**

Sähköpostin lähettäminen toteutettiin System.Net.Mail –nimiavaruuskirjaston avulla. Tämä nimiavaruuskirjasto, mahdollistaa sähköpostin lähetyksen C#-kielellä. System.Net.Mail omaisuus on tullut NET Framework -ympäristöön 2.0 versiosta alkaen.

### **4.3 Sähköisen lomakkeen toteuttaminen**

Aloitin projektin toteutuksen tutustumalla toimeksiantajan vaatimuksiin, sekä tietohallinnon ehdottamiin menetelmiin. Projektin toteutuksen kannalta ensimmäinen ongelmana oli ohjelmointi. Ohjelmointi piti tehdä C#-ohjelmointikielellä. Koska C#-ohjelmointikieli ei ollut entuudestaan tuttu minulle, niin C#-ohjelmointikielen opiskelu ja syvällisempi tutustuminen vaadittuun ohjelmointikielen oli välttämätöntä projektin onnistumisen kannalta. Lomakkeen tietoturva oli myös otettava huomioon lomakkeen toteutuksessa, koska lomakkeella käsitellään luottamuksellisia tietoja.

### **Lomakkeen ulkoasu ja sisältö**

Toimeksiantajalle ei ollut erityisiä toivomuksia lomakkeen ulkoasun suhteen. Ainoana poikkeuksena oli, että lomakkeen sisältö tulisi olla loogisesta ja yksinkertaisesti aseteltu, ja lomakkeen tulisi olla graafiselta värimaailmaltaan yhteensopiva Pohjois-Karjalan ammattikorkeakoulun internet sivujen kanssa.

## **Tietoturva**

Lomakkeen tietoturva on lomakkeen tärkeimpiä huomioon otettavia asioita, koska lomakkeella käsitellään luottamuksellisia asioita, jotka ovat tarkoitettu vain opintopsykologin nähtäväksi. Lomakkeen käyttäjälle pitää taata hänestä käsittelevien tietojen luottamuksellisuus ja että tietoja ei pääse näkemään muut kuin opintopsykologi. Lomake tullaan sijoittamaan Pohjois-Karjalan ammattikorkeakoulun opiskelijaportaali sivustolla opintopsykologin osioon. Opintopsykologin osiota ei ole mahdollista nähdä ellei opiskelija ole kirjautunut sisälle sivustolle, joten lomaketta ei pääse kuka tahansa näkemään tai lähettämään vaan opiskelijan tulee olla kirjautunut sisälle opiskelijaportaali sivustolle. Pakollinen kirjautuminen sivustolle estää turhia kontakteja, sekä se ennalta ehkäisee lomakkeen mahdollista väärinkäyttöä

Microsoftia ja erityisesti Windows-käyttöjärjestelmää on pitkään moitittu heikosta tietoturvasta. Niinpä yksi tärkeimmistä huomioista toteutuksessa ja ohjelmointiympäristön valinnassa oli tietoturva. .NET-ympäristön keskeisissä suunnittelun lähtökohdista oli tarjota ja kohentaa sovelluksen tietoturvaa tarjoamalla kehittäjälle perinteistä Win32:ta turvallisempi alusta, luotettava ajoympäristö, sekä mahdollisuus Windows-tietoturvaominaisuuksien helppo hyödyntäminen omissa sovelluksissa. Tämä tarkoittaa, sitä että .NET takaa jo sovelluskehys alustana turvallisen ohjelmointiympäristön. (Järvinen 2008, 27.)

## **Ohjelmointi**

Ohjelmointikieleen tutustumisen aloitin osallistumalla C#-ohjelmoinnin perusteet kursille. Kurssilla sain perustietoa C#-ohjelmoinnista. Kurssilla opittuja asioita pystyn hyödyntämään sähköisen lomakkeen tuottamisessa. Sähköisen lomakkeen ohjelmointi oli sinänsä helppo aloittaa, koska Pohjois-Karjalan ammattikorkeakoululla oli Visual Studio 2010 -lisenssi, jolla pystyin tuottamaan sähköisen lomakkeen.

Visual Studio 2010 tarjoaa käyttäjälleen mahdollisuuden valita monia eri ohjelmistoympäristöjä eri ohjelmistokielille. Tässä projektissa on tarkoituksena käyttää C#-ohjelmointikieltä hyödyntäen ASP.NET-ohjelmointimenetelmiä. Kyseinen ohjelmointiympäristö soveltui itselleni parhaaksi mahdolliseksi tuotantotavaksi. Itse ohjelman toteutus alkoi siitä, että suunnittelin sähköisen lomakkeen visuaalisen ilmeen sekä lomakkeen tekstielementtien loogisen sijoituksen.

Visuaalisen ilmeen ja tarkan elementtien suunnittelun seurauksena sähköiseen lomakkeeseen muodostui seitsemän tekstikenttää. Jokainen tekstikenttä on yksilöllinen ja siihen tulee syöttää vaadittu tieto, joka ilmenee tekstikentän yläpuolella. Tekstikenttä tiedot ovat niin sanottuja pakollisia tietoja, joten mikäli opiskelija ei täytä, jotain vaadittua tekstikenttää niin tällöin lomaketta ei ole mahdollista lähettää eteenpäin. Mikäli tekstikenttä on tyhjä niin antaa lomake tällöin virheilmoituksen tekstikentän oikealle puolelle, jossa ilmoitetaan englannin kielellä “You must enter a value in the textbox”. Tämä tarkoittaa, sitä että tekstikentässä ei ole sisältöä. Kun tekstikenttään on syötetty sen vaatima tieto, niin tekstikentän oikealla puolella oleva virheilmoitus poistuu näkyvistä.

Lomakkeen alareunaan on sijoitettu kaksi painiketta, jotka on nimetty ”Lähetä | Send” ja ”Tyhjennä | Clear” -painikkeiksi. Lähetä painikkeen tarkoituksena on lähettää lomakkeeseen täytetyt tekstikenttä tiedot opintopsykologin sähköpostiin siistinä ja koottuna versiona. Tekstikenttä tietojen lähetykseen käytetään System.Net.Mail kirjastoa. Lomakkeeseen on määritetty Pohjois-Karjalan ammattikorkeakoulun postipalvelimen vaatimat asetukset, jotta viesti ohjautuu suoraan opintopsykologin sähköpostiin.

Lomake antaa käyttäjälleen ilmoituksen onnistuneesta lomakkeen lähettämisestä. Mikäli lähetyksen aikana tapahtuu virhe, niin lomake antaa tästä käyttäjälleen virheilmoituksen. Ilmoitus ilmestyy lomakkeen oikeaan reunaan. Tyhjennä painikkeella lomakkeen käyttäjä voi tarvittaessa tyhjentää lomakkeeseen täytetyt tekstikenttätiedot.

#### **4.4 Testaus**

Testaukseen liittyvät työvaiheet ovat testauksen suunnittelu, testiympäristön luonti, testin suorittaminen ja tulosten tarkistelu. Näihin työvaiheisiin ja niihin läheisesti liittyvien virheiden jäljitykseen ja korjaukseen kuluu tyypillisesti yli puolen ohjelmistoprojektin ajasta, joten testaukseen ja läpivientiin on hyvä kiinnittää huomioita. Tavanomaisessa puhekielessä termillä testaus tarkoitetaan lähes mitä tahansa kokeilemista. Ohjelmistojen testauksen yhteydessä testaus määritellään perinteisesti suunnitelmalliseksi virheiden etsimiseksi ohjelmaa tai sen osaa suorittamalla. (Haikala & Mikkonen 2011, 205.)

Testauksen avulla on mahdollista osoittaa, että ohjelmassa on virheitä. Ohjelman virheettömyyttä ei ole mahdollista osoittaa edes yksinkertaisissa tapauksissakaan. Käytännössä ohjelman testauksessa pystytään kattamaan vain pieni häviävä murto-osa kaikissa mahdollisista toiminnoista. Tällä tarkoitetaan, sitä että kaikkien virheiden eliminointi ei ole mahdollista, mutta testauksella pyritään minimoimaan ja käymään läpi virheiden aiheuttamaan haittaa. Tämä ei tarkoita kuitenkaan sitä, että testaukseen ei kannata käyttää aikaa, vaan sitä, että ohjelman toimivuuteen ei kannata liiaksi luottaa hyvissäkiin testaustuloksista huolimatta. (Haikala & Mikkonen 2011, 205.)

Sovelluksien kehittäminen on niin vaativaa työtä, että pieniä virheitä syntyy väistämättä kenelle tahansa. Näiden virheiden etsiminen on toisinaan erittäin työlästä, joten Visual Studion laadukkaan apuvälineet ovat suureksi avuksi. Virheiden etsimistä sovelluksessa kutsutaan virheenjäljitykseksi. Virheiden jäljittäminen on useimmiten melko mekaanista ja suoraviivaista toimintaa, mutta vaatii silti Visual Studio 2010 -ominaisuuksien tuntemusta. (Järvinen 2008, 162.)

### **Virheenjäljitys**

Virheenjäljitys on muutamaa tyypillistä vaihetta toistava prosessi. Prosessin vaiheet ovat karkeasti ottaen virheen tunnistaminen, virheellisen ohjelmakoodin paikallistaminen, virheen korjaaminen sekä loppu tuloksen testaaminen. Ensimmäisessä vaiheessa selvitetään johtuuko havaittu virhe väärin toimivasta ohjelmasta vaiko ulkopuolisista seikoista, kuten esimerkiksi odottamattomasta syötteestä. Mikäli kyseessä on virheellisesti korjattu ohjelmakoodi, niin tällöin pyritään selvittämään missä kohdassa ohjelmakoodia ongelma on ilmennyt. (Järvinen 2008, 162.)

Virheellisen tai odottamattoman syöteen tapauksissa taas ei yleensä tarvita varsinaista virheenjäljitystä, sillä riittää, että ohjelmakoodia muutetaan ottamaan paremmin huomioon myös mahdollisen virheen syöttö. Aina virheen paikallistaminen ei kuitenkaan ole helppoa. Alkuperäinen virheen aiheuttama ohjelmakoodi ei välttämättä ole siellä, missä käyttäjän näkemä poikkeus nostettiin esille ja jonka sovelluksen käyttäjä lopulta näytti. Onneksi .NET-ohjelmoinnissa monet tällaiset tyypilliset ohjelmointivirheet jäävät pois, joten virheiden luonne muuttuu.

Myös C#-ohjelmoinnissa ohjelmointikieli pyrkii ohjaamaan kehittäjää kirjoittamaan laadukkaampaa koodia, sillä esimerkiksi alustamattomia muuttujia ei voida käyttää. Silti virheiden virheilmoitusten perusteella ei aina ole selkeää, mistä päin ohjelmakoodia virhettä tulisi hakea. (Järvinen 2008, 163.)

Virheiden minimoinnin haasteellisuutta lisää toinenkin asia. Mitä mutkikkaammaksi Visual Studiolla tehtävät sovellukset käyvät, sitä haastavammiksi virheiden etsiminen käy. Varsinkin monitasoarkkitehtuuria noudattavat monesta osasta koostuvat sovellukset, kuten hajautetut web-sovellukset, vaativat kehittäjältä paljon tietoa ja taitoa, ja tämä vaatimus heijastuu myös virheidenjäljitykseen. (Järvinen 2008, 163.)

Vaativassa sovelluskehityksessä on syytä turvautua muuttamaan ratkaisumalliin, joilla ongelmasta saadaan lisää tietoa ja löydetään lähes jokaisessa tilanteessa ratkaisu tai ainakin vaihtoehtoinen ohjelmointi ratkaisu. Näitä ovat toistettavuus, sovelluksen pilkkominen, sekä monelta ohjelmoijalta puuttuva taito eli nöyryys. Ensimmäinen lähtökoh- ta onnistuneelle virheenjäljitykselle on, että ongelma voidaan toistaa uudelleen. Toistet- tavuus on tärkeää, sillä ainoastaan selkeiden toisto ohjeiden avulla voidaan ongelmaa toistaa eri testaus vaiheiden jälkeen. Toistettavuus on tärkeää ongelman ratkaisun kaikil- la osa-alueilla, niin teknisenä ohjelmointi virheiden selvittämisen kuin loppukäyttäjän havaitsemien sovellusongelmien tasolla. (Järvinen 2008, 163.)

Lomakkeessa lähdekoodissa on yksinkertainen ohjelmoitu poikkeuksen käsittely. Tä- män tarkoituksena on ilmoittaa käyttäjälle mahdollisesta virheestä lomaketta lähettäes- sä. Poikkeuksen käsittely on tuotettu try catch -lauseella. Mahdollisesti virheen aiheut- tava koodi laitetaan try-lohkoon ja virheen sattuessa tehtävät toimenpiteet catch- lohkon. (Kolari 2012.)

Lomakkeen testausta on suoritettu enimmäkseen Visual Studion 2010 sisältämässä local server -ympäristössä, sekä Pohjois-Karjalan ammattikorkeakoulun tietohallinnon palve- limella. Visual Studio 2010 tarjoaa käyttäjälleen ASP.NET Development Server - ominaisuuden, jolla ohjelmoija voi testata tuottamaansa sovellusta paikallisella palveli- mella. Tällä ominaisuudella lomaketta on ollut mahdollista testata koko sen kehityksen ajan. ASP.NET Development Server -ominaisuuden ansiosta lomakkeen testaus ja kehi- tys on ollut tehokasta ja se on mahdollistanut lomakkeen testauksen koko ohjelmiston

tuottamisen ajan. Virheiden minimointi on ollut helppoa ja jokaiselle lomakkeen testauskerralla olen pystynyt vähentämään ja eliminoimaan virheen mahdollisuutta.

#### **4.5 Käyttöönotto ja projektin päättäminen**

Pohjois-Karjalan ammattikorkeakoulun tietohallinto vastaan lomakkeen siirtämisestä tuotantoympäristöstä Pohjois-Karjalan ammattikorkeakoulun palvelimelle. Valmistuneet lomakkeen tiedostot lähetetään tietohallinnon yhteyshenkilölle, joka hoitaa tiedostojen siirtämisen heidän aikataulujensa mukaisesti palvelimelle.

Lähdetiedostojen lähetys tapahtuu sähköpostitse pakattuna tiedostona tietohallinnon yhteyshenkilölle. Lähdekoodissa on kommentoitu ohjelmoituja toimintoja, jotka helpottavat lomakkeen toiminnan ymmärtämistä. Lähdekoodin kommentointi on tärkeää lomakkeen jatkokehityksen kannalta, jotta tuleva ohjelmoija ymmärtää ohjelmallista, että toiminnallista puolta lomakkeesta. (Vesterholm & Kyppö 2006, 6.)

Projektin päättäminen tapahtuu silloin kun projektityö on onnistuneesti lähetetty tietohallinnolle, sekä opintopsykologi on antanut oman hyväksynnän lomakkeen sisällölle, että ulkoasulle.

### **5 Yhteenveto**

Projektin tuloksena sain tuotettua opintopsykologille hänen vaatimuksiensa mukaisen sähköisen yhteydenottolomakkeen. Teknisiltä ominaisuuksiltaan sähköinen yhteydenottolomake on Pohjois-Karjalan ammattikorkeakoulun tietohallinnon vaatimuksien mukainen.

Projektin aikana jouduin kohtaamaan monia eri ongelmia teknisten ratkaisujen suhteen. Ongelmat koostuivat enimmäkseen ohjelmoinnin aikana tulleista ongelmista. Ongelmat johtuivat ohjelmointikielen tuntemuksen puutteesta. Koska ohjelmointikieli ei ollut ennudestaan itselleni tuttu, jouduin paneutumaan ohjelmointikielen saloihin syvällisemmin. Koko projektin edistymisen aikana opin entistä enemmän ohjelmointikielestä, ja

vastaan tulleiden ongelmien ratkaisu helpottui huomattavasti. Kaiken kaikkiaan koko projekti oli itselleni iso oppimisprojekti, joka kehitti itseäni niin ohjelmoijana kuin ihmisenäkin.

Yhteydenpito sekä toimeksiantajan, että opinnäytetyön ohjaajan kanssa oli tiivistä. Kävimme läpi lomakkeen toiminnallisuutta ja sen ulkoasua monena eri tapaamiskertana ja välillemme syntyi luonteva ja avoin vuorovaikutussuhde.

Koen, että projekti on onnistunut ja saavutin projektissa ne tavoitteet, mitkä oli asetettu toimeksiantajan itseni toimesta. Oma näkemykseni on, että sähköinen yhteydenottolomake tulee varmasti helpottamaan opintopsykologin työtä ajanvarauksen suhteen, ja lomakkeen ansioista opintopsykologi saa ne tiedot mitä hän tarvitsee sekä opiskelijan tunnistamista että ajanvarausta varten. Lomake myös mahdollistaa opiskelijalle helpomman ja nopeamman yhteydenottomahdollisuuden opintopsykologiin. Joissakin tapauksissa opiskelijoiden yhteydenottokynnys laskee lomakkeen yksinkertaisuuden ja nopean täyttämisen ansioista.

Sähköisen yhteydenottolomakkeen käyttöönoton jälkeen tulee usein tilanteita, jossa käytettävän lomakkeen käyttötarkoitus tai järjestelmä tulee muuttumaan. Käyttötarkoitusten muuttuessa tai järjestelmien kehittyessä lomakkeeseen halutaan kehittää yleensä uusia ominaisuuksia, jotka vastaavat sen hetkistä käyttötarkoitusta. Itselleni vastaan tulleita mahdollisia jatkokehitys vaihtoehtoja lomakkeen suhteen nousi muutamia esille. Lomaketta olisi tulevaisuudessa mahdollista laajentaa muillekin Pohjois-Karjalan ammattikorkeakoulun henkilöstölle ja näin helpottaa heidän sähköistä yhteydenpitoa opiskelijaan tai mahdolliseen sidosryhmään. Lomakkeen mahdollinen muokkaus on suhteellisen yksinkertaista, eikä se vaadi isoja ponnisteluja ohjelmoijalta, mikäli ohjelmoija on tutustunut lähdekoodin sisältöön huolellisesti. Mahdollisten uusien ominaisuuksien kartoitus ja lisäys täytyy suunnitella yhdessä lomakkeen käyttäjän ja ylläpitäjän kanssa.

Sähköisen yhteydenottolomakkeen jatkokehityksestä vastaa Pohjois-Karjalan ammattikorkeakoulu yhdessä tietohallinnon kanssa. Näkisin, että mikäli lomaketta halutaan laajentaa tai muokata eri käyttötarkoituksia varten niin tästä voisi saada uuden todella laajan ja mielenkiintoisen oppimisprojektin.

## Lähteet

Alfame system Oy. 2011. Ohjelmiston uuden ominaisuudet.

<http://www.alfame.com/development/uudet-ominaisuudet>. 20.4.2012.

Haikala, I & Mikkonen, T. 2011. Ohjelmisto tuotannon käytännöt. Helsinki. Talentum.

Horsmalahti, P. & Toivanen, M. 2011. Ohjelmointikielten periaatteet C#-

ohjelmointikieli. [http://www.cs.tut.fi/~okp/seminaari/27\\_CSharp.pdf](http://www.cs.tut.fi/~okp/seminaari/27_CSharp.pdf).

23.2.2012.

Järvinen, J. 2008. Visual Studio 2008 –käsikirja. Jyväskylä. Docendo.

Kolari, M. 2010. C#-poikkeukset.

<http://mikakolari.fi/csharp-dotnet/perusteet/poikkeukset>. 11.4.2012.

Moghadampour, G. 2009. C# ohjelmointi, Jyväskylä, WSOYpro Oy.

Polvinen, T. 1999. Tietokannat käytännön työssä. Porvoo. WSOY Oy.

Vesterholm, M. & Kyppö, J. 2006. Java-ohjelmointi. Helsinki. Talentum.

Viestintävirasto. 2008. Sähköisen viestinnän suojaaminen.

<http://www.ficora.fi/viestinsuojaus/index.html>. 10.2.2012.

Viestintävirasto. 2012. Tietoturvan peruskäsitteitä

<http://www.ficora.fi/index/palvelut/palvelutaiheittain/tietoturva.html>.

10.2.2012.